HAL
open science

Lire les lectures : analyse de données de séquençage

Pierre Peterlongo

▶ **To cite this version:**

Pierre Peterlongo. Lire les lectures : analyse de données de séquençage. Bio-informatique [q-bio.QM]. Université rennes1, 2016. tel-01278275

HAL Id: tel-01278275

https://inria.hal.science/tel-01278275v1

Submitted on 24 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Inria Rennes Bretagne Atlantique
Équipe projet GenScale

Habilitation à Diriger des Recherches
présentée par

Pierre PETERLONGO

Lire les lectures :

analyse de données de séquençage

Soutenue publiquement le 25 janvier 2016
devant le jury composé de

Hélène Touzet	Rapportrice	Directrice de Recherche, Inria Lille
Sophie Schbath	Rapportrice	Directrice de Recherche, INRA, Jouy-en-Josas
Gunnar Klau	Rapporteur	Professeur, CWI Amsterdam
Alain Viari		Directeur de recherches, Inria Lyon
Philippe Vandenkoornhuyse		Professeur, Université de Rennes1
Guillaume Blin		Professeur, Université de Bordeaux

Table des matières

Remerciements **iii**

Introduction **v**

1 Contexte **1**
1.1 Le séquençage 2
1.2 L'évolution des technologies de séquençage 3
1.3 Données issues de séquenceurs haut débits 4
1.4 Les défis algorithmiques associés aux données NGS 6
1.5 Pour résumer 14

2 Détection de variants **15**
2.1 Introduction 15
2.2 Présentation de quelques variants 16
2.3 Estimer la qualité de la détection de variants 19
2.4 Détection de variants par mapping sur séquence de référence 20
2.5 Modèles pour la détection de variants *de novo* 23
2.6 Mise en oeuvre algorithmique 28
2.7 Difficultés et solutions 29
2.8 Résultats 31
2.9 Perspectives 36
2.10 Présentation des publications associées 38

3 Comparaison de données de séquençage **41**
3.1 Introduction 41
3.2 Brève introduction à la métagénomique comparative 46
3.3 Outils de détection de lectures similaires entre jeux de séquences NGS 48
3.4 Résultats 55
3.5 Outils de comparaisons d'ensembles de k-mers similaires entre jeux de séquences NGS 60
3.6 Perspectives 62
3.7 Présentation des publications associées 63

4 Mapper les lectures sur des graphes **65**
4.1 Introduction . . . 65
4.2 Mapping sur graphe . . . 65
4.3 Solution pour le mapping intensif de séquences sur dBG . . . 67
4.4 Perspectives pour l'amélioration du mapping sur graphe . . . 71
4.5 Présentation des publications associées . . . 72

5 Perspectives **75**
5.1 Représentation des génomes . . . 75
5.2 Méthodologie pour la (multi-)métagénomique massive . . . 77
5.3 Perspectives personnelles . . . 80

Glossaire **81**

Bibliographie **87**

Curriculum Vitæ **101**
État civil . . . 101
Formation scientifique . . . 101
Expériences professionnelles . . . 102
Principales contributions à l'animation et à la diffusion de l'information scientifique . . . 102
Activités scientifiques . . . 103
Liste de mes publications . . . 106

Remerciements

Ces neufs années écoulées depuis ma soutenance de thèse ont conforté mon enthousiasme pour le métier de chercheur. De nombreuses facettes illuminent mon quotidien et l'une d'entres elles est la richesse des relations humaines et des interactions qui sont, selon moi, la colonne vertébrale de la réussite de ce métier. Ces quelques lignes de remerciements ne représentent que trop pâlement ce sentiment et se limitent à quelques personnes alors que ces pages pourraient en mentionner des dizaines et des dizaines.

Je tiens à remercier chaleureusement mes trois rapporteurs : Sophie Schbath, Hélène Touzet et Gunnar Klau. Votre travail m'a apporté un point de vue extérieur sur mon parcours scientifique et le réconfort d'apprendre que ce document peut avoir une visée pédagogique, ce qui me tenait à coeur lors de sa rédaction. Merci à tous les membres de mon jury : mes trois rapporteurs accompagnés de plus par Guillaume Blin, Philippe Vandenkoornhuyse et Alain Viari. Merci pour votre intérêt dans mon travail, et votre présence lors de ma soutenance. Je suis fier et vous suis reconnaissant de pouvoir vous compter dans ce jury d'HDR.

J'aimerais remercier du fond du coeur Marie-France Sagot. J'admire ta ténacité, ton engagement pour ton métier et tes convictions. Merci encore une fois de m'avoir mis sur le chemin de la bio-informatique et d'avoir continué à me soutenir à et à me promulguer de précieux conseils durant toutes ces années. Même si nous sommes géographiquement plus distants, ton point de vue et ta façon de considérer ce métier influent largement ma vision et mes choix.

Merci à Dominique Lavenier qui a cru en moi il y a quelques années et qui me supporte depuis ! Merci à celle et ceux que j'ai vu diriger les équipes Symbiose, Genouest, Dyliss, et GenScale : Dominique mais aussi Jacques Nicolas, Anne Siegel et Olivier Collin. Je ne sais pas quelle est l'alchimie que vous saupoudrez sur ces équipes, mais il y a quelque chose qui fait "que ça marche" et qu'on se sent bien tant humainement que scientifiquement. J'en profite pour remercier globalement toutes ces chouettes personnes qui font de Symbiose une communauté accueillante, vivante et dynamique. En particulier j'adresse un clin d'oeil spécial à mon co-bureau Olivier Dameron qui offre un parfait équilibre entre blagues pourries et moments sérieux ; tout est parfait, même si j'espère ne jamais avoir le hoquet dans le bureau.

Merci à toutes celles et tous ceux avec qui j'ai eu la chance de collaborer. La liste serait trop longue à établir ici et je risquerais d'oublier du monde. Cependant, je tiens à remercier plus particulièrement quelques personnes qui ont marqué scientifiquement et humainement cette partie de ma carrière. Merci à Guillaume Holley, qui, discrètement, était là lors des balbutiements de deux des contributions principales présentées dans ce manuscrit. Merci à Nico Maillet, mon thésard-ami pour ton travail sur la comparaison de métagénomes. C'était super de commencer l'encadrement de

doctorat par quelqu'un comme toi. J'attend avec impatience la suite de nos travaux communs et les prochaines séances de grimpe ensembles. Merci à Raluca pour ton passage remarqué chez nous. Merci pour ce que tu as fait pour la détection de variants et bien sur aussi pour ta bonne humeur et ta gentillesse, dont je n'ai pas pu entrevoir les limites. Merci au noyau dur de l'équipe GATB : Guillaume Rizk, Erwan le Drezen, Rayan Chikhi. On doit une large partie des succès de l'équipe à vos prouesses algorithmiques et techniques. Merci à Vincent Lacroix et Éric Rivals : Les moments de discussion scientifiques ou non sont toujours agréables, réconfortants et fructueux ; et donc trop rares.

Un merci particulier à Claire Lemaitre. T'as été une amie avant d'être une collègue. Aujourd'hui, cet ordre est toujours le même et pourtant on est très collègues. J'ai l'impression que notre complémentarité nous a permis de réaliser de beaux projets. J'espère qu'il y en aura de nombreux autres.

Merci à mes amis et mes proches. La vie est belle, et c'est en large partie grace à vous. Merci donc pour votre présence, que vous veniez de l'irisa, de la grimpe, de la voile, du quartier Castors ou d'ailleurs. Merci à ma famille à la fois si discrète et si intensément proche. Merci papa pour ta relecture de ce document, merci Grand-Papa de m'avoir poussé, sans le savoir, à aller vers ce métier et à continuer d'y avancer.

J'aimerais adresser un message particulier à Philou, et à Xavier. Ces fiertés, ces joies, j'aimerais pouvoir encore les partager avec vous. Vous êtes toujours bien présents dans ma vie et vous continuer d'en influencer les choix.

* *

*

Merci à Émilie pour tout ce que l'on partage, pour tout ce que l'on a construit. Merci pour ce petit miracle quotidien que tu m'offres depuis ces quelques années, passées comme un souffle à tes cotés. Merci à Olivia et Loïc. Vous voir grandir éclaire ma vie que vous comblez de votre amour sans limite. Je vous aime tous les trois.

Introduction générale

Présentation du document

Ce document est principalement écrit à la première personne du pluriel “nous”. Ceci ne révèle pas une quelconque forme de modestie ou de schizophrénie mais traduit le fait qu’aucun des résultats qui est présenté ici n’aurait été le même si j’avais effectué le travail seul. Lorsqu’il ne s’agit pas de décider du choix d’un resto, on est toujours plus intelligent à plusieurs que seul. Les meilleurs moments de cette première partie de ma de carrière ont toujours été des moments partagés, des moments où l’on sent poindre une bonne idée, où l’on découvre un beau résultat. J’aime ces moments de partages, de bouillonnement intellectuel où chacun a sa place, pour amener une nouvelle suggestion ou améliorer celles des autres. Pour être efficace et agréable, la recherche n’est pas et ne doit pas devenir une histoire d’individualité, de mise en concurrence camouflée par une *excellence* non partagée.

* *

*

Ce document se veut lisible par des lecteurs non-spécialistes du domaine de la biologie moléculaire ou de l’algorithmique. C’est pourquoi un glossaire est proposé page 81. À partir du chapitre suivant et lors de leur première apparition dans le texte, les mots présents dans le glossaire sont en italiques et sont suivis d’une *étoile**.

* *

*

Ce document offre une synthèse des principaux travaux que j’ai effectués ces dernières années. J’ai choisi de centrer le discours sur un projet central de mes activités, à savoir l’exploitation selon diverses méthodes des données de séquençage d’espèces non modèles. Il est à noter que, de part ce choix, j’ai délibérément mis de coté divers aspects de mes recherches, en particulier ceux concernant le développement d’algorithmes associés aux structures d’indexations, comme la mise à jour des tableaux des suffixes [Gallé et al., 2009], la création d’automates ou d’arbres des suffixes dits “*à trous*” [Antoniou et al., 2006 ; Peterlongo et al., 2006, 2007a] ainsi que les stratégies d’indexation adaptées à la comparaison protéique sur matériels reconfigurables [Peterlongo et al., 2007b, 2008a] ou encore la détection de marqueurs moléculaires pour la différenciation de souches bactériennes [Peterlongo et al., 2009a].

Organisation du document

Le document se divise en quatre parties distinctes. La première partie (Chapitre 1 page 1) présente le contexte biologique et le contexte technologique qui ont motivé et sur lesquels sont basés les travaux présentés dans cet ouvrage. Les trois parties suivantes présentent les trois apports principaux présentés dans ce document. Tous les travaux présentés concernent l'exploitation de données de séquençage haut débit en absence de génome de référence proche et de bonne qualité. Dans le Chapitre 2 page 15, nous proposons de nouvelles approches pour extraire des variants biologiques d'intérêt de ces données de séquençage. Dans le Chapitre 3 page 41 nous exposons des méthodes de comparaisons de jeux de données. Dans le Chapitre 4 page 65, nous proposons une méthode préliminaire à de meilleurs "assemblages" de ces données de séquençage.

Enfin, dans le Chapitre 5 page 75, nous proposerons les perspectives de ces travaux à moyen et plus long terme.

* *

*

Un CV est proposé page 101. Il présente les différents points nécessaires à l'établissement du dossier d'HDR.

Chapitre 1

Contexte

Contents

1.1 Le séquençage 2
1.2 L'évolution des technologies de séquençage 3
1.3 Données issues de séquenceurs haut débits 4
1.4 Les défis algorithmiques associés aux données NGS 6
1.5 Pour résumer 14

Ce court chapitre présente le contexte dans lequel s'inscrivent les travaux présentés dans ce document. Nous ne ferons pas de présentation détaillée de la biologie cellulaire. Il existe de nombreux ouvrages scientifiques décrivant nos connaissances de ces mécanismes, accessibles au grand public [Alberts et al., 1999], ou destinés aux lecteurs plus aguerris [Lodish et al., 2000]. Une telle présentation dans ce manuscrit serait redondante avec ce type d'ouvrage et n'est pas indispensable pour appréhender les problématiques et les méthodes présentées ici.

Nous nous contenterons ici de nous baser sur le "dogme central" brièvement présenté dans ce qui suit. Notons tout de même que cette vision de la biologie cellulaire est clairement limitée et ne permet pas, loin de là, de comprendre le fonctionnement de nombreux phénomènes biologiques à l'image par exemple des virus à ARN, ou de la transcription inverse présente chez les rétrovirus.

Le dogme de la biologie moléculaire, schématiquement représenté Figure 1.1 page suivante, décrit les étapes successives permettant de synthétiser des *protéines** à partir d'un *génome**, à savoir l'information portée par l'*ADN**. La *transcription** synthétise les molécules d'*ARN** à partir de l'information portées sur les *gènes** présents sur l'ADN. Dans certains cas, ces molécules d'ARN sont épissées : certaines portions, appelées introns sont excisés, et les parties restantes (les exons) sont réunies. Lorsque les introns ne sont pas supprimés dans leur totalité on parle d'épissage alternatif. Dans cette présentation simplifiée, la dernière étape, appelée la "traduction", consiste à générer des protéines à partir de l'information portée sur les molécules d'ARN.

Les protéines ont des rôles très divers comme par exemple un rôle messager (processus de signalisation cellulaire), structurel (cohésion structurelle entre cellules) ou encore enzymatique (catalyseurs de réactions biochimiques). Les protéines, directement dépendantes des gènes dont elles sont issues, représentent un chaînon du lien entre le *génotype** et le *phénotype**.

La clef de la compréhension du vivant passe donc par la connaissance des génomes, de leur

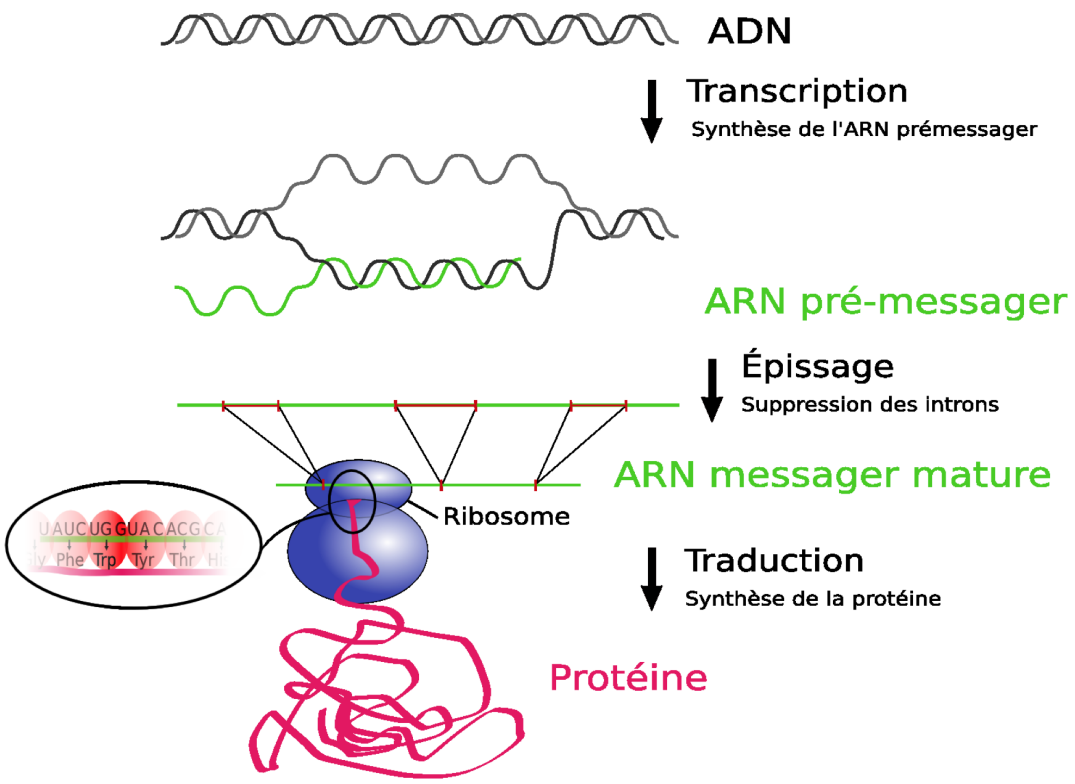


FIGURE 1.1 – Le *dogme central* de la biologie moléculaire

fonctionnement, et des mécanismes qui les régissent. Déterminer la séquence génomique d'individus vivants a des intérêts à la fois fondamentaux pour comprendre comment les organismes survivent et se reproduisent, mais a aussi des intérêts appliqués avec de très forts impacts dans des domaines majeurs de la santé, de l'agronomie, ou de l'environnement. Ceci a motivé de très gros efforts de recherche et de développement pour permettre le *séquençage** des génomes, comme nous le présentons dans la section suivante.

1.1 Le séquençage

À partir d'un échantillon biologique, le séquençage consiste à lire l'ADN ou l'ARN contenu dans les cellules de cet échantillon pour produire un texte numérique composé d'une succession de lettres dans l'alphabet A, C, G, et T ou A, C, G, et U dans le cas de l'ARN.

Dans les années 1970, la technologie Sanger [Sanger et al., 1977] a permis de séquencer les premiers organismes. Cependant, cette technologie avait de gros inconvénients. Elle était particulièrement onéreuse et nécessitait d'importantes interventions humaines. De plus, cette technologie avait un débit très bas. Malgré ces inconvénients, cette technologie n'a cessé d'être améliorée durant près de trente années. La technologie Sanger a connu son apogée dans les années 90 où elle a été employée

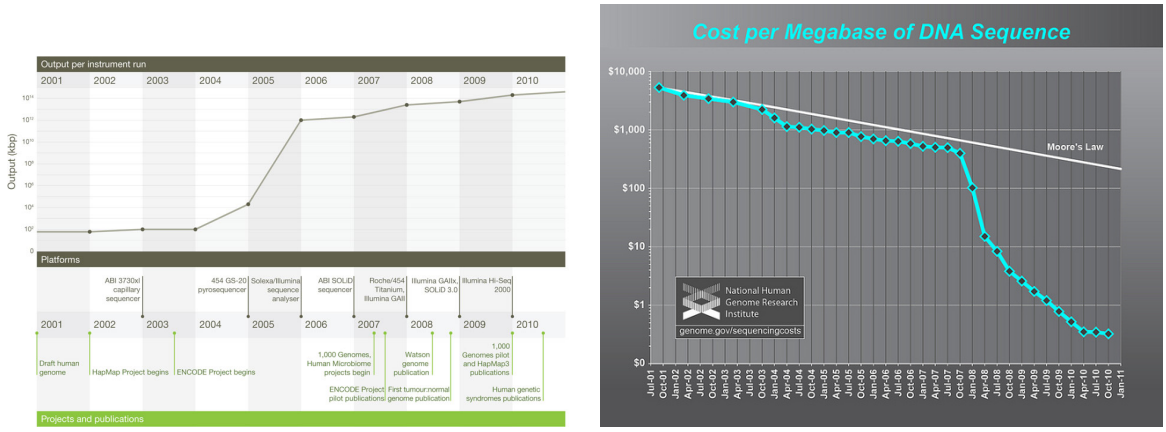
pour séquencer pour la première fois des individus humains, publiés en 2001 simultanément dans les revues *Nature* [Lander et al., 2001] et *Science* [Venter et al., 2001]. Ceci a nécessité dix années de travail et le montant total de cette réalisation a été estimé à plusieurs milliards de dollars.

Dans le milieu des années 2000, la biologie moléculaire a été témoin d'un changement drastique, un changement que certains, dont je fais partie, n'hésitent pas à appeler "révolution". Cette révolution est due à l'avènement d'une nouvelle technologie permettant le séquençage des génomes. Cette nouvelle technologie est connue sous le nom de *High Throughput Sequencing* (*HTS*) ou de *Next Generation Sequencing*, également appelé *NGS*, comme ce sera le cas tout au long de ce document.

La très grande majorité de mon travail de ces six dernières années a été motivée par les nouveaux problèmes soulevés par les données produites par les NGS. Il est assez étonnant de réaliser que les NGS sont nés de recherches à la pointe de la chimie, de la biologie et de la physique, mais que finalement les informaticiens, et en particuliers les bio-informaticiens dont je fais partie, ont été tant à la traine. En effet, lorsque les données NGS ont commencé à être produites en masse dans le courant des années 2006-2007, la communauté bio-informatique n'était pas prête à les recevoir : les analyser, les corriger, les compresser, les transférer ou même les stoker représentaient de nouveaux problèmes.

J'ai eu le privilège de faire partie de celles et ceux qui ont œuvré pour permettre aux biologistes d'accéder à l'information biologique contenue dans ces données.

1.2 L'évolution des technologies de séquençage



(a) **(b)**

FIGURE 1.2 – Deux témoins principaux de l'avènement des NGS : **(a)** la capacité de séquençage par run (image source :[Mardis, 2011]) ; et **(b)** le coût du séquençage par mégabase (Image Source : NIH National Human Genome Research Center).

Les années 2005 à 2007 ont été une période charnière durant lesquelles les NGS ont commencé à bouleverser le paysage du séquençage. Les technologies NGS sont basées sur une parallélisation massive des réactions biochimiques permettant la lecture de chacun des *nucléotides** composant l'ADN ou l'ARN. En outre, la fracture technologique entre la technologie de type Sanger et les NGS

est due à la disparition de l'étape d'*électrophorèse** qui ralentissait le séquençage et en augmentaait considérablement le coût.

Comme nous pouvons le constater Figure 1.2 page précédente, ces années, via l'avènement des données NGS, ont été témoins d'un accroissement considérable des capacités de séquençage et d'une baisse simultanée drastique des coûts associés. Le terme de "*démocratisation*" du séquençage a souvent été utilisé à cette période. Ce terme est quelque peu exagéré. Cependant, avant l'arrivée de ces technologies, l'accès au génome était limité à de très gros projets et donc à quelques espèces de références. Les NGS ont alors permis à nombre de projets de plus petite envergure (financière) d'avoir accès aux données génomiques des espèces étudiées.

Notons toutefois que la *démocratisation* du séquençage est en bonne voie. Le cap des 1000US$ pour le séquençage du génome humain est actuellement en train d'être franchi [Sheridan, 2014].

1.3 Données issues de séquenceurs haut débits

Les séquenceurs, quels qu'ils soient, n'ont pas (encore) la capacité de lire de gauche à droite l'intégralité de la séquence d'un génome. Jusque très récemment, les séquenceurs n'étaient capables que de lire des séquences de quelques dizaines à quelques centaines de nucléotides. En effet, les deux technologies dominant le marché, produisent des séquences de 100 à 300 nucléotides pour la technologie Illumina ou jusqu'à 400 nucléotides pour la technologie Roche 454.

Les données fournies par un séquenceur sont donc un ensemble de millions, voire de milliards, de courtes séquences représentées sur l'alphabet $\{A, C, G, T\}$ pour l'ADN ou $\{A, C, G, U\}$ pour l'ARN. Notons que la lettre N, issue de l'alphabet IUPAC [on Biochem. Nomenclature , CBN], est régulièrement utilisée pour désigner un caractère inconnu. Chacune de ces séquences est appelée une "*lecture**" issue du terme anglais "*read*". Ces *lectures* sont généralement fournies sous forme de fichiers FASTA. Chaque lecture est indiquée sur une ligne précédée par un en-tête débutant par le caractère '>' et proposant quelques indications sur cette lecture. Un second format, appelé FASTQ, ajoute une mesure de qualité pour chaque nucléotide séquencé. Notons que les informations connues sont bien maigres. C'est là l'une des principales limitations des NGS : pour chaque lecture on ne connait ni sa position dans le génome dont elle est issue, ni son orientation (un brin d'ADN pouvant être lu dans les deux sens).

Les NGS sont capables de lire en un *run* une quantité de nucléotide supérieure à la taille du génome séquencé (cf Figure 1.3 page suivante qui présente la taille et le nombre de lecture par run en fonction de la technologie employée). Ainsi, en moyenne, chaque position du génome, autrement-dit chaque nucléotide, est séquencé plusieurs fois. Le nombre moyen de fois que chaque nucléotide est séquencé est appelé la *couverture* du séquençage. Ainsi, à l'image d'un puzzle, même si les positions et l'orientation des lectures sont inconnus, il est théoriquement possible d'utiliser la redondance et donc le chevauchement de celles-ci pour les organiser les unes par rapport aux autres pour reconstituer la séquence originale dont elles sont issues. C'est ce que l'on appelle l'*assemblage**.

Comme nous le verrons plus en détails dans la Section 1.4 page 6, plusieurs limitations liées entre autres aux biais technologiques font de l'assemblage un problème difficile. Les NGS n'ont pas une précision de 100%. Par rapport au génome séquencé, les lectures peuvent contenir des insertions, des délétions ou des substitutions. Ainsi, deux lectures provenant de la même position sur le génome (le même locus), peuvent être différentes en raison de ces erreurs de séquençage. En

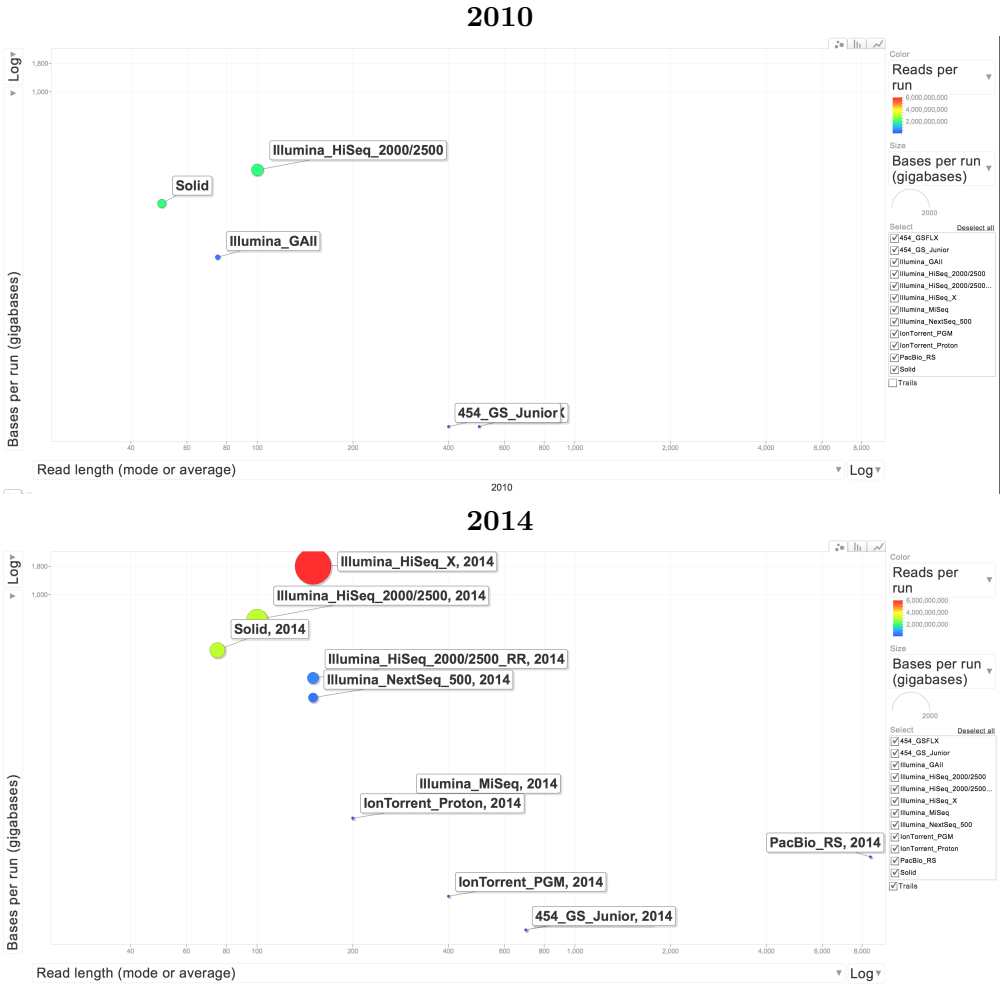


FIGURE 1.3 – Caractéristiques et évolutions des technologies de séquençage. Image générée à partir du travail de Lex Nederbragt https://flxlexblog.wordpress.com.

outre, la couverture le long du génome n'est pas uniforme. Des biais existent. Ils sont principalement liés à la phase d'amplification des génomes nécessaire au séquençage. Il en résulte que la couverture, bien qu'intéressante, n'est pas une information parfaitement fiable.

En résumé, les données issues des NGS sont composées d'un nombre très important de courts fragments de séquences, potentiellement erronés, dont on ne connait ni la position ni l'orientation sur le génome. En raison de leur redondance, les chevauchements entre ces fragments peuvent être utilisés pour tenter de retrouver la séquence originale. C'est ce que l'on appelle l'assemblage.

* *

*

Il est délicat de dresser un inventaire des technologies de séquençage et de leurs caractéristiques

tant elles évoluent rapidement. La technologie Illumina est actuellement majoritairement utilisée en raison de son prix qui est le plus bas du marché, de son faible taux d'erreurs (actuellement aux alentours de 0.1%) principalement limitées aux substitutions, et de son très important débit. Cependant, comme représenté Figure 1.3 page précédente, les technologies évoluent constamment et cette situation est susceptible de changer extrêmement rapidement.

Les annonces de technologies dites de troisième génération (appelées TGS pour *third generation sequencing*) sont nombreuses depuis quelques années. Les TGS consistent, à l'image de la technologie *PacBio*, au séquençage de lectures beaucoup plus longues pouvant atteindre jusqu'à plusieurs dizaines de milliers de nucléotides. Depuis peu, ces données d'un nouveau type commencent à arriver concrètement dans les laboratoires et génèrent de nouveaux besoins méthodologiques adaptés à leurs caractéristiques. Les méthodes algorithmiques présentées dans ce document concernent principalement les données de type Illumina, à savoir, rappelons-le, des centaines de millions de lectures courtes (100 à 300 nucléotides), dont les erreurs de séquençage varient de 0.1 à 1% et se limitent principalement à des substitutions.

1.4 Les défis algorithmiques associés aux données NGS

La plupart des méthodes d'algorithmique des séquences utilisées sur les données génomique issues de séquençage de type Sanger ne passaient pas à l'échelle lors de l'arrivée massive des données de séquençage NGS. Il faut garder à l'esprit que ces méthodes, à l'image de mon travail de thèse [Peterlongo, 2006], s'appliquaient principalement sur les génomes ou transcriptomes issus d'espèces modèles, alors limitées à quelques dizaines.

Les données NGS ont ravivé les besoins fondamentaux associés à l'algorithmique du texte. Les défis à relever étaient et sont toujours passionnants. En effet, ces données ont pour caractéristiques principales leur incroyable complexité et leur volume jusqu'alors jamais rencontré dans la discipline de la bio-informatique. Ainsi il fallait et il faut toujours trouver des méthodes ayant un bon compromis entre la qualité des résultats fournis et les ressources nécessaires à leur fonctionnement en terme d'espace disque, de mémoire et de temps d'exécution.

Outre les aspects de gestion pure des données (transfert, compression, stockage, sécurisation, ...), les besoins algorithmiques principaux associés à ces données sont la *correction**, le *mapping**, et l'assemblage. Bien entendu ces traitements ne sont pas des finalités en soi, et les méthodes d'analyse en aval permettent d'extraire l'information biologique attendue en fonction de l'application visée.

Mapping sur une référence Étant donnés un génome de référence et un fichier de lectures NGS, le mapping consiste à placer chaque lecture sur le génome de référence. Pour chaque lecture, il s'agit de l'alignement complet (global) de la séquence de cette lecture sur un locus de la référence (local). Généralement l'alignement est très "stringeant" car peu de différences sont autorisées entre la référence et chaque lecture. Les méthodes de mapping utilisent généralement des algorithmes heuristiques de type *seed-and-extend** permettant de bon résultats de précision/recall tout en limitant les utilisations de ressources temps et mémoire. L'outil Blast [Altschul et al., 1990] est l'un des outils d'alignement les plus connus et les plus utilisés. Il s'agit d'une heuristique permettant d'*aligner* rapidement des séquences requêtes sur une séquence de référence. La publication associée est l'une des publications scientifiques les plus citées, toutes disciplines confondues. Cependant, dans

le cas du mapping de lectures sur génome de référence, le problème est loin d'être simple car, nous faisons face à des masses de données considérables. La solution apportée par la suite d'outils dérivée de Blast ne passe pas à l'échelle. Les mappeurs actuels, à l'image de Bowtie [Langmead et al., 2009] ou BWA [Li and Durbin, 2009], sont capables de traiter plusieurs dizaines de millions de lectures par heure. D'autre problématiques sont liées à l'organisation des génomes en particulier *eucaryotes**, comprenant de nombreuses régions répétées (cf ci-dessous), sources de mapping multiple (lectures qui *mappent* à différents loci du génome de référence).

Les répétitions génomiques À l'image des difficultés qu'elles génèrent pour le mapping, les répétitions intra génomes représentent l'une des sources les plus importantes de difficultés lors du traitement de données NGS. Les répétitions intra génomes, en particulier chez les génomes eucaryotes, sont nombreuses et peuvent représenter une part significative allant de 10 à 20% du génome d'un individu [Gemayel et al., 2010]. Ces répétitions induisent une plasticité qui représente un mécanisme clef de l'évolution des espèces [Jurka, 1998]. Leur détection et leur analyse est donc un domaine d'intérêt particulièrement actif [Anisimova et al., 2015].

Dans le contexte de l'exploitation des données de séquençage, les répétitions, dès lors qu'elles sont plus longues que les lectures ou que les k-mers (cf définitions des k-mers Section 1.4.2 page suivante), sont une source importante de complications dans le traitement des données.

1.4.1 L'assemblage de données NGS

Comme nous l'avons mentionné, l'assemblage des données NGS consiste à résoudre le puzzle géant constitué de centaines de millions de pièces (les lectures). La décision d'assembler deux lectures se fait via leur chevauchement qui peut être approximatif du fait des erreurs de séquençage.

Les premières méthodes d'assemblage de génomes étaient basées sur la comparaison de chacune des paires possibles de lectures présentes dans les jeux de données. Ces méthodes sont désignées par le sigle "OLC" pour *Overlap Layout Consensus*. Une fois que les chevauchements entre paires de lectures sont déterminés, celles-ci sont organisées dans un graphe où chaque lecture est stockée dans un noeud et où une arrête relie deux noeuds si les deux lectures correspondantes sont considérées comme chevauchantes. La lecture de chemins dans ce graphe fournit des portions de séquences assemblées que l'on appelle *contigs**. Les assembleur OLC ont été utilisés avec succès pour assembler des données NGS composées de peu de lectures (moins d'un million) assez longues ($\geq$ 400 nucléotides). Parmi les plus utilisés nous pouvons citer Arachne [Batzoglou et al., 2002], Celera Assembler [Myers et al., 2000], PCAP [Huang et al., 2003], Phrap [de la Bastide and McCombie, 2007], ou Newbler [Margulies et al., 2005]. Les méthodes OLC atteignent leurs limites lorsque le nombre de lecture à assembler devient trop important. En effet, la comparaisons de toutes les lectures deux à deux et le stockage de l'intégralité de ces lectures demandent une quantité de mémoire et un temps de calcul rédhibitoires pour l'analyses de données de séquençage de type Illumina.

Comme nous le détaillerons dans la Section 1.4.2 page suivante, motivée par les limitations de l'approche OLC, une seconde approche, nettement moins intuitive a été développée et appliquée avec succès aux données NGS de type Illumina.

Débutons par quelques définitions préliminaires.

Définitions de base

Definition 1.1 (séquence, alphabet, k-mer)**.** Une séquence est une suite constituée de zéro caractère ou plus. Ces caractères appartiennent à un alphabet noté Σ. Une séquence de longueur n ($\in \Sigma^n$) est notée $s[0]s[1]\dots s[n-1]$. La notation $|s|$ désigne la taille de la séquence s. Avec $k \in [0, n-1]$ et $i \in [0, n-k-1]$, le mot $s[i]s[i+1]\dots s[i+k-1]$ est appelé un k-mer de s apparaissant position i. Notons qu'il existe $|s| - k + 1$ k-mers (possiblement redondants) sur une séquence s.

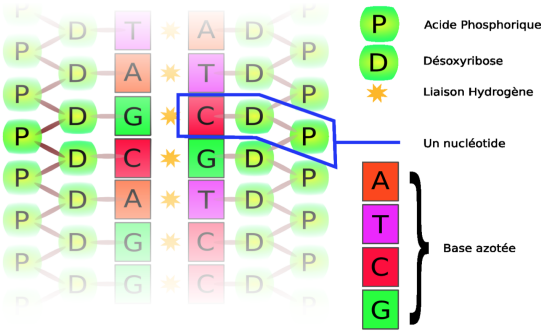


FIGURE 1.4 – Composition de l'ADN

Rappelons que l'ADN a une structure en double hélice. Comme représenté Figure 1.4, les nucléotides A et T s'apparient entre eux et les nucléotides C et G s'apparient entre eux. Ainsi une séquence d'ADN peut être lue dans les deux sens, appelés *forward* et *reverse*. La courte séquence présentée Figure 1.4 lue à gauche de haut en bas est $TAGCAGG$. La même séquence lue à droite de bas en haut est $CCTGCTA$, que l'on appelle le reverse complément de $TAGCAGG$.

Definition 1.2 (reverse complement)**.** Étant donnée une séquence s sur l'alphabet $\Sigma = \{A, C, G, T\}$, le reverse complément de s, noté $\overleftarrow{s}$ est la séquence $\overleftarrow{s[|s|-1]}\ \overleftarrow{s[|s|-2]}\dots\overleftarrow{s[0]}$, avec $\overleftarrow{A} = T$, $\overleftarrow{T} = A$, $\overleftarrow{C} = G$, et $\overleftarrow{G} = C$.

1.4.2 Assemblage par k-mers et dBG

Une seconde approche d'assemblage consiste à utiliser la notion de k-mer. L'idée sous-jacente consiste à énumérer l'intégralité des k-mers présents dans un jeu de lectures. En supposant que les lectures soient exemptes d'erreurs de séquençage, le nombre de k-mers distincts présents dans un ensemble de lectures ne peut être supérieur à la longueur du génome séquencé, quelque soit le nombre de lectures à assembler. La seconde idée majeure se base sur le fait que deux lectures suffisamment chevauchantes pour être assemblées partagent au moins un k-mer. Cependant, les assembleurs basés sur l'utilisation de k-mers ne procèdent pas à l'assemblage des lectures, mais à l'assemblage des k-mers. Les lectures ayant généré les k-mers sont en quelque sorte *oubliées* le temps de l'assemblage.

Comme présenté dans l'exemple Figure 1.5 page ci-contre, les k-mers sont organisés dans un graphe appelé graphe de *de Bruijn** que nous nommerons dBG dans la suite de ce document. Une

FIGURE 1.5 – Exemple de k-dBG avec $k = 5$. Dans cette exemple jouet, deux lectures sont à assembler : $ACGTTGCGT$ et $GTTGCGTAA$. Les 5-mers issus de ces 2 lectures sont $ACGTT, CGTTG, GTTGC, TTGCG, TGCGT, GCGTA$, et $CGTAA$. La lecture du chemin du dBG permet de reconstruire la séquence $ACGTTGCGTAA$ et donc d'assembler implicitement les deux lectures.

fois le dBG construit, la lecture du ou des chemins qui le constituent génère les séquences assemblées. Lors de la lecture de n noeuds consécutifs, le k-mer du premier noeud est lu dans son intégralité. Lors de la lecture des k-mers des noeuds suivants, seul le dernier caractère est lu.

Dans les sections suivantes nous définissons formellement les concepts utilisés et nous étudierons plus en détails les méthodes associées à l'assemblage par dBG.

Graphe de *de Bruijn*

Definition 1.3 (graphe de *de Bruijn* (dBG))**.** Étant donné un ensemble de séquences (lectures) $S = \{r_1, r_2, \ldots, r_n\}$ sur un alphabet Σ et une valeur entière $k \geq 2$, le graphe de *de Bruijn* pour S est un graphe dirigé (V,E) tel que

- $V = \{d \in \Sigma^k \mid \exists i \in [1, n]$ tel que d est une sous-séquence de $r_i\}$
- $E = \{(d_i, d_j)$: si le suffixe de taille $k-1$ de d_i est un préfixe de $d_j\}$

* *

*

Supposons que nous disposions pour une séquence de taille n d'un ensemble de lectures telles que :

1. les lectures soient exemptes de toute erreur de séquençage ;
2. les lectures soient toutes séquencées dans le sens *forward* (ou toutes séquencées dans le sens *reverse*)
3. la séquence à assembler soit exempte de toute répétition de longueur $\geq k-1$;
4. les lectures soient de taille $\geq k$;
5. les lectures successives se chevauchent sur au moins $k-1$ caractères ;

Alors dans ce cas, le k-dBG associé est composé d'un unique chemin simple contenant exactement $n-k+1$ noeuds. La lecture de ce chemin permet de reconstruire exactement la séquence de taille n recherchée.

Les deux dernières conditions sont généralement remplies par un choix judicieux de k (généralement de l'ordre de 30 à 40 pour des données usuelles Illumina). La première condition n'est généralement pas remplie totalement malgré une étape de correction des données (voir Section "*Correction de lectures et comptage de k-mers*" page 12). Les erreurs restantes dans le dBG génèrent des chemins parallèles qui peuvent perturber les assemblages. La seconde condition n'est pas non plus limitante comme nous l'expliquerons dans la suite. Enfin la troisième condition concernant la longueur des

répétitions présentes dans le génome à assembler, constitue elle la véritable et principale limitation aux techniques d'assemblage par k-mers. Nous tenterons de proposer une nouvelle solution algorithmique à ce problème dans le Chapitre 5 page 75.

Orientation des lectures Comme nous l'avons déjà évoqué, l'orientation des lectures n'est pas connues lors de l'assemblage. Deux solutions sont envisageables pour palier à cette inconnue. La première consiste à considérer chaque lecture dans les deux sens possibles : le sens dans lequel elle a été séquencée, et son *reverse complement*. Ceci conduit inévitablement à doubler la quantité de données à manipuler, ce qui n'est évidemment pas souhaitable. La seconde solution, qui est généralement celle utilisée consiste pour chaque k-mer à ne stocker qu'une des deux versions (*forward* ou *reverse complement*) de celui-ci, par exemple la plus petite lexicographiquement. Dans la suite de ce document, la version lexicographiquement la plus petite entre un k-mer et son *reverse complement* est appelée la version *canonique* de ce k-mer. Ainsi la présence d'un k-mer indique soit la présence de ce k-mer soit de son *reverse complement* soit des deux.

Représentation du dBG Un avantage majeur du dBG réside dans le fait qu'il n'est pas indispensable de le représenter de manière explicite à l'aide d'une structure de type noeuds/arêtes. En effet, les arêtes sont implicitement définies par le contenu des noeuds. Ainsi un ensemble de k-mers défini implicitement un dBG. Étant donné un dBG représenté par ensemble de k-mers canoniques, pour passer d'un noeud à son successeur, il suffit de requêter la présence de ses quatre voisins potentiels dans l'ensemble des k-mers.

k-mer	Version canonique
$ACGTT$	$AACGT$
$CGTTG$	$CAACG$
$GTTGC$	$GCAAC$
$TTGCG$	$CGCAA$
$TGCGT$	$ACGCA$
$GCGTA$	$GCGTA$
$CGTAA$	$CGTAA$

Table 1.1 – Ensemble des k-mers présents dans les lectures $ACGTTGCGT$ et $GTTGCGTAA$ et leurs représentations canoniques.

Exemple de parcours du dBG représenté implicitement par les k-mers canoniques des lectures En nous basant sur les deux lectures de l'exemple présenté Figure 1.5 page précédente, supposons que nous ne stockions que les versions canoniques de l'ensemble des k-mers issus de ces lectures. Comme présenté Table 1.1, il s'agit des k-mers $AACGT$, $CAACG$, $GCAAC$, $CGCAA$, $ACGCA$, $GCGTA$, et $CGTAA$. Supposons également que nous ne dispositions que de cet ensemble de k-mers canoniques pour toute représentation implicite du dBG. Pour lire ce graphe en partant du k-mer $ACGTT$ comme c'était le cas dans l'exemple présenté Figure 1.5 page précédente, il suffit de requêter ses 4 voisins potentiels $CGTTA$, $CGTTC$, $CGTTG$, et $CGTTT$ dont les représentations canoniques sont respectivement $CGTTA$, $CGTTC$, $CAACG$, et $AAACG$. Nous constatons que le

k-mer $CAACG$ est effectivement présent dans les données. N'oublions pas que nous avons requêté $CAACG$ dans le but de savoir si le k-mer $CGTTG$ était présent. À présent, si nous souhaitons poursuivre l'exploration du graphe, il est donc nécessaire de tester les quatre voisins potentiels de $CGTTG$, à savoir $GTTGA$, $GTTGC$, $GTTGG$, et $CTTGT$, dont les représentants canoniques sont $GTTGA$, $GCAAC$, $CCAAC$, et $ACAAG$, et ainsi de suite.

Unitigs et contigs

Le parcours du dBG permet de générer des séquences présentes dans la séquence à assembler. Deux types de parcours sont à distinguer, générant respectivement ce que l'on nome *unitigs** et *contigs*.

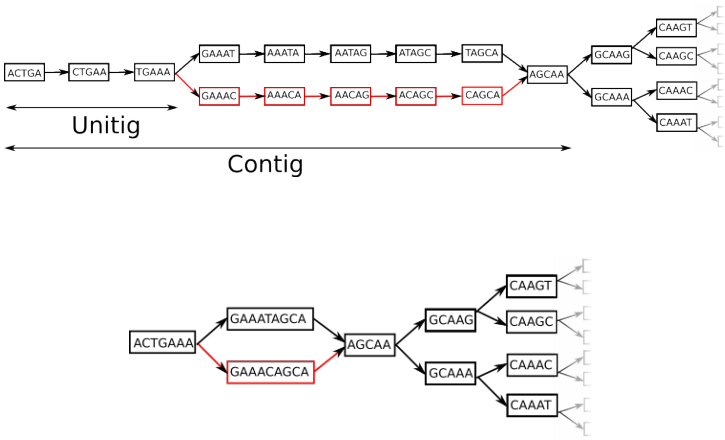


FIGURE 1.6 – Représentation d'un dBG (haut) et d'un CdBG (bas). **Haut** : l'unitig représenté est $ACTGAAA$. En supposant que l'assembleur ait choisit le chemin du bas (en rouge), alors le contig représenté est $ACTGAAACAGCAA$. Notons que dans cet exemple, le contig est stoppé au k-mer AGCAA car la suite du graphe est trop complexe pour y effecturer un choix pertinent. **Bas** : version compactée du dBG, chaque noeud contient un unitig.

Unitigs Le degré entrant d'un noeud u du dBG indique le nombre de k-mers du dBG ayant un suffixe de taille $k-1$ égale au préfixe de taille $k-1$ du k-mer stocké dans le noeud u. Ce degré est entre zéro et quatre. Inversement le degré sortant d'un noeud u, également entre zéro et quatre, indique le nombre de k-mers du dBG ayant un préfixe de taille $k-1$ égale au suffixe de taille $k-1$ du k-mer stocké dans le noeud u. Dans l'exemple donné Figure 1.6, le noeud correspondant au k-mer $AGCAA$ a un degré entrant égal à deux et un degré sortant aussi égal à deux.

On appelle un noeud branchant un noeud du dBG ayant un degré entrant ≥ 2 et/ou un degré sortant ≥ 2. Un chemin de taille maximale, pour lequel aucun noeud branchant n'a été traversé est appelé un *unitig*. Un exemple d'unitig est indiqué sur le graphe représenté Figure 1.6. Un unitig peut également être défini par la séquence contenue dans les noeuds d'un graphe de *de Bruijn* compacté, défini comme suit :

Definition 1.4 (graphe de *de Bruijn* compacté (CdBG))**.** Étant donnés deux noeuds u et v d'un dBG tels que u est l'unique prédécesseur de v et tel que v est l'unique successeur de u, alors les noeuds u et v peuvent être compactés en un unique noeud dont la séquence résulte de la concaténation des séquences de u et v après suppression du préfixe de taille $k-1$ de v. Le noeud résultant contient non plus un k-mer mais une séquence de taille $> k$. La compaction s'applique de la même manière à deux noeuds déjà eux-mêmes compactés ou à un noeud compacté avec un noeud non compacté.

Un dBG pour lequel toutes les compactions possibles sont effectuées est appelé un graphe de *de Bruijn* compacté, noté CdBG dans ce document.

Un exemple de dBG et du CdBG associé est présenté Figure 1.6 page précédente.

Contigs. Lors du parcours d'un dBG ou CdBG, des chemins traversant des branchements peuvent être lus. Dans le contexte de l'assemblage, les choix effectués lors de la traversée de noeuds branchants dépendent essentiellement de l'assembleur et de ses paramètres. En quelques mots, si les branchements sont considérés comme étant dus à des variations ponctuelles telles des SNPs (voir Chapitre 2 page 15) ou des erreurs de séquençage (substitutions, petites insertions ou délétions), alors un chemin est privilégié et c'est celui-ci qui est fourni comme résultat d'assemblage. Si les branchements deviennent trop importants (selon divers critères que nous ne détaillerons pas ici), alors le parcours s'arrête. Les séquences issues de ce type de parcours traversant des noeuds branchants sont appelées des *contigs*.

Correction de lectures et comptage de k-mers

Comme nous l'avons déjà évoqué, les données de séquençage contiennent des erreurs de séquençage. Ainsi des variations peuvent exister entre deux lectures issues du même locus d'un génome. Deux techniques distinctes peuvent être mises en oeuvre. La première corrige ces données. Elle consiste à aligner les lectures les unes par rapport aux autres et à détecter les positions divergentes minoritaires ou sous représentées et à les remplacer par la version de la séquence majoritaire. Aligner les lectures les unes par rapport aux autres est très couteux. Ainsi, l'application de cette technique se limite aux données composées de peu de lectures.

La seconde technique ne permet pas de corriger les lectures, mais de supprimer les k-mers supposés contenir au moins une erreur de séquençage. L'idée est assez simple. Supposons que la couverture attendue en reads soit connue et égale à C. Avec des lectures de taille L, chaque lecture contient $L-k+1$ k-mers. Ainsi chaque k-mer devrait apparaître en moyenne $C \times \frac{L-k+1}{L}$ fois. Des valeurs $C = 50$, $L = 100$ et $k = 31$ sont assez classiques. Ainsi on peut considérer dans ce cas que la couverture en k-mers devrait être aux alentours de 35. Inversement, une vision pessimiste des données de type Illumina indique en moyenne une erreur de type substitution par lecture. Supposons pour simplifier qu'une erreur de séquençage n'apparait pas deux fois au même endroit dans deux lectures distinctes (ce qui est vraisemblable avec des couvertures limitées à quelques dizaines). Alors les k-mers affectés par une erreur de séquençage ont une très forte probabilité de n'apparaître qu'une fois dans les données. Aussi, pour simplifier à l'extrême, il devient assez simple de séparer les k-mers vus une fois de ceux vus 35 fois.

Pour généraliser, les k-mers de très faible couverture par rapport à la couverture attendue, sont probablement dus à des erreurs de séquençage et sont supprimés des jeux de données. Les k-mers

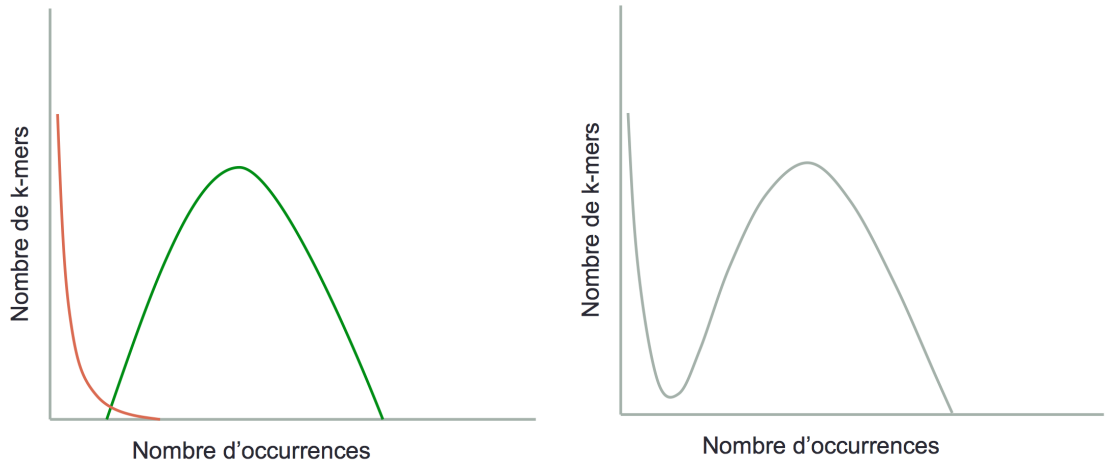


FIGURE 1.7 – Spectres de k-mers théoriques. Gauche : visualisation en séparant les k-mers erronés (rouge) des k-mers solides (vert). Droite : spectre de k-mer théorique sans séparation des k-mers erronés et solides. La position de la *vallée* entre les deux distributions permet de fixer un seuil pour séparer les k-mers solides des k-mers erronés.

restants, considérés comme non erronés sont dits *solides* et sont conservés lors de la construction des dBG.

Pour des données génomiques, comme présenté Figure 1.7, la distinction entre k-mers solides et k-mers erronés se fait en analysant le spectre de k-mers, qui indique le nombre de k-mers (en ordonnée) ayant une couverture donnée (en abscisse). Dans la pratique, comme en témoigne la Figure 1.8 page suivante, la distinction entre k-mers erronés et k-mers solides n'est pas toujours si franche. L'outil KmerGenie [Chikhi and Medvedev, 2014] analyse ce type de spectre pour y détecter la meilleure valeur seuil de séparation des k-mers erronés des k-mers solides.

Dans des données d'expression, comme le séquençage d'ARN (appelé RNA-seq), la couverture reflète le degré d'expression des gènes. Dans ce type de situation, un k-mer peu couvert peut résulter d'une erreur de séquençage, mais aussi de gènes faiblement exprimés. Ainsi, dans des données RNA-seq, associer des k-mers peu couverts à des erreurs de séquençage conduit à supprimer des k-mers solides. D'autres méthodes, basées sur des variations locales de couverture de k-mers, sont mises en place, comme nous l'évoquerons dans le chapitre suivant.

Quoi qu'il en soit, la première étape d'analyse de données NGS utilisant une approche basée sur des k-mers est de compter ceux-ci. Ici encore, même si une telle étape peut paraitre triviale, son application sur des dizaines de milliards voire centaines de milliards de k-mers nécessite des techniques d'algorithmique fines utilisant au mieux toutes les ressources matérielles disponibles (multi-coeurs, disques, mémoire RAM, ...). Les outils les plus usités pour effectuer cette tâche sont DSK [Rizk et al., 2013], Jellyfish [Marçais and Kingsford, 2011], ou encore KMC 2 [Deorowicz et al., 2014].

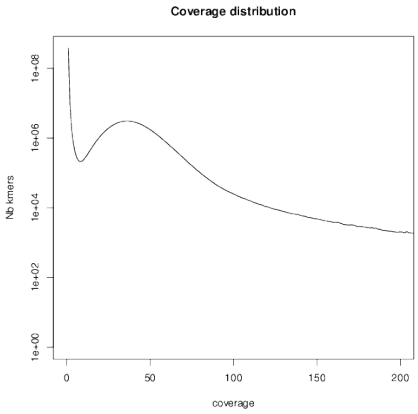


FIGURE 1.8 – Spectre de k-mers obtenu à partir de lectures réelles issues de *C. Elegans*. Le jeu de données (SRR065390) est composé de 33808546 lectures, chacune de taille 200. Données obtenues à partir de l'outil DSK [Rizk et al., 2013].

1.5 Pour résumer

Dans ce chapitre nous avons survolé les sujets relatifs à l'apparition des nouvelles techniques de séquençage. Nous avons évoqué les diverses possibilités permettant leur assemblage, ce qui nous a permis d'introduire la notion de k-mer et de graphe de *de Bruijn* qui nous seront utiles tout au long de ce document.

Chapitre 2

Détection de variants

Contents

2.1 Introduction 15
2.2 Présentation de quelques variants 16
2.3 Estimer la qualité de la détection de variants 19
2.4 Détection de variants par mapping sur séquence de référence 20
2.5 Modèles pour la détection de variants *de novo* 23
2.6 Mise en oeuvre algorithmique 28
2.7 Difficultés et solutions 29
2.8 Résultats 31
2.9 Perspectives 36
2.10 Présentation des publications associées 38

Ce chapitre présente de nouvelles approches pour la détection de variants dans les données génomiques et transcriptomiques.

2.1 Introduction

Connaitre les variants dans les données génomiques ou transcriptomiques est une source majeure d'information dans tous les domaines qui manipulent ce type de données. Connaitre un génome, ses spécificités, ou encore sa structure est une connaissance primordiale. Cependant, cette connaissance ne permet pas de déterminer précisément les éléments mis en cause dans des maladies ou de déterminer des traits phénotypiques d'intérêt comme en agronomie ou en environnement par exemple.

À l'inverse, la connaissance de variants permet d'y associer des différences phénotypiques et donc de comprendre le rôle de certains gènes ou d'associations de gènes, et de comprendre certains phénomènes évolutifs comme l'adaptation [Wright, 1949]. De plus, des variants aussi simples que des SNPs (voir plus bas) ont un vaste horizon d'applications. Ils peuvent être utilisés de manière localisée, un SNP ou quelques SNPs ayant un rôle prépondérant dans une maladie [Kim et al., 2012] ou un caractère de résistance [Barrett et al., 2012] par exemple. Inversement ils peuvent être utilisés de manière globale, afin par exemple de *cartographier** des génomes [Collard et al., 2005].

Il existe plusieurs types de variants génomiques et transcriptomiques. Ceux présentés dans ce documents sont ceux pour lesquels nous avons développé de nouvelles méthodes pour leur détection. Les choix effectués ici reflètent deux choses : d'une part l'utilité de la connaissance de ces variants et donc l'existence de méthodes d'analyses en post-traitement et, d'autre part, la simplicité de détection de tels variants.

2.2 Présentation de quelques variants

2.2.1 Les SNP (*Single Nucleotide Polymorphism*)

Cas 1

Individu1

...ACGGCGAGCGATCGCAGCAG**C**TACACACGCTATCGTAGCTG...

Individu2

...ACGGCGAGCGATCGCAGCAG**G**TACACACGCTATCGTAGCTG...

Cas 2

Individu1 - chromosome 1

...ACGTCAGGCAGGCTTATGCG**T**AACAACGGCATCAGATAGCTG...

Individu1 - homologue chromosome 1

...ACGTCAGGCAGGCTTATGCG**A**AACAACGGCATCAGATAGCTG...

Cas 3

Individu1

...TACCG**C**CAAA...//...TACCG**G**CAAA...

FIGURE 2.1 – Représentation de SNPs (cas 1 et 2), à ne pas confondre avec du polymorphisme résultant de répétitions inexactes (cas 3).

Un SNP (*Single Nucleotide Polymorphism*) est une mutation ponctuelle ne modifiant qu'un nucléotide entre les génomes de deux individus d'une même espèce. Le cas 1 présenté Figure 2.1 représente un tel SNP.

Rappelons que pour certaines espèces, les chromosomes d'une cellule sont tous différents. De tels génomes (et par extension de telles espèces) sont dits *haploïdes*. Pour certaines espèces ils sont présents par paires (génomes ou espèces *diploïdes*). Dans ce cas chaque chromosome non sexuel (appelés *autosomes*) a un chromosome homologue. Pour d'autres espèces les chromosomes sont présents en plus de deux copies (génomes ou espèces *polyploïdes*). Une mutation ponctuelle d'un nucléotide situé à la même localisation génomique (*locus*) entre deux chromosomes homologues d'un individu diploïde ou polyploïde est également appelée un SNP, comme représenté dans le cas 2 de la Figure 2.1.

Il est nécessaire de faire la distinction entre un SNP et un polymorphisme ponctuel dû à une *répétition inexacte*. Il existe en effet de nombreuses répétitions dans les génomes. Des portions génomiques peuvent être dupliquées et insérées à différents locus du génome. Les régions répétées évoluent indépendamment, et divergent petit à petit, au grès des mutations et de la pression de sélection [Wright, 1949]. Concrètement, il en résulte que les génomes peuvent contenir des portions

de séquences identiques à une ou quelques substitution(s) près, mais qui ne sont pas dues à des SNPs, mais à des répétitions inexactes, comme représenté dans le cas 3 de la Figure 2.1 page précédente.

2.2.2 Les insertions et délétions (indels)

Les insertions et les délétions sont des variants modifiant la quantité de nucléotides de la séquence affectée. L'ajout d'un ou de plusieurs nucléotides à une séquence donnée est naturellement appelé une insertion, et une délétion indique la suppression d'un ou plusieurs nucléotides d'une séquence.

```
...ACGGCGAGCGATCGCAGCAG---TACACACGCTATCGTAGCTG...

...ACGGCGAGCGATCGCAGCAGGATTACACACGCTATCGTAGCTG...
```

FIGURE 2.2 – Exemple d'indel

À l'image des SNPs, les indels sont situés sur des séquences homologues intra ou inter génomes (respectivement situées à la même localisation de chromosomes homologues d'un même individu ou situées à la même localisation de génomes d'individus distincts). Il convient donc également de les différencier d'insertions ou de délétions situées à des loci distincts intra génomiques, reflétant alors la présence de répétitions inexactes.

Le terme *indel* désigne soit une insertion soit une délétion. L'existence de ce terme traduit notre ignorance, lors de la détection de ce type d'évènement, de l'histoire associée à la variation. À l'image de l'indel présentée Figure 2.2, nous ne sommes pas en mesure, étant données ces deux séquences d'indiquer s'il s'agit de la séquence du haut qui a perdu trois nucléotides ou s'il s'agit de la séquence du bas qui en gagné trois.

Les *microindels* [Gonzalez et al., 2007] désignent des indels dont la taille est comprise entre un et 50 nucléotides. Dans les régions codantes du génome, les indels dont la taille n'est pas un multiple de trois résultent dans l'apparition d'un *frameshift*. Lors de la traduction de l'ARN en protéines (cf Figure 1.1 page 2), les nucléotides sont traduites par groupes de trois. Chaque triplet de nucléotide est à l'origine d'un acide aminé, constituant de la protéine. Un indel ω de taille $|\omega| = 3p$ ($p \in \mathbb{N}$) insère ou supprime p acides aminés dans la protéine synthétisée. Inversement, un indel de taille $|\omega| \neq 3p$ génère un décalage de cadre de lecture (*frameshift*), pouvant potentiellement modifier l'intégralité de la composition de la protéine en acides aminés.

Les microindels représentent un type de mutations qui joue un rôle important dans les maladies génétiques [Ball et al., 2005] et représentent également une source d'information pour la reconstruction de *phylogénies** [Snir and Pachter, 2006, 2011]. Leur détection est donc essentielle.

2.2.3 Les inversions

Une inversion est un réarrangement chromosomique dans lequel un segment contigu de chromosome est inversé de bout en bout et réintroduit au même locus de ce chromosome. Il a été reconnu que ce type d'inversions est impliqué par exemple dans l'adaptation des espèces [Hoffmann et al., 2004] ou l'évolution de chromosomes sexuels [Van Doorn and Kirkpatrick, 2007].

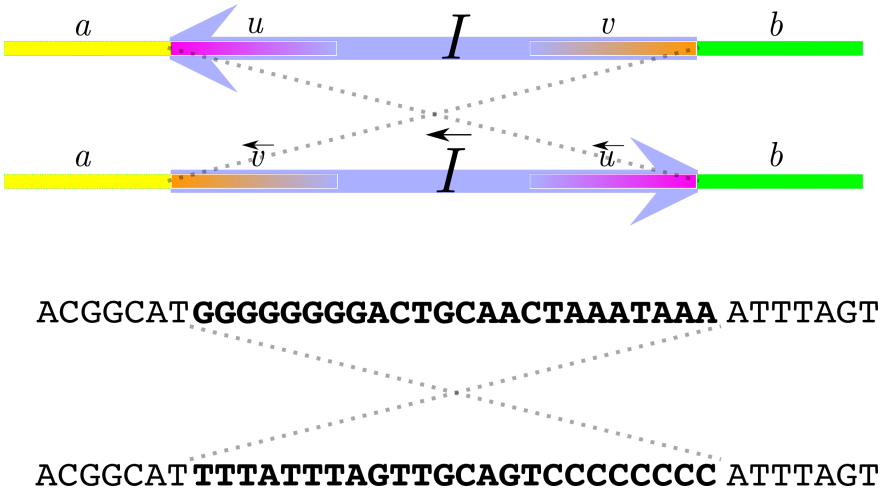


FIGURE 2.3 – Représentation graphique (haut, tiré de [Lemaitre et al., 2014]) et textuelle (bas) d'une inversion. Dans l'exemple du bas, la séquence inversée est $I = GGGGGGGGACTGCAACTAAATAAA$ (et donc $\overleftarrow{I} = TTTATTTAGTTGCAGTCCCCCCCC$)

2.2.4 L'épissage alternatif

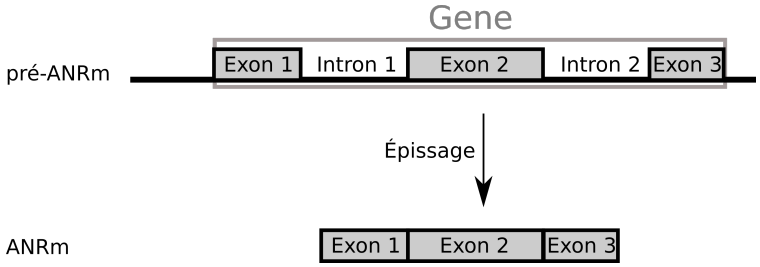


FIGURE 2.4 – Représentation de l'épissage. Un gène est composé de trois exons, et donc de deux introns. Lors de l'épissage les introns sont éliminés

L'ARN pré-messager (pré-ARNm) est synthétisé à partir du brin matrice de l'ADN dans le noyau lors de la transcription. L'ARN pré-messager est constitué de deux types de segments : les exons et les introns. Les exons sont conservés dans l'ARN final appelé ARN mature (ARNm), tandis que les introns sont excisés lors du processus appelé *l'épissage** (voir Figure 2.4).

Durant l'épissage de l'ARN, comme représenté Figure 2.5 page suivante, les introns ne sont pas tous systématiquement éliminés dans leur intégralité. L'épissage peut suivre diverses combinaisons qui conduisent chacune à la création d'ARNm distincts. Ce processus s'appelle l'épissage alternatif. Chez l'humain entre 40 et 60% des gènes sont sujets à l'épissage alternatif [Modrek and Lee, 2003].

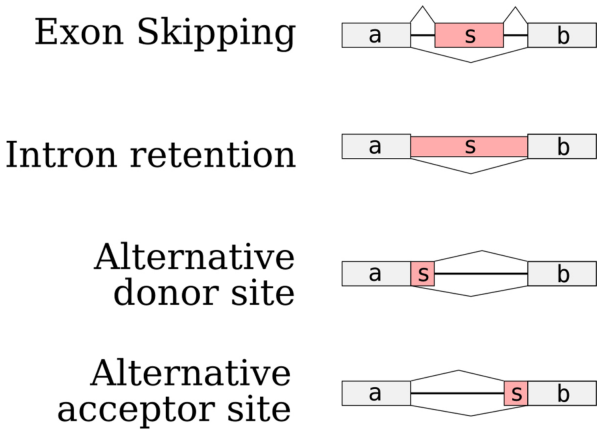


FIGURE 2.5 – Impage importée de [Sacomoto et al., 2012]. Représentation de différents scénarios d'épissage alternatif. L'"exon skipping" indique qu'un exon peut être entièrement épissé. L'"intron retention" indique qu'un intron est entièrement conservé dans l'ARNm. Les cas "alternative donor site" et "alternative acceptor site" indiquent une coupure exon/intron (respectivement une coupure intron/exon) alternative.

2.3 Estimer la qualité de la détection de variants

Les méthodes de détection de variants ne sont ni exacte ni exhaustives. Ainsi nous utilisons également le terme "prédiction" de variants afin de tenir compte de cette inexactitude.

Voici le formalisme utilisé afin d'estimer la qualité d'un résultat de prédiction de variants. Supposons que les données contiennent un ensemble V_{ref} de variants, et supposons qu'une méthode de prédiction de variants ait détecté un ensemble V_{predit} de variants. Pour chaque élément de V_{predit} il s'agit :

- soit d'un vrai positif (noté VP dans ce document) s'il appartient à l'ensemble V_{ref} ;
- soit d'un faux positif sinon, noté FP. Ce sont les éléments détectés à tort.

Inversement, pour chaque élément de V_{ref}, il s'agit :

- soit d'un VP s'il appartient à V_{predit} ;
- soit d'un faux négatif sinon, noté FN. Ce sont les éléments ratés par la méthode de prédiction.

Notons, même si cela n'est pas indispensable pour la suite, que les éléments n'appartenant ni à V_{ref} ni à V_{predit} sont appelés des vrai négatifs (VN). Ce sont les éléments non détectés, à raison. Par abus de langage on note $|VP|$, $|FP|$, $|FN|$ le nombre de vrais positifs, de faux positifs et de faux négatifs, respectivement.

À partir de ces métriques, nous définissons :

- Le *recall** de la méthode. Il s'agit de la faculté de la méthode de détecter les éléments de V_{ref}. On définit le recall comme $\frac{|VP|}{|V_{ref}|} = \frac{|VP|}{|VP|+|FN|}$
- La *précision** de la méthode. Il s'agit de mesurer la fiabilité des prédictions. On définit la précision comme $\frac{|VP|}{|V_{predit}|} = \frac{|VP|}{|VP|+|FP|}$

2.4 Détection de variants par mapping sur séquence de référence

Dans le contexte de ce document, nous considérons que la détection de variants s'applique lorsque l'on cherche à détecter des variants dans des données NGS génomiques ou RNA-seq, c'est à dire, rappelons-le, composées d'un grand ensemble de courtes séquences (lectures) dont le locus et l'orientation sont inconnus pour chacune d'entre elles.

Les méthodes usuelles de détection de variants sont basées sur l'analyse des différences observées entre les données de séquençage et une séquence de référence. En deux mots, ce type d'analyse comporte deux étapes principales. La première consiste à aligner chacune des lectures sur la séquence de référence. Il s'agit de *mapping*. Les outils utilisés pour cette étape sont appelés les mappeurs (parmi les plus connus et utilisés, nous pouvons citer BWA [Li and Durbin, 2009] ou BOWTIE [Langmead and Salzberg, 2012a] pour le mapping de données génomiques ou STAR [Dobin et al., 2013] pour le mapping des données transcriptomiques). Ils consistent à déterminer pour chaque lecture la position sur la référence dont elle est le plus probablement issue. Lors d'une seconde étape, appelée le *calling*, les différences ponctuelles ou structurelles observées entre les lectures et la ou les portions de séquences sur lesquelles elles s'alignent sont analysées afin de prédire les variants recherchés. Les variants peuvent ainsi être prédits, par exemple via l'outil GATK [McKenna et al., 2010] entre un jeu de lectures et une séquence de référence, ou entre deux jeux de lectures ou plus en s'appuyant sur la séquence de référence.

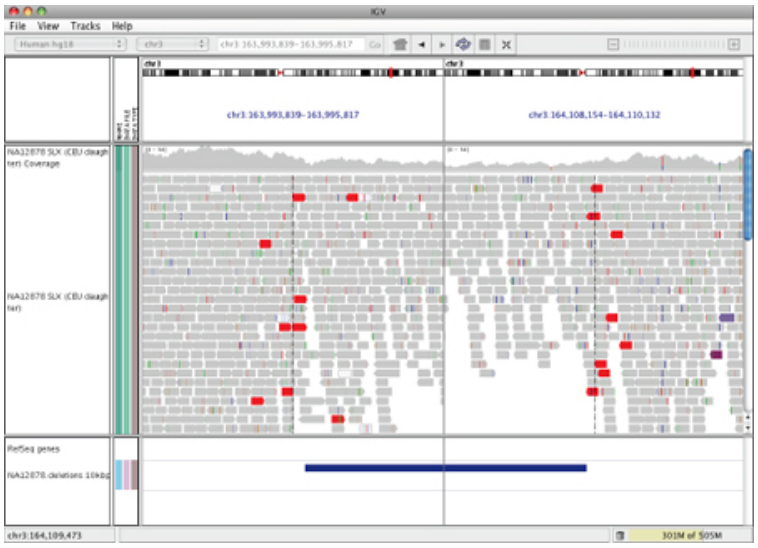


FIGURE 2.6 – Visualisation de variants (ici des délétions) suite au mapping de lectures sur une référence, via l'outil de visualisation IGV [Thorvaldsdóttir et al., 2012]

La détection de variants par mapping sur une séquence de référence présente de nombreux avantages. Comme présenté Figure 2.6, elle permet par exemple une visualisation intuitive des variants détectés. Elle a aussi l'avantage de permettre une localisation des variants détectés le long du génome de référence. Elle a été particulièrement utilisée dans de très gros projets comme le "*1000 Genomes Project*" [Altshuler et al., 2012] par exemple.

La détection de variants est une étape clef pour l'analyse de données de séquençage. Cette étape doit donc être :

- la plus rapide possible ;
- la moins gourmande en mémoire possible ;
- la plus exhaustive possible (un recall proche de 1) et la plus précise possible (une précision proche de 1).

Les méthodes de détection de variants par mapping sur une séquence de référence connaissent des limitations qui ont motivé les travaux présentés dans ce chapitre.

Temps d'exécution Les deux étapes clefs des méthodes de détection de variants par mapping sur séquence de référence ne sont pas particulièrement problématiques du point de vue du temps de calcul. Il ne s'agit pas d'une réelle limite dans l'utilisation de ce type de données. Plusieurs millions de lectures peuvent être analysées en quelques heures sur des machines performantes. Cependant, il va de soi que gagner un ou plusieurs ordres de grandeur sur les temps d'exécution lors de la détection ouvre les portes à plus d'analyses et limite la consommation énergétique.

Empreinte mémoire L'une des limitations des méthodes de détection de variants par mapping sur séquence de référence tient dans le fait que la redondance d'information des lectures n'est pas réduite. Pour chacune d'entre elles, il faut conserver l'information de sa position de mapping et de ses variations avec la séquence de référence. La détection des variants se fait alors sur la base de l'intégralité de ces informations de mapping, qui doivent donc toutes être connues simultanément. Ainsi, détecter des variants via ces approches nécessite l'emploi de machines disposant d'une grande quantité de mémoire RAM, de plusieurs centaines ou milliers de GB (giga bytes ou giga octets). Ceci représente une limitation de taille pour la plupart des laboratoires de biologie qui n'ont pas nécessairement d'accès évident et illimité à de grosses ressources de calcul.

Répétitions Un problème inhérent à la détection de variants est lié à la présence de répétitions dans les données. Dans le cas du calling par mapping, ceci se traduit par le fait que les lectures issues des régions répétées

- soit mappent à différents locus de la séquence de référence si celle-ci est de bonne qualité et que les répétitions ont correctement été assemblées et que leurs occurrences ont été séparées,
- soit mappent toutes au même locus de la séquence de référence si les différentes occurrences des répétitions ont été fusionnées lors de l'assemblage.

Dans le premier cas, il est difficile d'établir clairement la *vraie* position de mapping d'une lecture. Dans le second cas, la couverture locale (nombre de lectures mappées localement) est surestimée. Dans les deux cas, la précision et le recall des variants situés dans des séquences répétées sont sous ou sur-estimés.

Complexité d'utilisation La détection de variants par mapping sur une séquence de référence implique plusieurs étapes. Le mapping et le calling en sont les deux principales, mais elles impliquent également d'autres méthodes annexes (nettoyage des données, réalignement des lectures, normalisation des couvertures, ...). La Figure 2.7 page suivante, issue de la documentation de GATK,

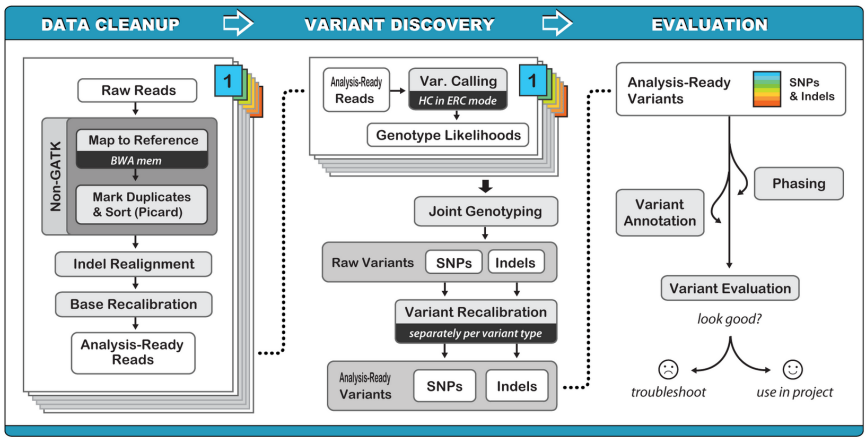


FIGURE 2.7 – Image issue de la documentation de GATK [McKenna et al., 2010]

témoigne de la complexité associée à l'utilisation de ce type d'approche. En outre, chaque méthode implique un lot de paramètres et d'options qui influent sur le résultat final.

Utilisation d'une séquence de référence La limitation la plus évidente de la détection de variants par mapping sur une séquence de référence est qu'elle nécessite l'utilisation d'une séquence de référence de bonne qualité. La présence d'une séquence de référence issue d'une espèce ou d'une souche distante et/ou de mauvaise qualité :

- augmente le nombre de lectures non mappées et donc diminue le recall ;
- augmente le nombre de lectures mal mappées et donc diminue la précision.

Il convient de replacer cette limitation dans un contexte où, malgré les progrès drastiques faits en matière de séquençage, le nombre d'espèces pour lesquelles il existe un génome de référence publiquement disponible et correctement assemblé reste restreint. En dehors de quelques espèces modèles pour lesquelles de très gros efforts de séquençage et d'assemblage ont été effectués (humain et quelques primates, rat, souris, *C.Elegans*), il existe peu de ressources fiables et exploitables. Or, l'un des effets de la "démocratisation" des NGS est que les biologistes travaillent de plus en plus sur des espèces non modèles pour lesquelles les ressources existantes sont moindres, voire inexistantes.

* *

*

Pour toutes ces raisons, il nous est apparu nécessaire de proposer de nouvelles approches permettant de répondre aux besoins de détection de variants dans les données NGS, en l'absence de génome de référence. En outre, les méthodes que nous proposons permettent d'outrepasser certaines des limitations présentées précédemment.

2.5 Modèles pour la détection de variants *de novo*

Afin de pouvoir se passer de génome de référence et espérer obtenir de meilleurs résultats en consommant moins de ressources, nous proposons des approches utilisant uniquement des données NGS. C'est ce que l'on appelle la détection *de novo* de variants.

À l'image de l'assemblage, deux possibilités sont envisageables pour la détection de novo de variants. Une possibilité consiste à comparer les lectures entre elles pour y déceler les témoins de la présence de variants, alors qu'une seconde possibilité consiste à utiliser les k-mers présents dans les lectures pour détecter les variants. Nous avons choisi la seconde approche pour des raisons similaires à celles exposées Section 1.4.1 page 7 : la comparaison de l'ensemble des lectures est trop consommatrice de ressources temps et mémoire.

Ainsi, toutes les approches présentées dans ce chapitre sont basées sur l'utilisation du dBG. L'idée fondamentale se résume ainsi : le dBG associé à un ensemble de lectures contient des motifs topologiques témoignant de la présence de variants dans les données. Ainsi, l'idée de base de la détection de variants de novo consiste à 1/ identifier les motifs associés aux variants recherchés ; 2/ mettre en place un cadre algorithmique efficace pour leur détection ; 3/ analyser plus en détails et classer les variants détectés.

Commençons par présenter les motifs topologiques associés dans le dBG à divers variants d'intérêt. Par abus de langage dans la suite du document nous associerons un dBG aux données qui ont servi à générer ce dBG. Par exemple, un SNP dans un dBG, signifie que les données qui ont servi à construire le dBG contiennent un SNP.

2.5.1 Motifs associés aux SNPs

SNP Isolé

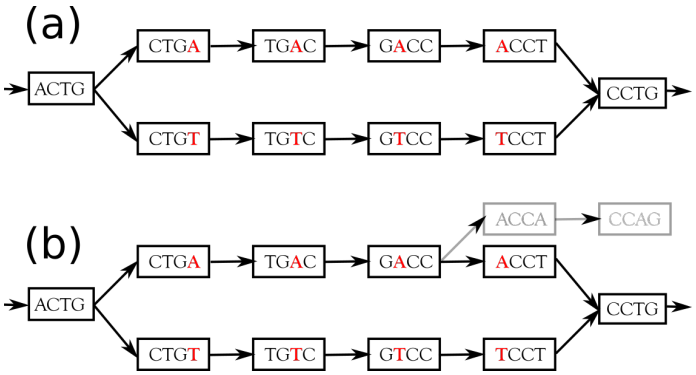


FIGURE 2.8 – Représentation de *bubbles* associées à la présence d'un SNP dans un dBG. **(a)** Dans cet exemple, $k = 4$ et les données contiennent les séquences ACTGACCTG et ACTGTCCTG. Les deux chemins distincts sont chacun composés d'exactement k noeuds. La bubble représentée en **(b)**, est identique à la première à la différence près qu'elle est dite *branchante* du fait de l'existence du noeud ACCA.

Nous définissons un SNP isolé comme un SNP *suffisamment éloigné* de toute autre source de

polymorphisme dans les données. Concrètement, pour qu'il soit considéré comme isolé, la distance (en terme de nombre de nucléotides) minimale d'un SNP à toute autre source de polymorphisme est de k nucléotides. Ainsi, la notion d'isolement pour un SNP dépend de k et n'est donc pas un critère biologique. La notion de SNP isolé est née de contraintes algorithmiques, mais présente d'intéressantes caractéristiques biologiques. L'unicité des séquences flanquantes gauche et droite d'un tel SNP permet la conception d'*amorces** spécifiques au SNP. Ceci est particulièrement pertinent pour utiliser un tel SNP comme marqueur à amplifier pour *génotyper** un individu par exemple.

* *

*

Dans un dBG, un SNP isolé génère un motif topologique très simple. En effet, dans les données, l'existence d'un SNP isolé est témoigné par la présence de deux séquences identiques à une substitution près, disons par exemple un A dans une séquence et un T dans l'autre. En terme de k-mers, tous les k-mers ne contenant ni ce A ni ce T sont identiques dans les deux séquences. Inversement il existe exactement k k-mers contenant le A et k k-mers contenant le T. Dans le dBG, comme représenté Figure 2.8 page précédente, ceci se traduit par la création d'une structure géométrique couramment appelée *bubble* (bulle). Le dernier k-mer commun aux deux séquences est branchant car il peut être étendu selon deux possibilités (avec un A ou un T). Deux chemins sont alors constitués chacun de k k-mers spécifiques à chaque version du SNP, avant de se *refermer* sur le premier k-mer post SNP qui soit commun aux deux séquences.

Formellement, dans le dBG, le modèle topologique associé à un SNP isolé est le suivant : un noeud branchant N_i peut être étendu vers deux noeuds distincts N_{up_1} et N_{low_1}. Ces noeuds peuvent ensuite être étendus avec le même nucléotide, respectivement vers les noeuds N_{up_2} et N_{low_2}. De nouveau, ces deux noeuds peuvent être étendus par le même nucléotide vers les noeuds N_{up_3} et N_{low_3}, et ainsi de suite jusqu'aux noeuds N_{up_k} et N_{low_k}. Enfin les deux noeuds N_{up_k} et N_{low_k} peuvent être étendus par le même nucléotide vers le noeud unique N_f. Les deux chemins générés sont chacun de taille $2k+1$.

La détection d'un motif topologique associé à un SNP isolé conduit à la détection de ce type de bubble, et donc à la détection d'un couple de séquences de taille $2k+1$, identiques à l'exception de leur nucléotide central, alors considéré comme le SNP isolé recherché.

2.5.2 SNPs proches

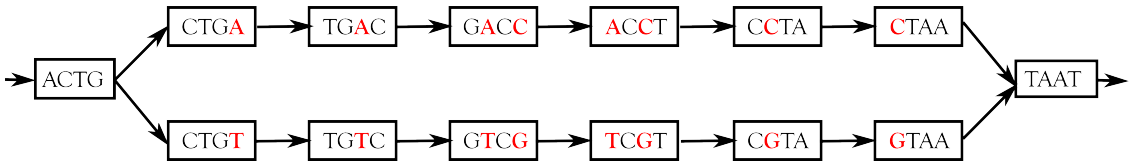


FIGURE 2.9 – Représentation d'une bubble générée par la présence de SNPs proches dans un dBG. Dans cet exemple, $k = 4$ et les données contiennent les séquences ACTGACCTAAT et ACTGTCGTAAT.

Notons que le motif topologique défini par la présence d'un SNP isolé dans les données d'un dBG impose qu'il existe au moins un couple de séquences de taille $2k+1$, identiques à une substitution centrale près. Dans le cas où, par exemple, deux SNPs se retrouvent distants de moins de k nucléotides, chacun d'entres eux ne génèrent alors pas un tel couple de séquences. Forts de cette remarque, nous pouvons définir la notion de SNP proches. En opposition aux SNPs isolés, des SNPs sont dits proches s'ils se situent à moins de k nucléotides l'un de l'autre. Dans le dBG, un couple de SNPs proches, ou la présence de n SNPs proches génèrent une bubble similaire à celle témoignant de la présence de SNPs isolés. Cependant la bubble associée aux SNPs proches contient deux chemins composés par au moins $k+1$ noeuds, générant donc deux séquences de taille au moins $2k+2$ nucléotides, telles que au moins les k premiers et derniers nucléotides sont égaux. La Figure 2.9 page ci-contre propose une représentation du motif topologique associé à des SNPs proches.

2.5.3 Indels

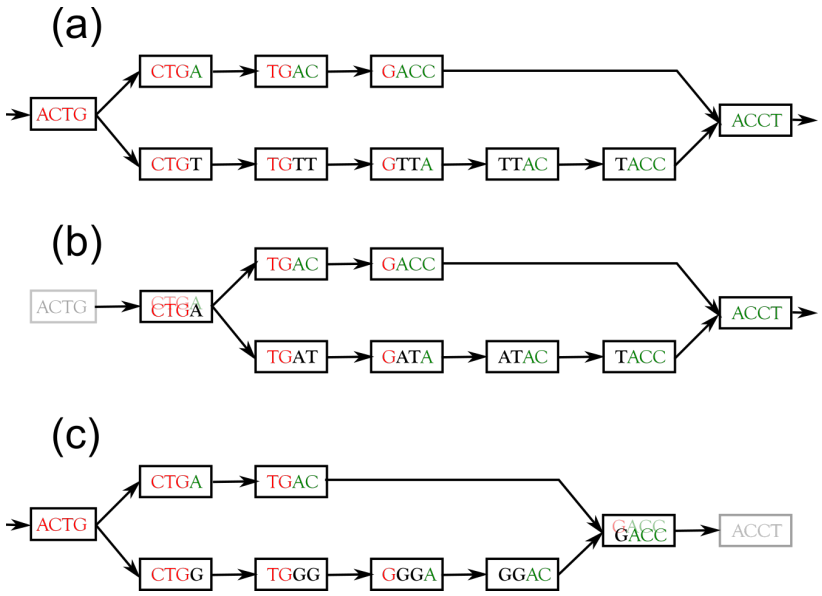


FIGURE 2.10 – Représentation de bubbles dues à la présence d'un indel dans les données, avec $k = 4$. **(a)** dans cet exemple, les données contiennent les séquences ACTGACCT et ACTGTTACCT. L'indel est donc TT. **(b)** dans cet exemple, les données contiennent les séquences ACTGACCT et ACTGATACCT. L'indel est AT. Dans ce cas, la taille du plus long préfixe commun entre l'indel (AT) et ce qui suit (ACCT) est de taille > 0 ('A'), le plus court chemin est donc composé de moins de $k-1$ noeuds. Inversement, dans **(c)**, les données contiennent les séquences ACTGACCT et ACTGGGACCT. L'indel est GG. Dans ce cas, la taille du plus long suffixe commun entre l'indel (GG) et ce qui précède (ACTG) est de taille > 0 ('G'), le plus court chemin est donc composé de moins de $k-1$ noeuds.

De manière analogue aux SNPs isolés ou proches, il est aisé de déterminer les motifs topologiques associés à des indels dans les données d'un dBG. Un indel isolé (distant d'autres sources de

polymorphisme par au moins k nulcotides) de taille d dans des données génère deux séquences, débutant par les mêmes k nucléotides et terminant par les mêmes k nucléotides. Comme représenté Figure 2.10**(a)**, dans le dBG, ceci se traduit par une bubble dissymétrique. Dans le cas général, le chemin le plus court est composé de $k-1$ noeuds et le chemin le plus long contient $k-1+d$ noeuds.

Comme représenté Figure 2.10**(b)**, le chemin le plus court peut être composé de moins de $k-1$ noeuds dans le cas où l'indel et ce qui suit l'indel ont un préfixe commun de taille > 0. Plus précisément, si ce préfixe commun est de taille p_1 et que la taille du plus grand suffixe commun entre l'indel et ce qui précède (comme présenté par exemple Figure 2.10**(c)**) est de taille p_2, alors le plus court chemin est composé de $\max(0, k-1-p_1-p_2)$ noeuds et le plus long est composé de $k-1+d-p_1-p_2$.

Notons que dans l'une et/ou l'autre des situations précédemment décrite(s), il n'est pas possible de déterminer précisément l'indel détectée. Par exemple, l'indel représentée Figure 2.10**(b)** provient des données ACTGACCT et ACTGATACCT où l'on a supposé que l'indel était AT. Cependant, nous aurions pu colorer les mêmes séquences ainsi : ACTGACCT et ACTGATACCT et ainsi supposer que l'indel est TA. Nous n'avons aucun moyen de prédire la position exacte de l'indel dans une telle situation. En pratique les méthodes de détection d'indel précisent la stratégie adoptée pour situer les variants détectés : usuellement "leftmost" or "rightmost"-based.

Indels non détectées Notons que la méthode de détection d'indels basée sur l'exploitation du dBG implique pour chaque indel qu'il génère des k-mers spécifiques. Bien entendu, si ce n'est pas le cas, l'indel est alors indétectable avec ce type de méthodes. C'est le cas, en autres, d'indels dus à des *homo-polymers* de tailles différentes. Par exemple l'indel AAA situé au centre de la séquence s_1 =CAAAAAAAAG génère la séquence s_2 =CAAAAAAAAAAAAG. Cependant, avec $k = 4$, les k-mers générés par s_1 et par s_2 sont identiques, l'indel AAA n'est donc pas détectable sans utilisation de génome de référence.

2.5.4 Épissage alternatif

Comme présenté Figure 2.4 page 18, dans les données de séquençages transcriptomiques, l'épissage alternatif génère des séquences qui sont, dans leur structure, très similaires aux indels. Le motif du dBG présenté dans la précédente section pour la détection d'indel s'applique également à la détection d'épissage alternatif. Ainsi, la mise en oeuvre algorithmique pourrait être identique pour la détection de ces deux types de variants. Cependant, dans le cas des données transcriptomiques, la mise en oeuvre algorithmique doit prendre en compte certaines spécificités inhérentes aux données d'expressions, en particulier l'hétérogénéité de couverture des séquences. Ceci sera présenté Section 2.6 page 28.

2.5.5 Inversions

Le motif associé aux inversions est également détectable dans le dBG. Dans ce cas, le motif est plus complexe que ceux présentés précédemment. Sans entrer dans les détails et comme représenté Figure 2.11 page suivante, il contient les k-mers témoignants de la présence de l'inversion, c'est à dire ceux situés aux jonctions entre au, vb, $a\overleftarrow{v}$ et $\overleftarrow{u}b$, comme représenté Figure 2.3 page 18. L'idée phare ici est que les k-mers contenus entièrement dans l'inversion I ou $\overleftarrow{I}$ correspondent aux mêmes

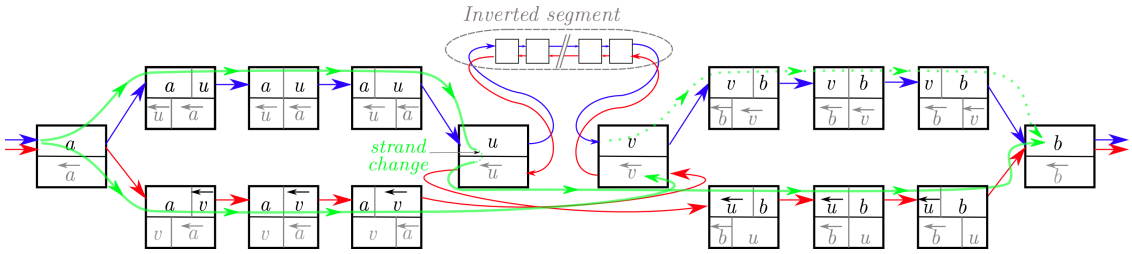


FIGURE 2.11 – Image tirée de [Lemaitre et al., 2014]. Exemple de motif généré par une inversion dans le dBG. Soit $a, I, b \in \Sigma^*$, avec $|a|, |b|$ et $|I| > k$. L'inversion traduite par les séquences aIb (chemin bleu) et $a\overleftarrow{I}b$ (chemin rouge) génère un cycle dans le dBG. Pour simplifier, nous considérons que tous les k-mers et leurs reverse complement sont stockés de manière explicite dans le dBG et qu'une arête entrant (respectivement sortant) la partie haut-gauche (respectivement haut-droite) d'un noeud concerne le k-mer stocké dans ce noeud alors qu'une arête entrant (respectivement sortant) la partie bas-droite (respectivement bas-gauche) d'un noeud lit le reverse complement du k-mer stocké dans ce noeud. Le chemin vert représente le parcours effectué par l'algorithme TakeABreak (voir Section 2.6 page suivante).

noeuds dans le dBG (chaque noeud représentant un k-mer et son reverse complément), alors que les k-mers traversant ces jonctions diffèrent entre les deux séquences impliquées dans l'inversion. Ainsi, le modèle proposé se base sur ces k-mers aux jonctions pour témoigner de la présence d'une inversion.

Notons que le modèle appliqué, basé uniquement sur les k-mers traversants les jonctions des inversions, permet la détection des points de cassure (positions où les chromosomes ont connu une rupture suite au réarrangement) de ces inversions, mais ne permet pas la détection de l'inversion elle même.

2.5.6 Bruit et bubbles dites "*branchantes*"

Motifs indésirables Les motifs du dBG présentés dans les sections précédentes témoignent de la présence de variants d'intérêt. Cependant d'autres éléments biologiques ou technologiques peuvent générer des motifs identiques. Par exemple la présence d'erreurs de séquençage dans les données ayant servi à construire le dBG génère le même motif que celui engendré par de véritables SNP. De plus, des répétitions approchées intra-génomiques peuvent être identiques à quelques substitutions, insertions ou délétions près. Ces évènements biologiques génèrent strictement le même motif que celui associé aux SNPs ou aux indels.

Il est important de proposer des outils capables de faire efficacement la différence entre les véritables éléments recherchés (variants inter chromosomes ou inters individus) et les éléments dus aux répétitions intra-chromosomes ou aux bruits technologiques comme les erreurs de séquençage.

Comme nous le verrons dans la Section 2.7.4 page 31 nous proposons une solution basée sur la couverture relative des éléments détectés pour séparer les vraies prédictions des prédictions fausses, dues à ces artefacts.

Branchements dans les bubbles Le bruit dans les données, les répétitions inexactes, ou plus globalement la complexité intrinsèque des génomes complexifient la topologie du dBG. Ainsi, les modèles précédemment présentés peuvent être "noyés" au sein de structures plus complexes. Tous les modèles que nous avons précédemment évoqués sont composés de bubbles. Ces bubbles peuvent être *branchantes* ou non. Nous dirons d'une bubble qu'elle est branchante lorsque l'un de ses noeuds contient, en plus des arêtes définies dans son modèle, une ou plusieurs arêtes entrantes ou sortantes. La Figure 2.8 page 23, fait la différence entre une bubble non branchante (a) et une bubble branchante (b).

* *

*

Les branchements dans les bubbles représentent l'une des principales difficultés dans les solutions algorithmiques que nous présentons dans la section suivante.

2.6 Mise en oeuvre algorithmique

Nous proposons ici un aperçu des méthodes mises en oeuvre. Les subtilités des modèles appliqués ainsi que les détails algorithmiques peuvent être trouvés dans les publications associées, présentées Section 2.10 page 38.

* *

*

Dans cette section nous présentons les solutions algorithmiques mises en oeuvre pour détecter les motifs précédemment définis pour la détection de SNPs et d'indel (outil discoSnp++), d'inversions (outil TakeABreak) et de variants (SNPs, indels, épissage alternatif) dans les données transcriptomiques (outil Kissplice).

Ces méthodes suivent toutes les trois le même scénario :

- Création du dBG à partir des données NGS brutes (les lectures) ;
- Détection des motifs d'intérêt dans le dBG ;
- Mapping des lectures sur les séquences des motifs ainsi détectés.

Création du dBG La première phase consiste en la création du dBG à partir des lectures. Elle est composée de deux étapes. Lors de la première étape, les k-mers des lectures sont comptés à l'aide de l'outil DSK [Rizk et al., 2013]. Ceci permet, comme présenté Section 1.4.2 page 12, de déterminer le seuil différenciant les k-mers solides des autres et également d'éliminer les erreurs des séquençage. La seconde phase de cette première étape consiste à créer la structure de données représentant le dBG. Ceci est effectué via la structure de données utilisée dans l'outil Minia [Chikhi and Rizk, 2013], en utilisant l'implémentation proposée dans la librairie GATB [Drezen et al., 2014]. Notons ici que les méthodes proposées doivent leur succès entre autres à l'utilisation de cette structure qui offre des performances particulièrement intéressantes en temps et en mémoire.

Détection des motifs Les outils TakeABreak et discoSnp++ parcourent tous les noeuds branchants et testent s'ils initient une bubble témoignant respectivement de la présence d'une inversion (TakeABreak) ou d'un indel ou d'un SNP (discoSnp++). Dans ce cas, les séquences associées à ces bubbles sont stockées dans un fichier au format Fasta. Il s'agit de couples de séquences pour discoSnp++ ou de quadruplés de séquences pour TakeABreak (au,vb,$a\overleftarrow{v}$, et $\overleftarrow{u}b$, comme présenté Figure 2.3 page 18).

Pour ces deux outils, les motifs détectés sont de taille fixe (ou très limitée lors de la détection des indels avec discoSnp++). discoSnp++ est prévu pour la détection d'indels de taille limitée de l'ordre du millier de nucléotides au plus, que nous pouvons donc rapprocher de la notion de microindels. Ainsi, comme nous le verrons dans la Section 2.8 page 31, les temps de calcul associés à la détection de ce type de motifs sont particulièrement réduits.

* *

*

L'outil Kissplice n'est pas limité dans la taille des motifs qu'il prédit. Ainsi, la longueur des exons épissés détectés par exemple, n'est pas limitée. L'inconvénient majeur est que le parcours du graphe lors de la détection des bubbles n'est pas contraint et peut être associé à des temps de calcul importants dans le cas de graphes complexes. Dans le cas des données d'expressions, les différents gènes exprimés ont peu de chances de générer des k-mers identiques. Ainsi, il est attendu que chaque gène exprimé génère une *composante connexe** distincte du graphe. De plus les motifs recherchés correspondent à des cycles dans le dBG. Ainsi, nous ne sommes intéressés que par les composantes contenant des cycles, que nous détectons via l'identification de composantes dites *bi-connexes**. Dans une composante bi-connexe, pour chaque couple de sommets, il existe au moins deux chemins distincts du graphe reliant ces deux sommets.

Dans le cas des données transcriptomiques, à l'inverse des données génomiques, la couverture des différentes régions séquencées est très hétérogène car elle dépend de l'expression des gènes. Ainsi, il est dommage de filtrer les k-mers dont la couverture est sous un seuil fixe, appliqué indifféremment à toutes les régions du graphe comme présenté Section 1.4.2 page 12. Dans la version actuelle de l'outil Kissplice, le seuil de couverture pour l'élimination des k-mers non solides est relatif. En pratique, lors de la traversée de noeuds branchants, les couvertures relatives des noeuds fils atteignables sont calculées. Il s'agit pour chacun de ces noeuds de sa couverture divisée par la somme des couvertures des noeuds atteignables. Les noeuds dont la couverture relative est inférieure à un seuil prédéfini, sont considérés comme non solides et ne sont pas explorés lors de la recherche de *bubbles* correspondant aux motifs recherchés.

Mapping des lectures sur les prédictions Nous exposons dans la Section 2.7.1 page suivante les raisons et les difficultés de cette étape.

2.7 Difficultés et solutions

Sur le papier, tout ce que l'on vient de décrire semble idyllique. Même si les approches de novo que nous proposons semblent plus simples que les méthodes de détection par mapping, il existe

cependant des difficultés que nous présentons dans cette section.

2.7.1 Perte des informations de couverture

Un inconvénient majeur inhérent à la structure en k-mers du dBG et à l'implémentation que nous utilisons (GATB) est que les séquences lues dans le graphe **i/** ne comportent aucune information de couverture ou de qualité des lectures associées ; et **ii/** peuvent être chimériques, c'est à dire dues à une succession de k-mers n'existant pas conjointement dans les lectures. Pour pallier à ces deux limitations, nous appliquons une étape de mapping des lectures sur les séquences des motifs prédits issues des parcours des motifs dans le dBG. L'analyse de ces résultats de mapping permet d'éliminer ces séquences chimériques (celles sur lesquelles certaines portions ne sont pas mappées par des lectures). Ces séquences sont appelées "*uncoherent*", alors que les autres sont appelées "*coherent*". De plus l'analyse du mapping des lectures permet d'assigner pour chaque séquence sa couverture en terme de nombre de lectures mappées et sa qualité moyenne dans chacun des jeux de données utilisés.

L'inconvénient majeur de cette phase tient dans le temps de calcul et la mémoire qu'elle nécessite. En pratique elle représente approximativement au moins la moitié des temps de calcul. En outre elle collecte pour chaque variant détecté des informations de couverture et de qualité. Si le nombre de variants prédits est important ces informations peuvent avoir un impact non négligeable sur l'empreinte mémoire. En pratique, sur l'intégralité des tests que nous avons effectués, la mémoire utilisée par cette phase n'a pas dépassé 8GB.

2.7.2 Unicité des prédictions

Comme nous l'avons mentionné, dans la structure de dBG utilisée, chaque noeud contient un k-mer et, implicitement, son reverse complément. Lors de la détection des motifs associés aux variants recherchés, les motifs sont détectés une fois de la "droite vers la gauche", et une autre fois de la "gauche vers la droite". C'est assez contre-intuitif, mais il n'est pas possible de déterminer à l'avance si un motif en cours de détection avait déjà été découvert précédemment dans le sens opposé. À l'exception de la détection des inversions, l'unicité est effectuée en comparant les couples de noeuds branchants aux extrémités des motifs prédits et en ne conservant que ceux respectant un ordre lexicographique prédéfini.

Dans le cas de la détection des inversions, l'unicité est plus complexe à prédire car il y a huit détections possible de la même inversion : (au, vb), $(a\overleftarrow{v}, \overleftarrow{u}b)$, $(\overleftarrow{u}\overleftarrow{a}, \overleftarrow{b}\overleftarrow{v})$, $(\overleftarrow{u}b, a\overleftarrow{v})$, (vb, au), $(v\overleftarrow{a}, \overleftarrow{b}u)$, $(\overleftarrow{b}\overleftarrow{v}, \overleftarrow{u}\overleftarrow{a})$, et $(\overleftarrow{b}u, v\overleftarrow{a})$. Dans l'outil TakeABreak, l'inversion représentée par le couple de mots le plus petit lexicographiquement est la seule reportée.

2.7.3 Locus des prédictions

La limitation principale des approches de prédiction de novo de variants que nous proposons est évidente : les variants prédits ne sont pas localisés sur le génome auquel ils appartiennent. Étonnamment, à travers l'utilisation actuelle des variants connus, la connaissance de cette information de locus n'est indispensable que dans quelques cas, essentiellement lorsque les variants prédits doivent être affiliés à des annotations connues (positions et rôles des gènes le long du génome).

Cependant, de nombreux outils d'analyse de ces variants utilisent un format de données appelé VCF ("*Variant Calling Format*") qui représente les variants par rapport à une référence et qui utilise celle-ci pour les localiser.

Ainsi, dans le cas où un génome de référence est disponible nous proposons de l'utiliser en fin de prédiction pour y mapper les variants prédits et pour ainsi générer un fichier de type VCF. Ceci est implémenté pour l'outil Kissplice (KisSplice2refgenome `http://kissplice.prabi.fr/tools/kiss2refgenome/`) et pour l'outil discoSnp++ (VCF_creator [Riou et al., 2015]).

Cette étape de mapping des variants sur un génome de référence peut paraître contre-intuitive dans le cadre de la détection de novo de variants. Cependant, nous tendons à montrer qu'il est nettement plus efficace selon tous les critères (temps, mémoire, qualité des résultats) de détecter de novo des variants et de les mapper ensuite sur une référence que de détecter les variants via un mapping des lectures sur cette référence.

Comme discuté Section 2.9.1 page 36 l'un des travaux futurs consistera à déterminer l'exactitude de cette assertion ou de déterminer avec précision les cas où celle-ci est vraie.

2.7.4 Séparer l'ivraie du bon grain : utilisation des informations de couverture et de mapping

Comme nous l'avons évoqué Section 2.5.6 page 27, tous les motifs prédits ne correspondent pas à de véritables motifs présents dans les données brutes. Les prédictions contiennent en effet des faux positifs.

Les faux positifs dus à des erreurs de séquençage sont limités par l'élimination des k-mers non solides (voir Section 1.4.2 page 12).

Comme nous l'avons évoqué, une source majeure de faux positifs est due à la présence de répétitions inexactes qui génèrent les mêmes motifs que ceux témoignant de la présence d'un motif d'intérêt. Ce type de faux positif peut être limité/éliminé grâce à l'idée suivante lors de la comparaison de deux jeux de données ou plus. Les répétitions inexactes sont similaires d'un individu à l'autre. Ainsi, lors de la comparaison de plusieurs individus (donc de plusieurs jeux de données), nous nous attendons à ce que ce type de faux positif ait une couverture similaire dans tous les jeux de données. Concrètement nous préfèrerons les prédictions pour lesquelles il existe au moins un couple d'individus pour lesquels les couvertures varient fortement d'un individu à l'autre. En pratique, pour chaque prédiction, nous utilisons une mesure basée sur le coefficient Phi de la table des couvertures de chacune des deux séquences et pour chaque jeu de données. Ce coefficient vaut $\sqrt{\frac{\chi^2}{n}}$, basé sur la statistique du chi2 de la table de contingence de couverture des allèles par les jeux de données et où n est la somme des couvertures associées au variant. Phi varie entre 0 et 1, une valeur proche de zéro témoigne d'un variant ayant des couvertures similaires dans tous les jeux de données, et inversement, une valeur proche de un, indique un variant dont les couvertures varient fortement d'un jeu de données à l'autre.

2.8 Résultats

Nous présentons dans cette section les résultats majeurs des outils de détection de novo de variants.

2.8.1 Détection de variants transcriptomiques avec l'outils Kissplice

Kissplice a été appliqué sur des données réelles humaines composées de 32 millions de lectures provenant du cerveau et de 39 millions de lectures provenant du foie (Projet Illumina Body Map 2.0 (Sequence Read Archive, identifiant ERP000546).

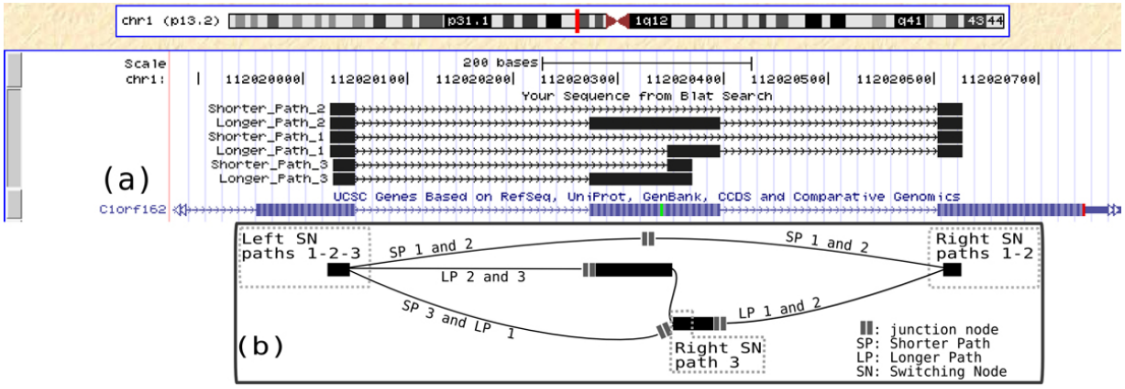


FIGURE 2.12 – Divers visualisations d'une composante bi-connexe correspondant à un évènement d'épissage complexe. **(a)** Image issue du "*Genome Browser UCSC*" http://genome.ucsc.edu/. Les séquences représentées en noir ont été prédites par Kissplice. La séquence représentée en bleu est issue des annotations connues. L'exon intermédiaire (rectangle bleu du milieu de la séquence "Ciorf162") est soit présent, partiellement présent ou épissé. **(b)** Représentation du dBG compacté associé à cette composante. Image tirée de [Sacomoto et al., 2012]

Sur ces données, Kissplice a identifié 3657 composantes bi-connexes (voir Section 2.6 page 28) représentant un intérêt en terme de détection de transcrits alternatifs. Parmi ces 3657 prédictions, 3497 (95%) ont été validées par mapping (les deux séquences de chaque variant mappant les mêmes positions génomiques : début et fin). La majorité des 5% de prédictions restantes correspond à des transcrits chimériques pouvant être générés par des réarrangements génomiques. Ceci n'a pas été plus investigué. Parmi les 3497 prédictions, seules 1538 étaient connues, contre 1959 alors considérées comme de nouvelles descriptions d'évènements d'épissages alternatifs. De plus, 719 prédictions contenaient plus d'un évènement d'épissage alternatif, révélant la présence de gènes pouvant générer plus de deux transcrits. Un exemple d'un tel évènement est présenté Figure 2.12.

2.8.2 Détection d'inversions avec l'outil TakeABreak

	Recall (%)	Précision (%)	# FP	Temps(s)	Mémoire(GB)
E. coli	100.00	100.00	0	1	1
C. elegans	96.00	99.07	9	935	1
chromosome humain 22	87.60	92.50	71	5412	1

TABLE 2.1 – Résultats de TakeABreak avec ses paramètres par défaut sur des données simulées à partir de génomes de diverses taille et complexité. Données issues de [Lemaitre et al., 2014]

L'outil TakeABreak n'a été testé avec succès pour l'heure que sur des données simulées pour lesquelles le profil des erreurs et les inversions sont parfaitement maitrisés et connus. Les simulations étaient basées sur les génomes d'*E.coli*, de *C.elegans* et sur le chromosome humain n°22. Les résultats présentés Table 2.1 page ci-contre montrent que sur un génome simple tel *E.coli* les résultats sont parfaits et ont une très faible empreinte mémoire. Les résultats sur un génome plus complexe à l'image du chromosome 22 humain montrent la difficulté de faire la différence entre les motifs générés par de véritables inversions et les motifs dus à des répétitions inexactes ou à des erreurs de séquençage. Notons que les temps d'exécution et l'empreinte mémoire restent très limités.

* *

*

Une difficulté principale liée à l'utilisation de TakeABreak sur des données réelles est due au fait que les jonctions d'inversions sont sujettes à une forte variabilité Lieber et al. [2003] ; O'Driscoll and Jeggo [2006]. Ainsi, les k-mers branchants initiant ou terminant le motif associé à une inversion (k-mers a et b représentés Figure 2.11 page 27) ont une importante probabilité d'être distincts dans les deux versions de l'insertion. Dans ce cas, le motif associé à l'inversion n'existe pas dans le dBG. L'un des axes de recherche futur (voir Section 2.9.2 page 37) consiste à intégrer différentes approches de détection de motifs permettant la détection conjointe de la variabilité locale (SNPs, micro indels) et globale (inversions), ce qui limitera ce problème.

2.8.3 Détection de petits indels et de SNP avec l'outil discoSnp++

Résultats sur des données simulées

L'utilisation de données simulées a permis de connaitre avec précision les résultats qualitatifs (en terme de précision et de recall) de discoSnp++ et des autres outils pouvant être utilisés pour effectuer le même type de détection de variants. Nous proposons des résultats comparés avec les outils populaires de l'état de l'art. Il s'agit de l'outil Cortex, décrit dans [Iqbal et al., 2012], qui est un outil permettant la détection de variants de novo. Nous nous comparons également à une approche dite *hybride* qui consiste à 1/ assembler les lectures en utilisant l'assembleur SOAPdenovo2 [Luo et al., 2012], 2/ à mapper les lectures sur les assemblages obtenus en utilisant l'outil Bowtie2 [Langmead and Salzberg, 2012b] et finalement à en déduire la présence de variants en utilisant l'outil GATK [McKenna et al., 2010].

Les résultats présentés Figure 2.13 page suivante permettent d'estimer la qualité des résultats sur des données issues du génome humain (chromosome 1). Ces résultats mettent en évidence plusieurs points avantageux pour discoSnp++ :

- L'indice *Phi* associé à chaque variant détecté est particulièrement efficace. En effet, qu'il s'agisse de SNPs ou indels, les prédictions obtenues avec une valeur de $Phi \geq 0.2$ sont quasiment toutes de vrais positifs (97.78% des SNPs et 98.97% des indels). L'utilisateur ne cherchant pas nécessairement un recall excellent privilégiera donc les prédictions avec une grande valeur de *Phi*. Nous pouvons noter que ce système de classement des prédictions est à l'inverse mauvais pour les variants obtenus avec l'approche hybride. Les résultats détectés par Cortex ne sont pas classés et ne permettent donc pas de privilégier certaines prédictions plutôt que d'autres.

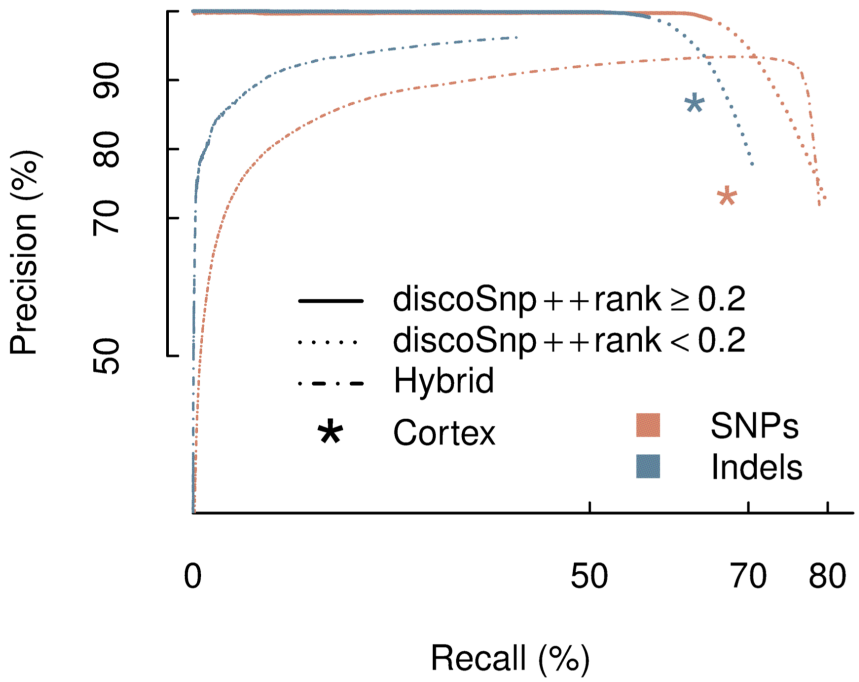


FIGURE 2.13 – Résultats de détection d'indels et de SNP sur des données simulées provenant du chromosome 1 humain. Les courbes de précision par rapport au recall sont obtenues en classant les prédictions par rapport à leur score *Phi* pour discoSnp++ (voir Section 2.7.4 page 31) ou par rapport à leur score fourni par GATK pour l'approche hybride. Les lignes pointillées des résultats discoSnp++ concernent les prédictions pour lesquelles $Phi < 0.2$.

- Concernant les SNPs, la précision et le recall globaux des trois approches testées sont assez similaires, même si le recall obtenu par l'outil Cortex est en léger retrait.
- Concernant les indels, nous constatons ici les limites des approches par mapping. Dans cette situation, les outils de détection de novo (discoSnp++ et Cortex) obtiennent de bien meilleurs résultats en terme de recall. Ceci s'explique par le fait que le mapping de lectures est rendu délicat par les insertions et/ou délétions qu'elles contiennent.

La Figure 2.14 page ci-contre présente les résultats en terme de temps de calcul et de quantité de mémoire consommée pour les trois approches testées. Ces résultats montrent que l'approche discoSnp++ est particulièrement efficace en terme de temps de calcul. De plus, ce résultat met en lumière le très faible impact mémoire de discoSnp++, en particulier en regard des autres approches étudiées. Ce faible impact mémoire est l'un des atouts majeurs de l'approche discoSnp++, lui permettant d'être utilisé sur de simples ordinateurs de bureau. Ceci offre la possibilité d'utiliser discoSnp++ pour les utilisateurs n'ayant pas ou peu d'accès à de grands centres de calcul. En outre, discoSnp++ ne mobilisant pas l'intégralité des ressources mémoires disponibles, il est possible de lancer en parallèle plusieurs instances de discoSnp++ sur des données différentes ou avec des paramètres différents.

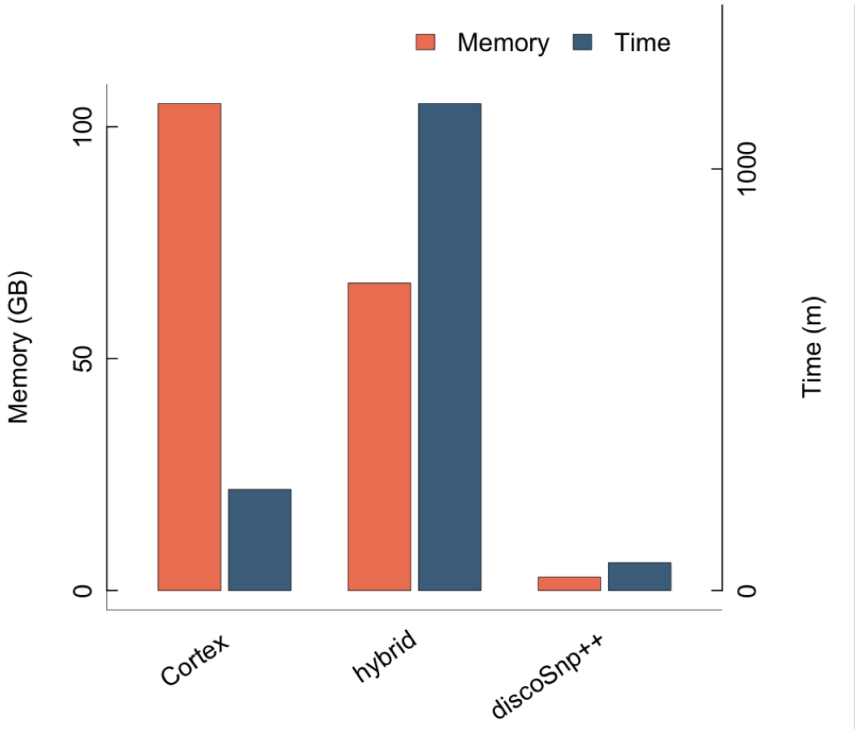


FIGURE 2.14 – Impact en terme de temps de calcul et de mémoire consommée des trois méthodes testées sur les données du chromosome 1 humain composée de deux fichiers de lectures de taille 100, représentant chacun 40x de couverture.

Résultats sur des données réelles

Application à deux souches de souris Un test a été effectué pour la détection de SNPs entre deux souches de souris issues de l'étude [Wong et al., 2012]. Ce test n'est pas particulièrement informatif en terme de qualité des résultats obtenus car nous ne disposons pas dans ce cas de la liste exacte et exhaustive des variants existant effectivement dans les données. Les informations de précision et de recall ne peuvent donc pas être calculées.

Un point intéressant concerne les ressources utilisées. Les jeux de données employés contiennent près de trois milliards de lectures. Sur ces données, les prédictions ont été effectuées par discoSnp++ en moins de six jours, en utilisant au plus 4.5GB de mémoire et en ne nécessitant qu'une ligne de commande. En comparaison, les autres outils testés ont dépassé la mémoire disponible qui était de 512GB et les résultats de l'étude présentés dans [Wong et al., 2012] ont nécessité l'utilisation d'un pipeline complexe composé de 6 outils distincts, et appliquant 14 filtres non automatisés.

Application aux données tique L'outil discoSnp++ a été utilisé pour effectuer plusieurs bio-analyses. Nous mentionnons ici un résultat obtenu dans [Quillery et al., 2014] où il a été utilisé sur des données du génome de la tique (*I.ricinus*). Appliqué sur ces données, discoSnp++ a détecté 321088 SNPs. Parmi elles, les collègues biologistes en ont sélectionné 384 pour *génotyper* 464 individus à

l'aide de la technologie *Fluidigm*. Parmi ces 384 SNPs, 368 (95.8%) ont pu être détectés chez les individus génotypés.

2.9 Perspectives

Nous sommes heureux d'avoir réussi à proposer des outils de détection de variants de novo aboutis, en particulier discoSnp++ et Kissplice, l'outil TakeABreak étant encore limité à un cadre théorique voué à évoluer pour obtenir de meilleures prédictions sur des données réelles. La faible empreinte mémoire, le temps d'exécution réduit et la qualité des résultats de Kissplice et discoSnp++ font qu'ils sont maintenant régulièrement utilisés par la communauté. Ceci se traduit pour l'heure par quelques publications citant les outils, mais aussi et surtout par les retours informels sur le site de Bio* (https://www.biostars.org/) ou par messagerie électronique et par les statistiques de visite du site web. Le site web du projet qui héberge les outils (http://colibread.inria.fr) reçoit en effet environ 600 visiteurs uniques par mois.

* *

*

Diverses perspectives s'offrent à nous pour la poursuite de ce travail.

2.9.1 Changement des mentalités pour l'utilisation de méthodes de détection de novo

La prédiction de variants de novo n'est pas commune. Si les utilisateurs déclarés de nos outils sont satisfaits de leurs utilisation, nombreux sont ceux qui préfèrent la méthode "classique" qui consiste à mapper les lectures sur une séquence de référence, bonne ou mauvaise, quitte à en créer une par assemblage si besoin. Ce processus est tout à fait acceptable et a conduit à de nombreux très bons résultats, à l'image de ceux obtenus chez l'humain Altshuler et al. [2012]. Ainsi, une des étapes clefs pour le succès de nos méthodes consiste à diffuser le plus largement possible nos outils via divers supports (forums, formations, publications, blogs, conférences, ...).

L'une des futures étapes clefs consistera à prouver que les approches de novo offrent de meilleurs résultats que les approches mapping+calling, ou , à défaut, à déterminer dans quelle condition il est opportun d'utiliser une méthode plutôt que l'autre. Nos expériences actuelles et les retours de nos utilisateurs n'ont pas encore permis de déterminer de cas où l'approche mapping+calling est préférable à notre approche, mais ceci doit être précisément mesuré et prouvé de la manière la plus honnête possible.

Ceci représente un travail important car pour le mener à bien il est nécessaire de maitriser l'éventail des outils testés, ainsi que les jeux de données et les questions biologiques associées qui sont très diverses. Dans le cadre du projet ARN *Colib'read* nous disposons de nombreux jeux de données et nous réunissons de nombreuses compétences couvrant une majeure partie du spectre des problématiques de détection de variants. L'un de nos prochains défis sera donc de fédérer ces forces pour effectuer cet important travail.

2.9.2 Amélioration des assemblages

L'une des difficultés de l'assemblage réside dans la prise en compte des organismes séquencés et notamment dans la gestion des variants rencontrés au sein des génomes complexes. Ces variations, bien que souvent informatives et recherchées, ne sont généralement pas reportées par les assembleurs et sont gênantes pour le processus d'assemblage. Ces variations induisent des choix et, sachant que les assembleurs classiques sont pensés pour produire un texte unique, elles sont sources d'erreurs d'assemblage et aboutissent à des textes plus fragmentés. En outre, la détection de variants est un problème central en génomique. Elle est habituellement effectuée a posteriori de l'assemblage et souffre des erreurs et des informations perdues durant cette étape.

L'un de nos objectifs est de proposer un outil unique capable à la fois de détecter les variants par les méthodes présentées dans ce chapitre et d'utiliser cette information pour produire de meilleurs assemblages (moins fragmentés et moins erronés).

L'axe de recherche initial sera de mettre en relation les outils de détection de variants avec les outils d'assemblage classiques de génomes. Plusieurs idées seront à explorer : (1) proposer une réconciliation harmonieuse des résultats d'assemblage et de détection de variants effectués indépendamment ; (2) Détecter dans un premier temps les variants, supprimer ces variants des données de séquençage puis finir par un assemblage classique sur ces données ainsi "nettoyées" ; (3) Développer de nouveaux algorithmes gérant les deux aspects de front en s'inspirant des réalisations et de l'expertise développées dans ces domaines au sein de l'équipe. Chacune des trois idées proposées a ses qualités et ses défauts qu'il faudra précisément identifier et comprendre avant de décider de la stratégie à mettre en oeuvre. Une attention particulière sera portée à l'évolution des technologies de séquençage et aux masses de données en jeu, en particulier via l'utilisation de structures de données telles que le filtre de Bloom, adaptées à l'indexation efficace des données représentées sous forme d'ensembles de k-mers

2.9.3 Suivi des modifications technologiques

Comme nous l'avons évoqué dans le contexte de ce document (Section 1.3 page 4), les caractéristiques des données de séquençage évoluent, notamment à travers l'arrivée de la technologie de séquençage TGS (*third generation sequencing*) qui produit des lectures plus longues avec un taux d'erreurs plus grand.

L'utilisation de ces données TGS va changer le paysage de l'analyse bioinformatique. Les variants structuraux de grande taille seront plus aisément détectables dans ces nouvelles données et il sera plus facile de détecter et d'assembler correctement les répétitions génomiques.

Il n'est pas encore clair de savoir si ce nouveau type de données remplacera ou s'additionnera aux données NGS actuellement majoritairement utilisées (données Illumina). Outre les questions de prix et de logistiques liées à la génération de données TGS, il n'est pas certain que ces données, constituées de peu de grandes lectures, soient une bonne solution pour tous les cas d'utilisation. En effet, les lectures plus courtes et abondantes sont plus adaptées au diagnostic, à la détection d'éléments rares, à la détection de séquences courtes telles que celles que l'on utilise dans l'ADN dégradé ou le séquençage transcriptomique.

Quoi qu'il en soit, les perspectives des travaux présentés dans ce chapitre s'inscrivent nécessairement dans l'adaptation de nos méthodes à ce nouveau type de données et dans la réflexion sur les change-

ments potentiellement profonds qu'elles engendreront.

2.10 Présentation des publications associées

▶ Peterlongo, P., Schnel, N., Pisanti, N., Sagot, M.-F., and Lacroix, V. (2010). Identifying snps without a reference genome by comparing raw reads. In *String Processing and Information Retrieval*, pages 147–158. Springer

Cette publication (fournie en annexe page 2) est la première à présenter l'idée de la détection de SNP dans des données de séquençage brutes, sans utilisation de génome de référence. La solution proposée n'est applicable que pour la comparaison de deux jeux de données. Le modèle de détection est basé à la fois sur le motif topologique induit par les SNPs dans le dBG, mais aussi sur la différence des couvertures des deux jeux de données. Les solutions présentées avaient un apport théorique intéressant mais leur application était limitée à de petits jeux de données du fait du modèle appliqué et des structures de données implémentées.

L'outil présenté dans cette publication n'est plus utilisé actuellement, mais a servi de preuve de concept et a ouvert la voie aux autres outils présentés dans ce chapitre.

▶ Sacomoto, G. A., Kielbassa, J., Chikhi, R., Uricaru, R., Antoniou, P., Sagot, M.-F., Peterlongo, P., and Lacroix, V. (2012). Kissplice : de-novo calling alternative splicing events from rna-seq data. *BMC bioinformatics*, 13(Suppl 6) :S5

Cette publication (fournie en annexe page 15) présente l'outil Kissplice, de détection d'épissage alternatif dans les données transcriptomiques et montre des résultats préliminaires obtenus sur des données simulées et sur des données réelles.

▶ Uricaru, R., Rizk, G., Lacroix, V., Quillery, E., Plantard, O., Chikhi, R., Lemaitre, C., and Peterlongo, P. (2015). Reference-free detection of isolated snps. *Nucleic acids research*, 43(2) :e11–e11

Cette publication (fournie en annexe page 28) présente l'outil discoSnp++. Des résultats sont présentés sur des données simulées à partir de génomes réels ainsi que sur des données réelles. L'outil présenté, discoSnp++ ne détectait alors que des SNPs isolés. La nouvelle version détectant également les indels et les SNPs proches et utilisant possiblement en plus un génome de référence est appelée discoSnp++ ++ et n'est pas encore publiée.

▶ Quillery, E., Quenez, O., Peterlongo, P., and Plantard, O. (2014). Development of genomic resources for the tick Ixodes ricinus : isolation and characterization of single nucleotide polymorphisms. *Molecular ecology resources*, 14(2) :393–400

Cette publication (fournie en annexe page 40) présente l'un des premiers résultats biologiques obtenus grâce à l'utilisation de discoSnp++ sur des données issues du génome de la tique.

▶ Lemaitre, C., Ciortuz, L., and Peterlongo, P. (2014). Mapping-free and assembly-free discovery of inversion breakpoints from raw NGS reads. In Dediu, A.-H., Martín-Vide, C., and Truthe, B.,

editors, *Algorithms for Computational Biology*, volume 8542, pages 119–130, Tarragona, Spain

La publication (fournie en annexe page 49) présente l'outil TakeABreak. Les résultats qui y sont présentés sont limités à des tests effectués sur des données simulées à partir de génomes réels.

Chapitre 3

Comparaison de données de séquençage

Contents

3.1 Introduction 41
3.2 Brève introduction à la métagénomique comparative 46
3.3 Outils de détection de lectures similaires entre jeux de séquences NGS 48
3.4 Résultats 55
3.5 Outils de comparaisons d'ensembles de k-mers similaires entre jeux de séquences NGS 60
3.6 Perspectives 62
3.7 Présentation des publications associées 63

3.1 Introduction

Ce chapitre est largement inspiré du manuscrit de thèse de Nicolas Maillet [Maillet, 2013] que j'ai co-encadré. Certains passages sont repris dans ces lignes.

3.1.1 Métagenomique

Pour séquencer un microorganisme, il est essentiel de le cultiver en laboratoire, ne serait-ce que pour l'isoler et ainsi ne séquencer que lui. Or, beaucoup de microorganismes ne survivent pas en milieu contrôlé : il est donc impossible d'isoler et de séquencer ces espèces. La génomique n'est donc pas un moyen d'étudier ces organismes. Par exemple, on estime que seules 0,001% à 0,1% des bactéries présentes dans l'eau de mer peuvent effectivement vivre en milieu contrôlé [Amann et al., 1995].

Cependant, il est possible d'étudier à l'échelle génomique les microorganismes non cultivables en les séquençant directement dans leur milieu. Il est alors impossible d'isoler une seule espèce des autres et la génomique classique ne peut être utilisée. On fait face ici à une différence d'échelle : la

génétique étudie les gènes d'un organisme, la génomique, le génome d'un organisme, ici on travaille sur un ensemble de génomes inconnus : on parle alors de **métagénomique**.

La métagénomique vise à étudier le contenu génétique d'un échantillon provenant d'un environnement naturel. Par exemple, on prélève un litre d'eau de mer, on récupère l'ensemble de l'ADN présent dans cette eau et on séquence cet ADN. Il n'est ainsi pas nécessaire de pouvoir cultiver les espèces. Par contre, il n'est pas possible de savoir directement quelles espèces ont été séquencées.

La métagénomique soulève de nouvelles problématiques :

- Combien d'espèces y-a-t'il dans cet échantillon ? Quelles espèces sont présentes ? Quelles sont les relations entre ces espèces ? Quelles protéines sont sécrétées dans le milieu ? Qui sécrète quoi ? Comment assembler ces données ?
- Comment traiter ces données ? La masse de données peut être de l'ordre de plusieurs dizaines à centaines de gigaoctets pour un seul échantillon. Le projet **Tara Oceans** (`http://oceans.taraexpeditions.org/`), dont nous reparlerons plus bas, génère plus de 2000 échantillons, pour un volume de données estimé à plus d'un pétaoctet. Traiter une telle quantité de données demande de nouveaux algorithmes et de nouvelles structures de données adaptés.

3.1.2 Assemblage de métagénomes

Pouvoir assembler un métagénome permettrait d'obtenir séparément tous les génomes présents dans un échantillon. Mais les assembleurs couramment utilisés en génomique sont dédiés à l'assemblage d'un unique génome, ayant une bonne couverture relativement uniforme.

Dans le Chapitre 1, nous avions listé les conditions nécessaires à l'assemblage parfait de données génomiques (Section 1.4.2 page 9). Dans le cas de la métagénomique, la première condition que nous avions évoquée ("*que les lectures soient exemptes de toute erreur de séquençage*") devient particulièrement délicate à remplir. Il n'est plus possible de déterminer un seuil de solidité (voir Section 1.4.2 page 12) pour séparer les k-mers supposés dus aux erreurs de séquençage des autres. En effet, les espèces les plus présentes peuvent avoir une abondance très largement supérieure (plusieurs dizaines ou centaines de fois supérieures [Magurran and Henderson, 2012 ; Bianchi and Kersten, 2014]) à l'abondance des espèces rares. Fixer un seuil trop bas conduit inévitablement à éliminer les données associées aux espèces rares alors que fixer un seuil plus haut conduit à conserver un nombre important d'erreurs de séquençage.

La troisième condition que nous avions évoquée ("*que la séquence à assembler soit exempte de toute répétition de longueur* $\geq k-1$") reste bien entendue vraie et ce pour chacun des génomes présents dans l'échantillon séquencé. De plus une sixième condition entre en jeu dans le cadre de l'assemblage métagénomique :

- qu'il n'existe aucune répétition de longueur $\geq k-1$ **entre** les génomes des différentes espèces en présence.

En pratique, des espèces proches partagent une partie de leur génome, et la sixième condition que nous venons d'évoquer ne peut donc être remplie. Dès lors, les logiciels d'assemblage tendent à fusionner les séquences provenant d'espèces différentes et créent alors des séquences chimériques [Wooley et al., 2010]. Ainsi, les logiciels d'assemblage ne fonctionnent pas correctement sur les données métagénomiques.

L'assemblage de métagénomes représente un défi stimulant pour la communauté, non seulement pour son intérêt applicatif mais aussi de part les difficultés algorithmiques qu'il met en jeu. Quelques assembleurs métagénomiques ont été publiés ces dernières années, à l'image de MetaVelvet [Namiki et al., 2012], Meta-IDBA [Peng et al., 2011], Ray Meta [Boisvert et al., 2012] ou MEGAHIT [Li et al., 2015]. Ces assembleurs métagénomiques sont de plus en plus efficaces. Quoi qu'il en soit, l'assemblage de métagénomes reste à l'heure actuelle une problématique ouverte pour laquelle il n'existe pas de solution efficace et universelle. Ceci est vérifié par les résultats du challenge CAMI ("Critical Assessment of Metagenomic Interpretation" http://www.cami-challenge.org/). Les résultats préliminaires (https://data.cami-challenge.org/analyseGetView) montrent que les méthodes génèrent toutes des résultats fragmentés, ne couvrant pas la totalité des génomes présents et font des erreurs d'assemblage. Notons également que ce challenge se base sur l'analyse de métagénomes "simples", composés de quelques espèces seulement. Ceci est à mettre en opposition à des métagénomes extrêmement complexes à l'image de ceux issus de projets métagénomiques tels que le projet Tara Oceans que nous présentons dans la section suivante.

3.1.3 Le projet Tara Oceans

Tara Oceans est une expédition scientifique visant à analyser, décrire et étudier les microorganismes marins. La goélette Tara a emmené à son bord, pendant trois ans, de nombreux scientifiques venant de disciplines variées. Le consortium Tara Oceans réunit plus d'une centaine de scientifiques venant de différents domaines comme l'océanographie, l'écologie microbienne, la génomique, la biologie cellulaire, la biologie moléculaire, la biologie des systèmes, la taxonomie, la modélisation d'écosystèmes ou la bio-informatique. Le programme d'échantillonnage comprend des relevés optiques

et génétiques de virus, bactéries, archées, protistes, et métazoaires mais aussi des conditions physico-chimiques dans lesquelles évoluent ces organismes. L'étude à l'échelle du globe sur la morphologie, la génétique et la bio-diversité fonctionnelle des microorganismes planctoniques, tout cela mis en relation avec les changements physico-chimique des océans, devient critique pour comprendre et gérer nos océans [Karsenti et al., 2011].

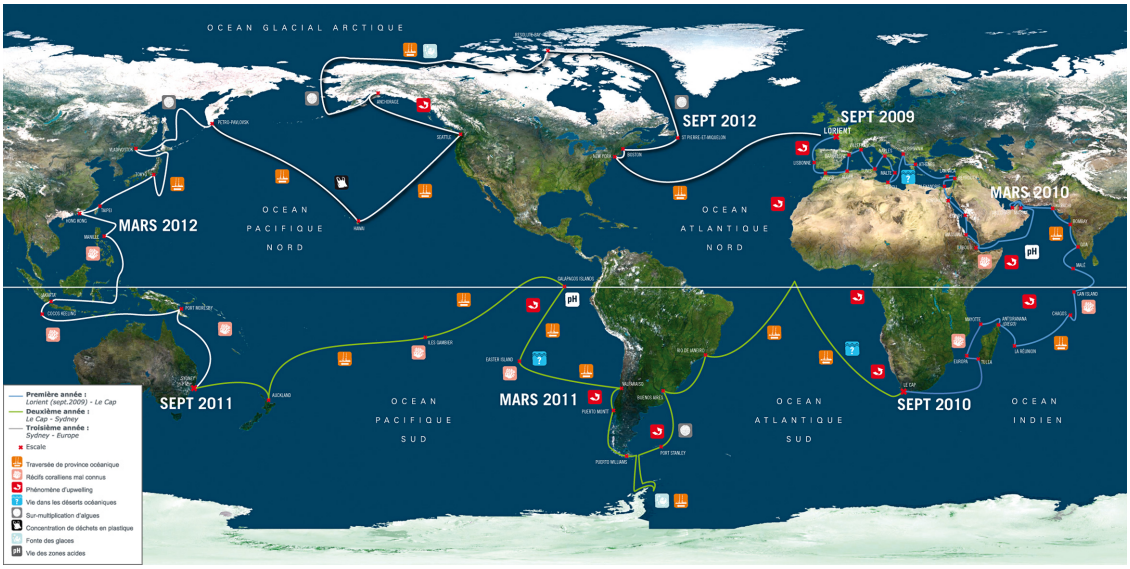


FIGURE 3.1 – Le trajet suivi par la goélette Tara lors de l'expédition "Tara Oceans"

Analyse génomique Un enjeu principal du projet est d'analyser et de comparer les différentes populations d'espèces microscopiques (phages, virus, bactéries, protistes et métazoaires) présentes dans les océans. De plus, le projet vise à étudier l'effet des variations environnementales sur ces populations [Karsenti, 2012].

La goélette Tara a prélevé durant son voyage (Figure 3.1) de l'eau en 155 positions différentes en mer. Ces arrêts sont nommés "stations". Pour chaque station, des prélèvements de plancton à deux profondeurs distinctes ont été réalisés : une en surface et la seconde à la profondeur où la chlorophylle est maximale (DCM pour "*Deep Chlorophyl Maximum*"). Ce DCM représente la couche dans la colonne d'eau où le phytoplancton est le plus présent. Ces deux prélèvements conduisent chacun à plusieurs échantillons. L'eau de mer est filtrée à différentes tailles pour cibler plusieurs types de microorganismes, allant des virus jusqu'aux petits métazoaires.

Exploitation des données Tara Oceans L'exploitation des données issues de l'exploration Tara Oceans a débuté et prendra encore de nombreux mois ou années avant de livrer ses secrets. Cette année, un numéro spécial de *Science* a réuni quatre publications `http://www.sciencemag.org/content/348/6237/873.short` présentant les premiers résultats issus de ces données.

Les données métagénomiques issues du projet Tara s'assemblent extrêmement mal. Les assemblages génomiques n'ont donc, pour le moment, pas pu être exploités. Cependant les gènes ribosomiques issus des données métatranscriptomiques ont été analysés [de Vargas et al., 2015] pour estimer la diversité des organismes *eukaryotes* de la *zone photique**. Ceci a permis de déterminer de nouveaux résultats comme le fait qu'une large source de biodiversité est issue de lignées de protistes unicellulaires *hétérotrophes** non cultivés, peu connus jusqu'alors.

L'exploitation des données métagénomiques se fait en combinant l'exploitation de fragments de séquences connues (les *marqueurs** 16S ou V9 des gènes 18S) appelées des *metabarcodes* avec des analyses comparatives *de novo* des métagénomes complets [Lima-Mendez et al., 2015]. L'exploitation des marqueurs a également permis de modéliser les interactions entres virus, procaryotes, et eucaryotes [Villar et al., 2015].

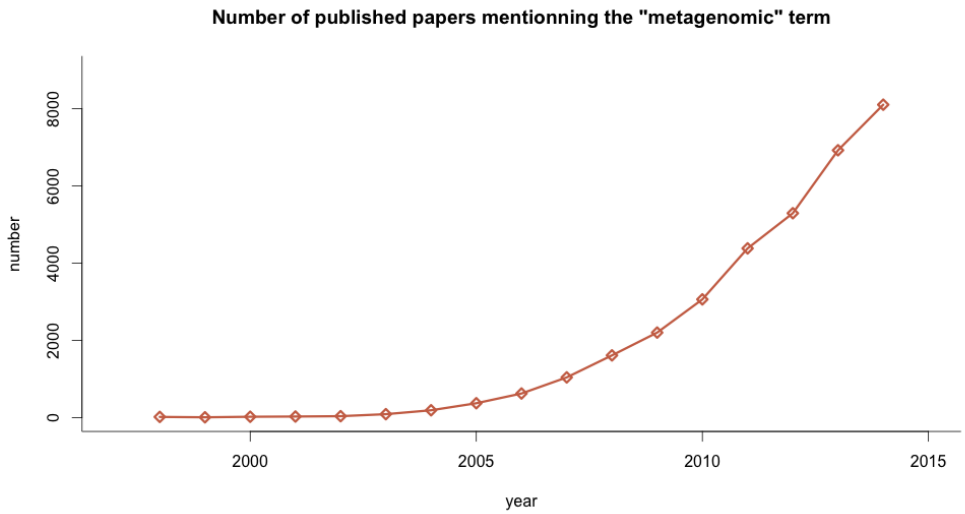


FIGURE 3.2 – Nombre de publications scientifiques comportant le mot "metagenomic". Statistiques issues de google scholar http://scholar.google.com

.

La métagénomique offre un moyen d'observer le comportement d'un milieu dans son ensemble et dans son état "naturel". Il est également aujourd'hui l'unique moyen d'accéder aux séquences génomiques et transcriptomiques des nombreuses espèces non cultivables en laboratoire. Ainsi, le nombre de projets métagénomiques ne cesse de croître (voir Figure 3.2). Parmi les projets d'envergure, nous avons mentionné le projet Tara dans lequel nous sommes largement impliqués.

Nous pouvons également mentionner d'autres projets métagénomiques grande échelle :

- Le projet **EMP** *Earth Microbiome project* http://www.earthmicrobiome.org se propose de collecter et d'analyser de l'ordre de 200000 métagénomes microbiens répartis autour du globe. Le but affiché de ce projet est de caractériser les espèces en présence pour créer un "*Gene Atlas*" à l'échelle planétaire. Ce projet fait intervenir des analyses métagénomiques, métatranscriptomiques et du séquençage d'amplicons (marqueurs d'espèces connues). Le projet a pour ambition d'aller jusqu'à la reconstruction de génomes de microbes (caractérisés par

des *unités taxonomiques**), de proposer des modèles *métaboliques** et d'offrir des outils de visualisation et d'analyse de données de ces informations [Gilbert et al., 2014].

- Le projet **GOS** : Global Ocean Sampling. L'expédition global ocean sampling (GOS) a été menée par le laboratoire JCVI (*J. Craig Venter Institute*). Durant cette expédition, des échantillons d'eau de mer ont été prélevés dans 41 lieux différents, entre l'océan atlantique nord, le canal de panama et le sud de l'océan pacifique. Chacun des prélèvements a été séquencé et l'étude a fourni 44 métagénomes représentatifs d'une partie des océans du globe (3 métagénomes provenant de sources extérieures au projet).
  Dans l'une des études associées au projet [Rusch et al., 2007], les auteurs ont procédé à une comparaison brute des métagénomes sur la seule base de leur contenu en séquences. Cette comparaison nous a servi de référence pour estimer les qualités et défaut des outils de comparaisons de métagénomes que nous avons développés, Compareads et Commet, présentés dans ce qui suit.
- Le projet **HMP** : *Human Microbiome Project* [Proctor, 2015]. Ce projet, démarré en 2008, a pour mission de générer des ressources pour permettre une caractérisation du microbiome humain et de comprendre son rôle dans la santé. Le microbiome humain est composé de tous les microorganismes vivant en association avec le corps humain. Ils sont composés de petits eukaryotes, d'archées, de bactéries et de virus. Il est impressionnant de constater que le nombre moyen de bactéries d'un individu représente environ dix fois le nombre de cellules de cet individu. Les microorganismes représentent 1 à 3% de la masse d'un individu adulte.
  Entre autres buts affichés, le projet HMP collecte 3000 jeux de lectures métagénomiques microbiennes. Ces données sont publiquement disponibles `http://hmpdacc.org/`.
- Le projet **Metasoil**. Ce projet a pour but d'étudier la composition microbienne d'un sol de référence, le *Park Grass Experiment* de Rothamsted, Angleterre. La comparaison des différents métagénomes prélevés dans ce sol permet de mieux comprendre l'adaptation des microorganismes aux changements subis (saisons, profondeurs, etc) et d'étudier le rôle de certains microorganismes dans différents écosystèmes [Delmont et al., 2011].
  Les données issues de ce projet sont, au regard de projets tels Tara Oceans ou HMP, de taille modeste car limitées à quelques dizaines d'échantillons. Elles nous ont servi à tester et étalonner l'outils Compareads, présenté dans ce qui suit.

Une caractéristique commune de ces projets métagénomiques d'envergure tient dans le fait qu'ils génèrent un nombre important de jeux de données distincts. Connaitre la similarité entre chaque paire de jeux de données constitue une information exploitable. Ceci a donné naissance à la métagénomique comparative, présentée dans ce qui suit.

3.2 Brève introduction à la métagénomique comparative

À l'image de l'analyse des données issues du projet Tara Oceans, dans des jeux de données pas ou peu assemblables et mappables, une possibilité consiste à utiliser la similarité entre ces jeux de données de séquençage, sous forme d'ensembles de lectures, pour extraire de l'information biologique. Il s'agit de "Métagénomique comparative".

La métagénomique comparative vise donc à comparer plusieurs métagénomes entre eux. Comparer deux métagénomes ou plus est un moyen efficace de comprendre les relations entre les

différences génomiques d'une communauté et les facteurs physico-chimiques d'un écosystème. Par exemple, [Ishii et al., 2009] ont comparé différents métagénomes impliqués dans la dénitrification de plusieurs sols différents. Cette approche informe sur la manière dont l'environnement agit sur différents métagénomes [Wooley et al., 2010]. La métagénomique comparative permet aussi de regrouper différents métagénomes sur la base de leurs contenus. Dans une étude du *Global Ocean Sampling expedition* (GOS), les métagénomes ont été analysés sur la base de leurs contenus en séquences puis clusterisés. Les résultats montrent que les métagénomes provenant d'endroits proches géographiquement ou partageant des facteurs environnementaux communs tendent à se regrouper ensemble [Rusch et al., 2007].

* *

*

La comparaison de métagénomes peut s'effectuer selon diverses échelles, soit en utilisant des données connues (des marqueurs) soit purement *de novo*.

Comparaison de marqueurs Une possibilité consiste à détecter des marqueurs identifiables et spécifiques à certains organismes. Dans cette optique, les informations des ARN 16s sont utilisées pour comparer la composition de plusieurs métagénomes [Jaenicke et al., 2011]. Les séquences des ARN 16s ont une vitesse d'évolution relativement lente. La détection de ces ARN permet d'être à la fois générique (présents dans la plupart des organismes) et spécifique (les singularités d'une copie d'ARN 16s permettent d'identifier l'espèce associée). Diverses méthodes permettent d'utiliser ce type de séquences pour analyser et comparer les métagénomes (MG-RAST [Port et al., 2012], MEGAN [Shakya et al., 2013], IMG/M [Cardoso et al., 2012]).

Ces méthodes ont l'avantage d'être relativement simples à mettre en oeuvre et de fournir des résultats "rassurants" dans le sens où ils permettent de retrouver la présence / absence d'espèces connues, ce qui offre une facilité d'analyse et de compréhension. Cependant, les marqueurs de type 16s, bien que particulièrement efficaces, ne sont pas renseignés pour toutes les espèces en présence, loin de là [Qin et al., 2010], et ne suffisent pas toujours à différencier les espèces.

Comparaison *de novo* de séquences Des outils de comparaisons génériques peuvent être utilisés pour comparer directement les données de séquençage contenues dans les métagénomes, et ce, sans utilisation de connaissances complémentaires. Nous pouvons citer BLAT [Kent, 2002] qui serait le plus efficace dans le contexte de la métagénomique comparative. Cependant, ces méthodes, si efficaces soient elles, ne permettent pas de passer à l'échelle de métagénomes complets. La publication de BLAT mentionne qu'il faudrait 12 jours sur une machine composée de 100 processeurs pour comparer les données de séquençage de deux génomes de souris. Ainsi, l'application de BLAT sur de grosses masses de données métagénomiques n'est pas envisageable.

Le logiciel CRASS [Dutilh et al., 2012] offre une autre approche pour comparer des métagénomes. Lors d'une première phase, l'ensemble des données à comparer est assemblé, formant des contigs de références. Notons que ces contigs n'ont pas nécessairement de sens biologiques du fait des biais des assemblages sur ces données. Ces références sont utilisées lors d'une seconde phase : les lectures des divers jeux de données y sont alors mappées. L'analyse de la répartition du mapping des lectures en

fonction des contigs et des jeux de données offre alors diverses possibilités de mesures de variabilité inter-métagénomes.

Enfin le logiciel TriageTools [Fimereli et al., 2013] compare la composition en k-mers de plusieurs métagénomes. Le nombre de k-mers partagés entre jeux de données sert alors d'indicateur pour estimer la similarité / distance entre jeux de lectures métagénomiques. La méthodologie mise en oeuvre dans cet outil est proche de celle que nous utilisons dans les outils que nous avons développés dans le cadre de la métagénomique comparative et que nous présentons dans les sections suivantes.

* *

*

Dans le cadre du projet ANR *mappi*, lors des premières analyses des données Tara Oceans, les constats étaient qu'aucune des méthodes existantes ne permettait de comparer les métagénomes non assemblés Tara Oceans. Les espèces en présence n'avaient pas pour la plupart de marqueur 16S associé. Les méthodes classiques utilisant ce type de données ne pouvaient donc pas être appliquées. D'autre part, les méthodes de comparaison *de novo* des séquences ne passaient pas à l'échelle, aussi bien pour des raisons de temps d'exécution que pour des raisons d'empreinte mémoire trop élevée.

Nous avons donc proposé de nouvelles méthodes de comparaison de jeux de données métagénomiques. Ces méthodes sont présentées dans les sections suivantes. Les premières méthodes présentées Section 3.3 sont abouties, publiées et les outils issus de ces études sont utilisés en routine. La seconde section (3.5 page 60) présente les résultats préliminaires d'études en cours.

3.3 Outils de détection de lectures similaires entre jeux de séquences NGS

Face à la complexité et à la masse des données métagénomiques, toute méthodologie d'analyse bio-informatique doit être pensée pour sa simplicité et son efficacité d'application. Ainsi nous avons pensé et mis en oeuvre une méthode de comparaison de séquences métagénomiques la plus simple possible, utilisant une structure de donnée adaptée, la plus rapide possible et la plus légère possible en terme d'impact mémoire.

Commençons par définir quelques termes que nous utilisons dans ce contexte.

3.3.1 Quelques définitions

k-mer partagé Nous utilisons l'idée de k-mer partagé. Dans le contexte comparatif, par exemple entre un jeu de données A et un jeu de données B ou entre deux séquences A et B, un k-mer partagé est un k-mer (donc un mot de taille k) dont la version *forward* et/ou *reverse complement* existe dans A et dans B (voir Section 1.4.2 page 10 pour un rappel des notions de *forward* et *reverse complement*).

Séquences similaires Dans le contexte de ce travail nous considérons que deux séquences sont *similaires* dès lors qu'elles partagent au moins t k-mers non chevauchants où $t \in \mathbb{N}$ est un seuil prédéfini.

Par exemple les séquences $AACGGCATCAGGATCACGT$ et $CGGCATCAATGATCACGTA$ sont similaires pour $t = 2$ et $k = 8$ car les 8-mers $CGGCATCA$ et $GATCACGT$ sont partagés par les deux séquences.

Cette définition de similarité de séquences est particulièrement grossière. Elle n'est qu'un pâle reflet de la réelle similarité de séquences. Par exemple deux séquences A_{diff} et B_{diff} de taille 100 peuvent partager 2 31-mers par chance et différer sur tout le reste de leurs séquences, contenant ainsi 38 substitutions. À l'autre extrémité, deux séquences A_{sim} et B_{sim} de taille 100 distinctes de 2 substitutions peuvent ne partager qu'un k-mer. Ceci serait le cas par exemple si l'une des substitution se situe en position 50 et l'autre en position 75 : un seul k-mer non chevauchant peut être partagé entre les positions zéro et 49 et aucun k-mer ne pourrait être partagé entre les positions 51 et 99 du fait de la présence d'une substitution en position 75. Dans ces deux exemples, notre définition de similarité de séquence avec $t = 2$ et $k = 31$ serait contre intuitive car elle considérerait A_{diff} et B_{diff} comme similaires alors qu'elles ne le sont pas et considérerait A_{sim} et B_{sim} comme distinctes alors qu'elle sont similaires.

D'autre part on peut constater que l'ordre des k-mers partagés n'est pas nécessairement identique entre les séquences considérées comme similaires.

À l'inverse, cette notion de distance présente l'avantage d'être extrêmement simple à calculer. Il n'est pas nécessaire d'aligner deux séquences pour décider de leur similarité. Toute utilisation de *programmation dynamique** ou même calcul de nombre de substitutions est donc évité. Par la même occasion, il n'est pas nécessaire de disposer des séquences pour pouvoir décider de leur similarité, seuls leur contenu en k-mer est suffisant. Ceci offre des perspectives algorithmiques intéressantes, permettant de ne pas devoir charger en mémoire un couple de séquences pour les comparer.

Similarité de jeux de lectures Étant donnés deux jeux de lectures A et B et les paramètres t et k, nous indiquons par $(A \overrightarrow{\cap} B)$ les lectures du jeu A similaires à au moins une lecture du jeu B.

3.3.2 Mise en oeuvre algorithmique

Comme nous le verrons dans ce qui suit, les algorithmes développés pour calculer efficacement $(A \overrightarrow{\cap} B)$ sont des *heuristiques** et leur résultats peuvent contenir des lectures considérées à tort comme similaires entre deux jeux de données. Afin de garder à l'esprit cette imprécision, nous utiliserons le symbole $(A \widetilde{\overrightarrow{\cap}} B)$ pour désigner le résultat des calculs effectués avec ces méthodes heuristiques.

Vue globale de l'algorithme Les différentes étapes du calcul de $(A \widetilde{\overrightarrow{\cap}} B)$ sont les suivantes :

- Indexation des k-mers de B. La représentation canonique de chaque k-mer des séquences B est stockée dans un index. Cet index permet de répondre à la requête suivante en temps constant : étant donné un k-mer quelconque, existe-t-il dans B ?
- Pour chaque lecture l de A :
  - Pour chaque k-mer de l, existe-t-il dans B (en requêtant la version canonique de ce k-mer dans l'index) ?

  - Si oui, incrémente le nombre de k-mers partagés pour l et teste le prochain k-mer non chevauchant
  - Si non, teste le k-mer apparaissant à la position suivante.
- Si le nombre de k-mers partagés est égal ou dépasse t alors, l est ajoutée à l'ensemble $(A \overset{\sim}{\cap} B)$.

Structure d'indexation Le temps et la mémoire utilisés par les outils dépendent grandement de la structure d'indexation utilisée. Nous avons donc chercher à définir notre propre structure de données, adaptée à ce type de requêtes et d'indexation.

Nous nous sommes basés sur le filtre de Bloom [Bloom, 1970]. Le filtre de Bloom est une structure de données probabiliste. Chaque mot à indexer est associé à un entier h obtenu à partir d'une fonction de hashage $f : \Sigma^* \to \mathbb{N}$. Cette fonction n'est pas bijective, ainsi deux mots distincts peuvent avoir la même valeur de hashage.

Un vecteur de bit est initialisé à '0'. À chaque mot indexé est associée une adresse dans ce vecteur à partir de la valeur de hashage de ce mot. Le bit correspondant est alors fixé à '1'. Lors de la requête, si le bit correspondant à l'adresse d'un mot est égal à '0', alors ce mot n'était pas indexé. À l'inverse, si le bit est égal à '1', alors soit le mot était indexé, soit le bit à cette adresse avait été fixée à '1' par un autre mot ayant la même valeur de hashage.

En pratique plusieurs (disons p) fonctions de hashage sont associées à chaque mot, et ainsi plusieurs bits du vecteur sont fixés à '1'. Lors de la requête il suffit qu'un seul des p bits testés soit égal à '0' pour être certain que le mot requêté n'était pas indexé dans le jeu de données.

Ainsi, et pour résumer, un filtre de Bloom est une structure simple (un vecteur de m bits, où m est paramétrable). Une réponse négative à une requête indique nécessairement que le mot requêté n'était pas indexé. Lors d'une réponse positive, il existe un risque ϵ de faux positif : le filtre répond "oui" alors que le mot requêté n'était pas indexé.

Il existe une approximation asymptotique du taux de faux positif qui est $\epsilon = 0.6185^{\frac{m}{n}}$, avec m la taille du vecteur et n le nombre d'éléments insérés, en utilisant $f = \ln 2 \cdot (m/n)$ fonctions de hashage différentes [Broder and Mitzenmacher, 2004]. Les filtres de Bloom utilisent peu de mémoire : $(n \log_2 e \cdot \log_2(1/\epsilon))$ bits sont nécessaires pour stocker n éléments avec une probabilité de faux positif ϵ ([Broder and Mitzenmacher, 2004]).

* *

*

Nous avons cherché à améliorer la structure pour l'adapter à notre cas, à savoir l'indexation et le requêtage de mots successifs sur une séquence. Par exemple lors de l'indexation de la séquence $AATTCAGCAGT$ avec $k = 8$, nous indexons consécutivement les k-mers $AATTCAGC$, $ATTCAGCA$, $TTCAGCAG$, et finalement $TCAGCAGT$. Ces mots ne sont pas indépendants, chacun débute par le suffixe de taille $k - 1$ du précédent. Nous avons donc mis en oeuvre des fonctions de hashage simples pouvant être mises à jour lors du calcul d'un k-mer apparaissant position $i + 1$ sachant que la valeur de hashage avait déjà été calculée pour le k-mer apparaissant position i. Cette simple modification permet de changer la complexité de $O(|S| * k)$ à $O(|S|)$ lors de l'indexation ou du requêtage des k-mers d'une séquence S.

Cette modification contraint à utiliser des vecteurs de bit dont la taille est définie par k. Cette taille (2^{k-1} octets) est convenable pour une plage d'utilisation de k-mers de taille de l'ordre de 30 à 34. En dessous, le taille des vecteurs ne garantit pas un taux de faux positifs ($\epsilon = 0.6185^{\frac{2^{k+1}}{n}}$) bas. Au dessus, la mémoire nécessaire au stockage de ces vecteurs devient trop importante pour des machines classiques. Par exemple, pour $k = 35$, la mémoire nécessaire est de 16 Go. En pratique nous avons utilisé $k = 33$ ce qui permet d'utiliser 4 Go de mémoire tout en limitant le taux de faux positifs à 0.144% lorsqu'un milliard de k-mers sont indexés.

Les détails de cette structure de données, appelée BDS (pour "*Bloom Data Structure index*"), sont présentés dans la première publication associée à ce sujet [Maillet et al., 2012].

* *

*

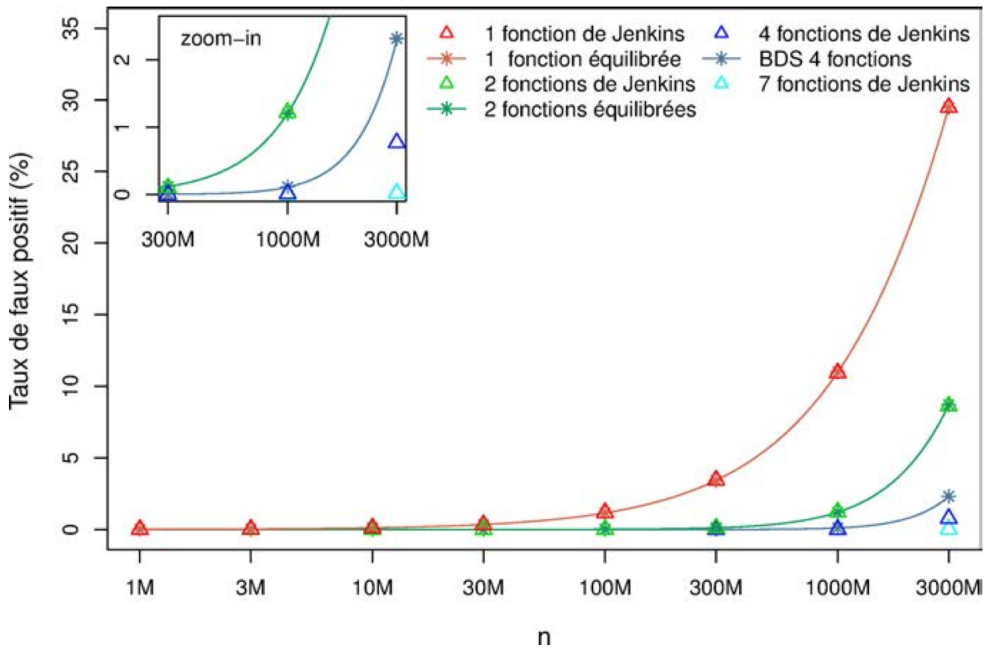


FIGURE 3.3 – Image tirée de [Maillet, 2013]. Taux de faux positifs en fonction de la méthode d'indexation utilisée et du nombre de mots indexés (n). La courbe "BDS 4 fonctions" est la solution utilisée dans les méthodes développées.

Dans le travail de thèse de Nicolas Maillet, les temps associés à l'indexation ou au requêtage de la BDS et d'autres structures d'indexation de référence de l'état de l'art ont été comparés. Les tests ont montré que notre structure d'indexation permet de gagner un ordre de grandeur sur la vitesse d'indexation ou de requêtage par rapport à l'utilisation d'un filtre de Bloom classique. Ceci n'est pas négligeable lorsque plusieurs milliards de telles opérations doivent être effectuées. Avec la BDS, près de 50 millions de requêtes peuvent être effectuées par seconde.

En revanche, comme présenté Figure 3.3 page précédente, le taux de faux positifs de la BDS est d'environ 10 fois supérieur au taux de faux positifs observé avec un filtre de Bloom classique (visible dans cette figure par les résultats de la courbe "4 fonctions de Jenkins" à comparer à la courbe "BDS 4 fonctions"). Bien que ce taux de faux positifs soit 10 fois plus élevé, il reste négligeable au regard de notre problématique, et le gain significatif au niveau temps d'exécution incite à utiliser notre structure d'indexation.

Faux positifs "de séquence" et comparaison complète de deux jeux de données A et B En utilisant $t > 1$, notre méthode peut identifier une séquence comme similaire bien qu'elle ne respecte pas strictement la similarité définie Section 3.3.1 page 48. En effet, la méthode décrite dans le paragraphe "*Vue globale de l'algorithme*" identifie les séquences de A partageant t k-mers avec les séquences de B indexées. Or, ceci est moins strict que trouver les séquences de A partageant au moins t k-mers avec au moins une séquence de B. En effet, quand on calcule $A \overset{\rightsquigarrow}{\cap} B$, les t k-mers partagés de A peuvent être répartis sur plusieurs séquences de B. Autrement dit, un faux positif existe pour $t > 1$ quand les t k-mers partagés d'une séquence requête se trouvent sur au moins deux séquences distinctes du jeu indexé (voir Figure 3.4). Ce type de faux positif est appelé *faux positif de séquence*. Avec notre structure d'indexation, cet effet est inéluctable, lorsque l'on indexe un k-mer dans le filtre de Bloom, on perd l'information liant un k-mer et sa séquence d'origine : on ne peut pas savoir si deux k-mers distincts proviennent de la même séquence ou non.

* *

*

Lors de la comparaison complète de deux jeux de données A et B, les faux positifs de séquences peuvent être limités. En pratique pour comparer dans les deux sens A et B, nous ne nous limitons pas à effectuer les deux opérations $A \overset{\rightsquigarrow}{\cap} B$ puis $B \overset{\rightsquigarrow}{\cap} A$. Nous utilisons l'idée suivante : nous avons $\left(A \overset{\rightsquigarrow}{\cap} (B \overset{\rightsquigarrow}{\cap} A)\right) \subseteqq \left(A \overset{\rightsquigarrow}{\cap} B\right)$ car l'ensemble des k-mers présents dans l'ensemble $(B \overset{\rightsquigarrow}{\cap} A)$ est plus petit que l'ensemble des k-mers présents dans B. Ainsi, nous avons $\left(A \overrightarrow{\cap} B\right) \subseteqq \left(A \overset{\rightsquigarrow}{\cap} (B \overset{\rightsquigarrow}{\cap} A)\right) \subseteqq \left(A \overset{\rightsquigarrow}{\cap} B\right)$.

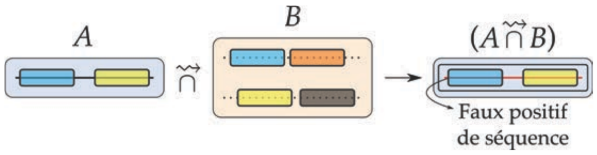


FIGURE 3.4 – Image tirée de [Maillet, 2013]. Représentation d'un faux positif de séquence. Pour faciliter la vision des faux positifs de séquence, les k-mers partagés qui sont indexés sont ici représentés sur leur lecture d'appartenance en pointillés gris. Avec $t = 2$, la lecture représentée dans le jeu A n'a aucune lecture similaire dans le jeu B avec laquelle elle partage au moins 2 k-mers. Pourtant, cette lecture partage 2 k-mers avec l'ensemble du jeu B et notre méthode détectera à tort cette lecture dans le jeu $A \overset{\rightsquigarrow}{\cap} B$. C'est un faux positif de séquence.

Donc l'ensemble $\left(A \overset{\rightsquigarrow}{\cap} (B \overset{\rightsquigarrow}{\cap} A)\right)$ est plus proche que $\left(A \overset{\rightsquigarrow}{\cap} B\right)$ du résultat attendu $\left(A \overrightarrow{\cap} B\right)$.

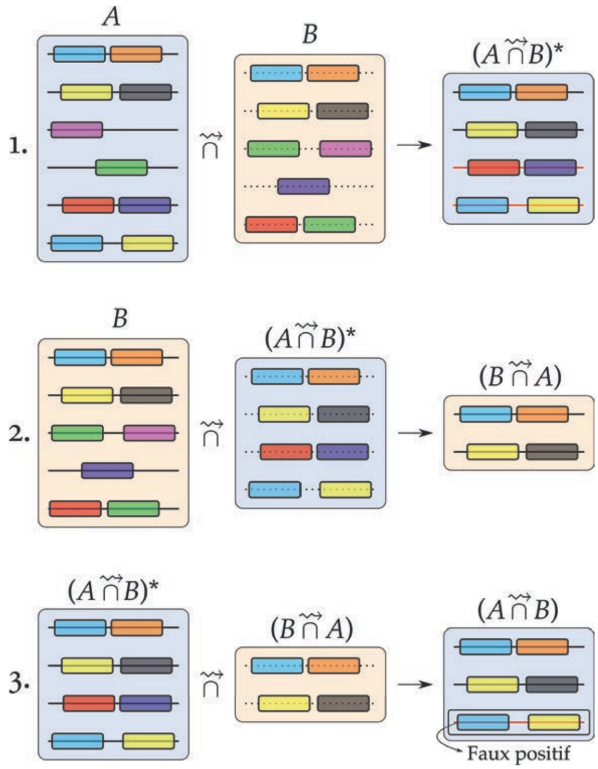


FIGURE 3.5 – Représentation des trois étapes principales permettant de comparer les jeux A et B

Concrètement nous exploitons cette idée en appliquant le processus suivant pour comparer dans les deux sens les jeux de lectures A et B.

- Calculer $\left(A \overset{\rightsquigarrow}{\cap} B\right)$;
- Calculer $set_{AinB} = B \overset{\rightsquigarrow}{\cap} \left(A \overset{\rightsquigarrow}{\cap} B\right)$;
- Calculer $set_{BinA} = \left(A \overset{\rightsquigarrow}{\cap} B\right) \overset{\rightsquigarrow}{\cap} \left(B \overset{\rightsquigarrow}{\cap} \left(A \overset{\rightsquigarrow}{\cap} B\right)\right)$.

Au final, les résultats de la méthode sont les ensembles set_{AinB} et set_{BinA} issus respectivement de la deuxième et de la troisième étape. La Figure 3.5 représente ces différentes étapes et offre une représentation graphique permettant de comprendre pourquoi cette triple comparaison permet d'éliminer des faux positifs de séquences.

Les faux positifs de séquence restants sont caractérisés par t k-mers partagés sur au moins deux lectures distinctes du jeu indexé, elles-mêmes considérées comme similaires à des lectures du jeu requête. Bien que cet effet soit difficile à quantifier, les tests (cf [Maillet, 2013]) montrent que cette méthode permet d'obtenir des résultats fiables et très similaires à ceux obtenus par des méthodes plus classiques sur des jeux de données réels.

* *

*

Pour résumer, la méthode que nous proposons a été pensée pour être la plus rapide et la plus légère possible en terme d'impact mémoire. Les techniques utilisées souffrent de la présence de faux positifs, à la fois dus à la structure d'indexation basée sur l'idée du filtre de Bloom et dus au fait que le lien entre k-mer et lecture associée(s) est perdu. Nous verrons dans les résultats, que le bruit induit par ces faux positifs n'est pas rédhibitoire et que l'application des outils implémentés et présentés dans la section suivante sur des données biologiques fournit des résultats exploitables pour obtenir de l'information biologique pertinente.

3.3.3 Implémentation : les outils Compareads et Commet

Deux outils implémentent les méthodes précédemment décrites. Il s'agit de Compareads [Maillet et al., 2012] et de Commet [Maillet et al., 2014]. Ils ont été développés dans l'idée d'être le plus simple possible à analyser. Leurs opérations de base consistent à filtrer des lectures et à comparer les jeux de lectures selon les méthodes présentées dans la section précédente. Le filtrage des lectures est effectué selon certains critères de qualité (quand les données de qualité associées à chaque lecture sont présentes), de complexité (les séquences de faible complexité telle ATATATATATAT... sont supprimées) ou de longueur.

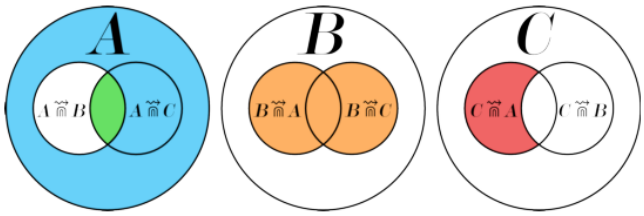


FIGURE 3.6 – Image tirée de [Maillet et al., 2014]. Exemples d'opérations possibles grâce à l'utilisation de la représentation de résultats de comparaison de métagénomes sous forme de vecteurs de booléens. Trois métagénomes A, B et C sont représentés. L'ensemble bleu correspond à l'ensemble $A\ ET\ NON(set_{AinB})$, le vert à $(set_{AinB})\ ET\ (set_{AinC})$, le orange à $(set_{BinA})\ OU\ (set_{BinC})$, et enfin le rouge correspond à $(set_{CinA})\ ET\ NON(set_{CinB})$. La représentation $A \overrightarrow{\Cap} B$, empruntée à l'article [Maillet et al., 2014], représente ce que nous avons noté set_{AinB} dans ce présent manuscrit.

La seconde implémentation, Commet, a bénéficié d'un important travail d'ingénierie et a été entièrement recodée.

Cette nouvelle version a permis de factoriser les phases d'indexation lorsque plus de deux métagénomes doivent être comparés deux à deux.

En outre, la version Commet a bénéficié d'un nouveau format de représentation des résultats. Pour représenter par exemple l'ensemble set_{AinB}, plutôt que de créer un fichier au format FASTA ou FASTQ (voir Section 1.3 page 4) pouvant prendre au pire autant de place que le fichier A, un vecteur de booléens BV *(boolean vector)* est créé, par exemple dans un fichier $set_{AinB}.bv$. Le vecteur

contenu dans $set_{AinB}.bv$ contient exactement autant de valeurs (0 ou 1) qu'il y a de lectures dans A. Pour chacune de ces lectures, si elle appartient à set_{AinB} alors la valeur correspondante dans le vecteur est fixée à 1, sinon elle est fixée à 0. Notons que cette représentation permet de stocker de la même manière les lectures non-filtrées.

Tous les outils de Commet sont donc capables d'utiliser et de fournir des données stockées sous ce format. Un module permet également de retrouver les lectures au format classique (FASTQ ou FASTA) à partir de telles représentations.

Cette représentation en vecteurs de booléens présente plusieurs avantages. Bien entendu, l'espace de stockage des résultats finaux et intermédiaires est limité à son strict minimum. Un sous ensemble d'un fichier de cent millions de lectures est stocké sur 12 Mo de disque, ce qui est dérisoire comparé au stockage des fichiers FASTQ originaux. Ceci est particulièrement important lorsque N métagénomes sont à comparer deux à deux. Comparer N métagénomes nécessite d'effectuer environ N^2 comparaisons générant chacune un fichier de résultat. Ainsi, sans utiliser de représentation sous forme de vecteurs de booléens, l'espace disque nécessaire est quadratique par rapport à l'espace disque nécessaire au stockage des données initiales. Ceci est bien entendu inacceptable sur de gros projets où plusieurs centaines de jeux de données doivent être comparés.

Un second avantage de taille va de pair avec cette représentation : il devient extrêmement aisé et rapide d'appliquer des opérations booléennes sur ces vecteurs. Ceci est particulièrement utile pour combiner différents résultats de comparaisons de métagénomes. Nous proposons Figure 3.6 page ci-contre une représentation graphique de quelques possibilités d'utilisation de ces opérations booléennes. Ceci est particulièrement intéressant par exemple pour supprimer des contaminants (ensemble rouge) ou pour ne conserver que les lectures présentes dans un jeu de données et dans l'un ou l'autre de deux autres jeux de données alors considérés comme *poolés** (ensemble orange) ou encore pour conserver les lectures présente dans un jeu de données et dans les deux autres jeux de données, permettant ainsi d'accéder aux lectures d'espèces présentes dans les trois jeux de données considérés (ensemble vert). Toute la combinatoire des comparaisons possibles avec N jeux de données est bien entendu possible. Ces opérations, consistant simplement à des opérations binaires sur des vecteurs de bits sont effectuées en quelques micro-secondes.

3.4 Résultats

3.4.1 Passage à l'échelle

Nous avons effectué divers tests afin d'évaluer la capacité de nos outils à passer à l'échelle des données générées par les gros projets de métagénomique. Ceux-ci, présentés en détails dans le manuscrit de thèse de Nicolas Maillet, ont permis de mettre en lumière le fait que

- l'approche que nous proposons permet d'aller beaucoup plus loin que les autres approches envisageables (BLAT, Triage-Tools, CRASS). Soit ces méthodes sont limitées par la mémoire ou leur temps d'exécution soit leurs résultats sont invraisemblables.
- les approches que nous proposons permettent de comparer quelques dizaines de métagénomes. Cependant, elles restent des méthodes de comparaisons deux à deux (même si certaines phases d'indexation sont factorisées entre les différents processus). Ainsi, le nombre de comparaisons à effectuer est quadratique. Les très gros jeux de données issus de projets de l'ampleur de

Tara Oceans ne peuvent être traités dans leur intégralité sans nécessiter de très gros moyens de calculs.

3.4.2 Résultats sur des données réelles

Application aux données GOS *Global Ocean Sampling*

Les 44 métagénomes issus du projet *Global Ocean Sampling* ont été analysés avec Compareads. Les échantillons contiennent, en moyenne, 174 759 séquences de 1 249 nucléotides en moyenne. Il s'agit d'un "petit" jeu de données, issu de séquenceurs précédant l'apparition des technologies NGS (technologie sanger). Les 990 intersections ont été calculées avec Compareads ($t = 4$ et $k = 33$) en 72 heures et 30 minutes. En moyenne, une intersection a pris 4 minutes et 23 secondes.

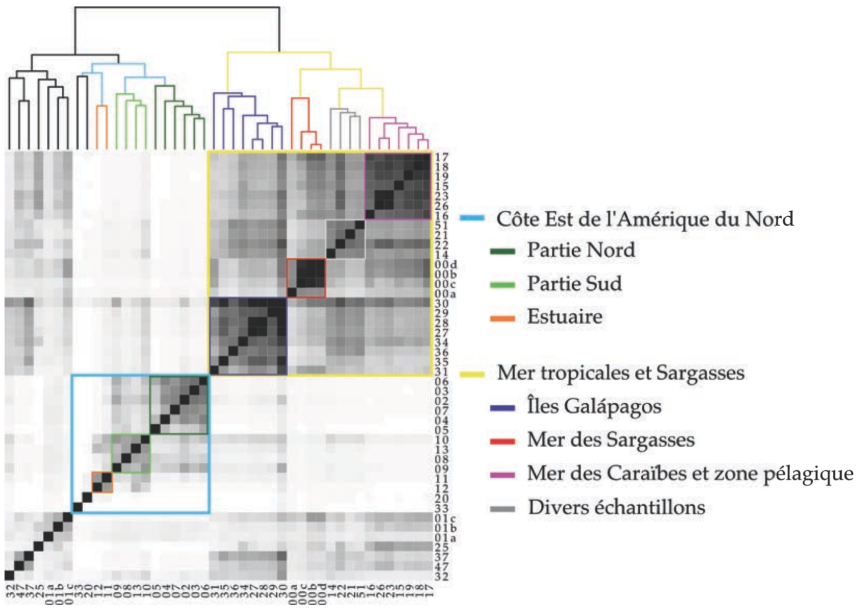


FIGURE 3.7 – Image tirée de [Maillet, 2013]. Représentation de la similarité entre les 44 échantillons de GOS obtenus avec Compareads en recherchant 4 33-mers partagés entre chaque séquence. Les intersections en noir représentent un score de similarité de plus de 50% et les intersections en blanc représentent un score de similarité de 0%.

Les résultats sont présentés Figure 3.7. Deux groupes principaux sont bien établis. Le premier, représenté en bleu, regroupe presque tous les échantillons venant d'eaux tempérées de la côte est de l'Amérique du nord. Seul l'échantillon 14, comme dans la publication d'origine, n'est pas inclus dans cet ensemble. Ce groupe contient aussi deux échantillons très différents de tous les autres. Le premier provient d'eau douce et le second d'eau hypersaline. On peut diviser ce premier groupe principal en trois sous-parties. La première, en vert foncé, regroupe les échantillons provenant de la partie nord des États-unis et le groupe en vert clair, de la partie sud. Le troisième groupe, en orange, contient les deux échantillons venant d'estuaires. Ces trois groupes sont identiques à ceux de la publication d'origine.

Le second groupe principal coloré en jaune, contient les échantillons tropicaux et ceux de la mer des Sargasses. La sous-partie en violet foncé regroupe des échantillons provenant exclusivement des Galápagos. La sous-partie rouge contient les échantillons de la mer des Sargasses. Dans la publication d'origine, l'échantillon $00a$ n'est pas dans ce groupe. Selon les métadonnées, le groupe en gris, comme dans la publication d'origine, contient des échantillons venant de plusieurs endroits. Finalement, le groupe en magenta contient des échantillons de zones pélagiques et des caraïbes, comme dans la publication d'origine.

Ces résultats montrent que Compareads est capable de classer de nombreux métagénomes en fonction de leurs lieux de prélèvements, en utilisant seulement des informations de séquence. La provenance géographique générale (tropique ou Amérique du nord) permet de séparer en deux les échantillons. La localisation plus précise (Galápagos, sud des États-unis, etc) est aussi retrouvée.

Étude préliminaire sur les données Tara Oceans

Nous avons appliqué Compareads sur un sous-ensemble des données issues de stations de l'expédition Tara Oceans. Depuis Compareads puis Commet ont été appliqués à plus grande échelle sur les données Tara. Cependant, ces études étant en cours, les résultats présentés dans ce document se limiteront à la présentation de résultats sur les stations du Mozambique.

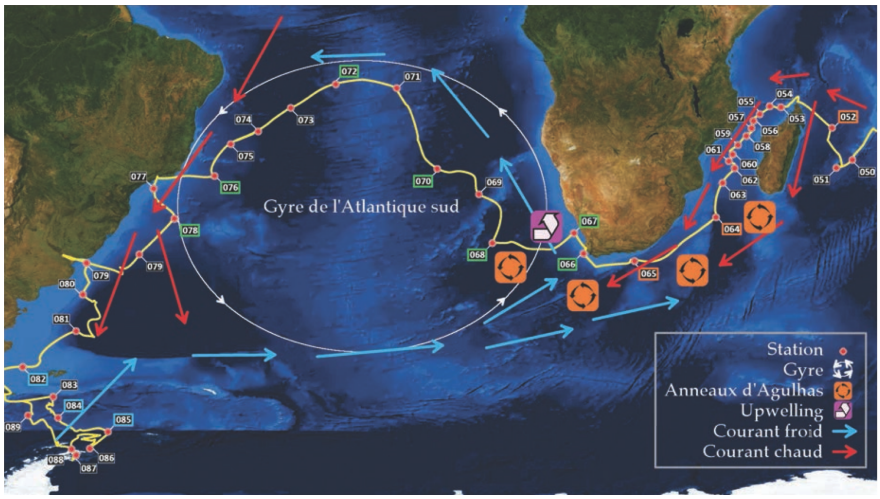


FIGURE 3.8 – Prélèvements effectués au cours de l'expédition Tara Oceans dans l'atlantique sud

D'un point de vue métagénomique, les stations dans le canal du Mozambique visent à en déterminer la composition en organismes (stations 52, 64 et 65, en orange sur la Figure 3.8). Ensuite, l'expédition a échantillonné un tourbillon en formation au sud du Cap (station 66). Finalement, la goélette a suivi la course des anneaux d'Agulhas dans le gyre de l'atlantique sud pour déterminer à quel point ces tourbillons influent sur la circulation des microorganismes. Le long de la côte ouest sud-africaine, les eaux froides du courant de Benguela produisent un phénomène nommé *upwelling* (remontée d'eau en français) : le long du courant du Benguela, des eaux profondes et froides remontent en surface et charrient de nombreux microorganismes. La station 67 est située dans un site d'upwelling.

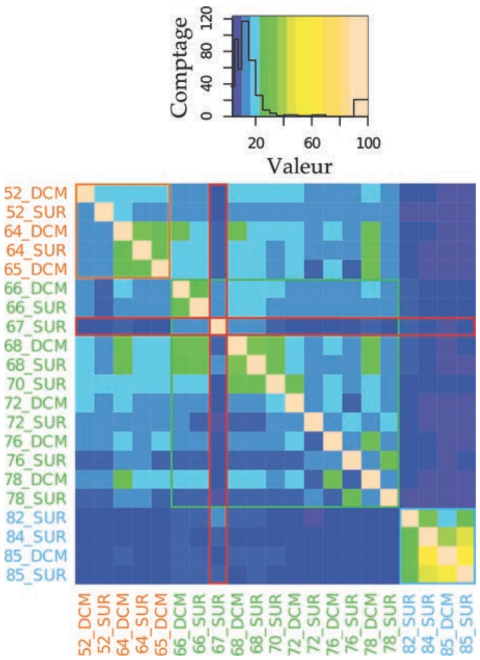


FIGURE 3.9 – *Heat map* des intersections de 13 stations de Tara Oceans pour des organismes d'une taille de 0,8 μm à 5 μm. Les stations dont la légende est en orange proviennent du canal du Mozambique, les stations dont la légende est en vert du gyre de l'atlantique sud et les stations dont la légende est en bleu, de l'océan austral. La station 67 a été réalisée dans un site d'upwelling.

Les résultats de l'application de Compareads sur ces données sont présentés sous forme de *Heat map* (Figures 3.9 et 3.10 page ci-contre). La *Heat map* offre une représentation de la similarité de chaque couple d'échantillon. Chaque couleur représente la similarité de l'échantillon en ligne (disons A) dans l'échantillon en colonne (disons B). Il s'agit de $\frac{|set_{AinB}|}{|A|} \times 100$. La matrice est asymétrique car lors de la comparaison de deux échantillons A et B il se peut que beaucoup de lectures de A soient similaires à des lectures de B (A est alors fortement similaire à B), et que l'inverse ne soit pas vrai (B est peu similaire à A).

La heat map représentée Figure 3.9 a été obtenue avec des organismes d'une taille de 0,8 μm à 5 μm, contenant des bactéries et des petits eucaryotes. Sur cette figure, on constate que les échantillons provenant de l'océan austral n'ont que très peu de séquences similaires avec les échantillons d'océan atlantique ou indien. Il semble ainsi que, pour les organismes de la taille étudiée, il n'y ait pas une frontière nette de répartition entre l'océan indien et l'océan atlantique. Par contre, l'océan austral a un contenu en ADN très différent des deux autres océans étudiés.

Un autre point intéressant sur cette figure est la station 67 (encadrée en rouge), clairement différente de toutes les autres. Elle ne compte que très peu de séquences similaires avec les autres échantillons et semble donc contenir un écosystème différent. Pour rappel, le prélèvement a été réalisé dans un site d'*upwelling*. Il est connu que les sites d'*upwelling* sont particulièrement riches en organismes vivants, ce qui explique probablement une telle différence au niveau ADN entre cette

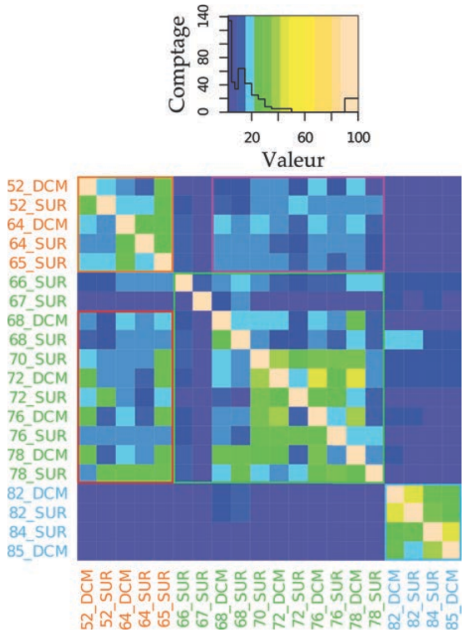


FIGURE 3.10 – *Heat map* des intersections de 13 stations de Tara Oceans pour des organismes d'une taille de 180 μm à 2000 μm. Le code couleur est donné dans la légende de la Figure 3.9 page ci-contre. De plus, le rectangle délimité en rouge représente le pourcentage de séquences de l'océan atlantique sud similaires à des séquences de l'océan indien. Le rectangle délimité en mauve représente le pourcentage de séquences de l'océan indien similaires à des séquences de l'océan atlantique sud.

station et les autres.

La *Heat map* représentée sur la Figure 3.10 a été obtenue avec des organismes d'une taille de 180 μm à 2000 μm, contenant des protistes, du zooplancton et d'autres eucaryotes. Comme sur la Figure 3.9 page précédente, les échantillons provenant de l'océan austral n'ont que très peu de séquences similaires avec les échantillons d'océan atlantique ou indien. De même la station 67 est clairement différente de toutes les autres stations.

L'observation la plus intéressante que l'on peut faire sur cette figure est délimitée par le rectangle rouge et le rectangle mauve. Le rectangle rouge représente le pourcentage de séquences de l'océan atlantique sud similaires à des séquences de l'océan indien et le rectangle mauve représente le pourcentage de séquences de l'océan indien similaires à des séquences de l'océan atlantique sud. On constate que le rectangle mauve contient 7 résultats supérieurs à 15% de similarité, mais aucun supérieur à 20%. Le rectangle rouge contient 19 valeurs supérieures à 15% dont 12 supérieures à 20% de similarité. Dès lors, on peut dire que pour la taille étudiée, une partie des séquences de l'océan atlantique sud est similaire à des séquences de l'océan indien, mais une moindre partie de séquences de l'océan indien est similaire à des séquences de l'atlantique sud. Une partie de l'ADN du zooplancton et des protistes de l'océan indien est donc retrouvée dans l'océan atlantique sud, mais l'inverse est moins fréquent. Une hypothèse est que cet ADN est transporté, depuis le Mozambique vers l'atlantique sud, par les anneaux d'Agulhas.

Ces premiers résultats de Compareads sur les données de Tara Oceans semblent indiquer qu'un mélange s'opère entre les eaux de l'océan indien et de l'océan atlantique, peut-être au travers des anneaux d'Agulhas. Ces résultats tendent à indiquer que les grands systèmes océaniques influent sur la répartition globale des microorganismes marins.

Ces études ont été exploitées et diverses sources d'information ont été croisées pour mettre en évidence ces hypothèses. Ces résultats sont restitués dans la publication [de Vargas et al., 2015].

* *

*

Ces études placent exactement Compareads et Commet dans leur contexte : étudier des données pour émettre des hypothèses ou pour permettre aux bio-analystes de localiser où focaliser leurs recherches pour être le plus efficace possible.

3.5 Outils de comparaisons d'ensembles de k-mers similaires entre jeux de séquences NGS

Le coeur des méthodes présentées précédemment, et implémentées à travers des outils Compareads et Commet, s'applique uniquement à la comparaison deux à deux de métagénomes. Même performantes, de telles approches ne permettent pas de passer à l'échelle de très gros projets métagénomiques à l'image du projet Tara Oceans où près d'un millier de métagénomes sont attendus. En utilisant Commet sur ces données, à raison d'environ 10h par comparaison entre deux métagénomes sur un processeur 50 coeurs, pas moins de 578 années seraient nécessaires pour faire la comparaison entre tous les métagénomes. Il est donc nécessaire de concevoir des méthodes spécifiques à la comparaison métagénomique "all-vs-all".

3.5.1 Historique

Dans cette optique, nous avons proposé avec Guillaume Holley, lors d'un stage de M2, les premières pierres d'une analyse basée sur la répartition de k-mers entre les métagénomes. Nous avons proposé une approche basée sur des méthodes statistiques ($KPCA^*$) associées à des outils d'*algorithmes génétiques** pour l'estimation des paramètres (algorithme du "*recuit simulé*"). L'approche consistait à créer une matrice *aussi grosse que possible* dont les lignes sont des k-mers et dont les colonnes sont des jeux de données. Chaque entrée de la matrice contient l'abondance d'un k-mer dans un jeu de données. La taille des matrices était essentiellement contrainte par les méthodes statistiques appliquées par la suite sur ces données et se limitait à quelques centaines de milliers ou millions de k-mers.

Les travaux entamés lors de ce stage ont pu servir de preuve de concept à ce changement d'échelle passant de la comparaison de lectures à la comparaison de mots de taille k, largement inférieure à la taille de celles-ci.

De plus, les travaux entamés durant ce stage ont permis de fédérer un petit groupe de chercheurs réunissant des experts de données Tara Oceans (producteurs et analystes de ces données) et statisticiens. Ceci a motivé la création du projet *Hydrogen* : "*Comparative Metagenomic for Measuring Biodiversity. Application to Ocean Life Studies*".

3.5.2 Outil de comparaison multiple de métagénomes

Au sein du projet Hydrogen, nous développons de nouvelles approches pour la comparaison multiple de métagénomes. La méthode en cours de développement est particulièrement prometteuse. Elle est basée sur l'exploitation des données d'abondance de l'ensemble des k-mers de l'ensemble des métagénomes à comparer. L'idée est de générer une matrice intégrant l'ensemble des k-mers distincts issus des jeux de données, et pour chacun d'entre eux, d'en connaitre l'abondance dans chacun des jeux de données. L'exploitation de ces valeurs d'abondance permet alors d'estimer les distances deux à deux de ces jeux de données.

Plusieurs mesures sont en cours de mise en place. Pour chaque paire de jeux de données, une mesure de similarité peut être basée sur l'abondance des k-mers présents dans les deux jeux de données ou basée sur le nombre de k-mers dont l'abondance dépasse un seuil dans les deux jeux de données. En outre nous étudions l'effet du sous-échantillonnage des lectures, à la fois pour valider la robustesse des mesures proposées via des méthodes de *bootstrap** mais aussi pour estimer la complexité interne des jeux de données et donc également l'*alpha-diversité** des jeux de données.

Implémentation : la méthode Simka D'un point de vue algorithmique, la méthode décrite précédemment (comparer des abondances d'occurrences de k-mers dans un ensemble d'échantillons métagénomique) peut paraître particulièrement simple. C'est vrai, cependant on ne peut pas en dire autant de la mise en oeuvre pratique d'une telle approche.

Pour commencer, il est impensable de faire tenir en mémoire une matrice d'abondance k-mers $\times$ jeux de données. Plusieurs dizaines de milliards de k-mers sont à considérer. Une matrice stockant les abondances de 10 milliards de k-mers pour mille métagénomes nécessiterait près de dix terabytes (10000 gigabytes) de mémoire. Il n'existe pas de machine accessible disposant de tant de mémoire. L'algorithme développé a été pensé pour traiter la matrice ligne par ligne et donc k-mer par k-mer. Ainsi la matrice n'est jamais stockée et l'empreinte mémoire est dérisoire.

La seconde difficulté majeure tient dans le comptage simultané des occurrences de tous les k-mers présents dans un ensemble de m jeux de données. Je ne développerai pas cette partie dans ce document car ne suis pas techniquement impliqué dans son développement. Je tiens cependant à préciser qu'il s'agit d'un défi de très grande ampleur relevé par le formidable travail de, entres autres, Guillaume Rizk, Rayan Chikhi, Erwan Drezen et Gaëtan Benoît. Au sein du projet GATB [Drezen et al., 2014] que nous avons déjà évoqué, l'outil DSK [Rizk et al., 2013] permet d'effectuer un comptage des occurrences de k-mers de manière rapide et à faible impact mémoire. Cet outil a été modifié afin de permettre son application à plusieurs jeux de données.

La méthode a été implémentée dans un outil nommé Simka (similarité de k-mers) [Benoit et al., 2015]. Lors de premiers tests, les résultats obtenus avec Simka ont été comparés à ceux obtenus avec Commet sur 21 jeux de données issus du projet Tara Oceans. Ces tests ont montré que les résultats Simka, basés uniquement sur un comptage de k-mers sont très similaires à ceux obtenus

par l'application Commet basée elle sur une similarité au niveau des lectures. Les résultats produits par les deux méthodes ont été utilisés pour regrouper les jeux de lectures par similarité et ainsi former un *clustering* de ceux-ci. Un résultat principal est que les *clusters* obtenus par l'une ou l'autre méthode sont identiques. Les temps de calcul sont, eux, très différents. Sur ces jeux de données, Commet a nécessité quelques semaines de calcul alors que Simka n'a nécessité que quatre heures pour effectuer les comparaisons.

3.6 Perspectives

Le domaine de la métagénomique me semble absolument fantastique. D'une part car la métagénomique permet d'obtenir de l'information à laquelle il serait impossible d'accéder par des méthodes de génomique, mais aussi car le paradigme change. On ne parle plus du génome d'un individu. On considère les données génomiques d'un verre d'eau de mer, d'un microbiote, d'une pelletée de terre, ou même de l'air que l'on respire [Whon et al., 2012]. Les applications sont aussi diverses que fondamentales. Les analyses de terre ou de mer permettent de comprendre le comportement de notre planète face aux modifications qu'elle subit (pollutions, modifications climatiques). Les analyses de microbiotes intestinaux apportent une ouverture nouvelle sur la compréhension de maladies et de leurs traitements.

Ce changement d'échelle applicatif passe également par un changement d'échelle de complexité. À travers les outils que nous avons proposés, nous avons apporté une pierre à l'édifice de l'analyse de ce nouveau type de données. C'est un premier pas. Il devrait y en avoir beaucoup d'autres.

Les perspectives de nos travaux à moyen terme resteront, à l'image de nos précédentes contributions, focalisées sur des méthodes utilisables sans moyens informatiques faramineux.

Nous chercherons à établir l'ensemble des caractéristiques qui pourront être prédites, simplement à partir du comptage de k-mers. Nous avons conscience que l'accès au comptage de l'ensemble des k-mers par jeu de données, est une information particulièrement précieuse. Nous allons chercher à définir comment, précisément, ce comptage peut nous renseigner sur le nombre d'espèces en présence, sur les espèces majoritairement abondantes, mais aussi sur les espèces dites "dans les queues de distributions" du fait de leur rareté. Ce comptage peut également, comme nous l'avons évoqué précédemment, nous informer sur la complexité de la biodiversité intra-métagénome.

Jusqu'alors les méthodes que nous avons proposées n'utilisent que les informations brutes des métagénomes. Nous sommes conscients que d'autres types d'informations pourraient et donc devraient être intégrées pour enrichir les résultats que nous pourrions proposer. Nous pouvons penser à coupler automatiquement les analyses purement basées sur la comparaison de séquences avec des informations complémentaires utilisants des ensembles de marqueurs connus. Les corrélations entres ces deux sources seraient un gage de qualité des résultats, alors que des différences pourraient indiquer la présence d'espèces pour lesquelles ces marqueurs sont trop peu sensibles, sinon des limitations de l'une ou l'autre méthode d'analyse.

Il nous faut également nous poser la question de l'utilisation de k-mers (de taille réduite à quelques dizaines de nucléotides) quand les lectures produites par les séquenceurs TGS peuvent atteindre plusieurs dizaines de milliers de nucléotides. Il semble que l'utilisation de k-mers seuls pour tenter l'assemblage de métagénomes devient inapproprié sur de très longues lectures. Cependant, dans

le cas de projets appliqués à du matériel génétique dégradé, ou visant à déterminer la biodiversité d'un milieu, ou visant à détecter des espèces rares ou des expressions de gènes rares, il semble peu probable que les données à forte profondeur de séquençage de type Illumina disparaissent. Il est en revanche assez probable que l'on voit grandir le nombre de projets cumulant des données de lectures courtes (NGS) avec des données de lectures longues (TGS). Il faut maintenant repenser les méthodes prenant en considération conjointement ces deux types de données.

L'assemblage des données métagénomiques est un défi de taille, aussi stimulant d'un point de vue technique que d'un point de vue applicatif. Comme nous l'avons évoqué, l'assemblage métagénomique a un niveau de difficulté largement supérieur à l'assemblage de métagénomes, déjà bien difficile lui même. À moyen terme, nous chercherons à proposer des méthodes intégratives (données hétérogènes incluant grandes et petites lectures, indications de marqueurs génomiques, données comparatives) pour proposer des assemblages exploitables. Nous pourrons nous appuyer sur nos premiers résultats basés sur une version de l'outil Minia [Chikhi and Rizk, 2013], modifié pour l'assemblage de métagénomes, et en cours d'analyse dans le cadre du challenge cami http://www.cami-challenge.org/.

Les projets présentés dans le chapitre suivant seront, je l'espère, une clef de la réussite de ce type d'intégration de données hétérogènes pour l'exploitation optimale de données de séquençage.

3.7 Présentation des publications associées

▶ Maillet, N., Lemaitre, C., Chikhi, R., Lavenier, D., and Peterlongo, P. (2012). Compareads : comparing huge metagenomic experiments. In *RECOMB Comparative Genomics 2012*, Niterói, Brazil

Cette publication (fournie en annexe page 63) présente l'outil Compareads.

▶ Maillet, N., Collet, G., Vannier, T., Lavenier, D., and Peterlongo, P. (2014). COMMET : comparing and combining multiple metagenomic datasets. In *IEEE BIBM 2014*, Belfast, United Kingdom

Cette publication (fournie en annexe page 74) présente l'outil Commet.

▶ Villar, E., Farrant, G. K., Follows, M., Garczarek, L., Speich, S., Audic, S., Bittner, L., Blanke, B., Brum, J. R., Brunet, C., Casotti, R., Chase, A., Dolan, J. R., D'Ortenzio, F., Gattuso, J.-P., Grima, N., Guidi, L., Hill, C. N., Jahn, O., Jamet, J.-L., Le Goff, H., Lepoivre, C., Malviya, S., Pelletier, E., Romagnan, J.-B., Roux, S., Santini, S., Scalco, E., Schwenck, S. M., Tanaka, A., Testor, P., Vannier, T., Vincent, F., Zingone, A., Dimier, C., Picheral, M., Searson, S., Kandels-Lewis, S., Acinas, S. G., Bork, P., Boss, E., de Vargas, C., Gorsky, G., Ogata, H., Pesant, S., Sullivan, M. B., Sunagawa, S., Wincker, P., Karsenti, E., Bowler, C., Not, F., Hingamp, P., and Iudicone, D. (2015). Environmental characteristics of Agulhas rings affect interocean plankton transport. *Science*, 348(6237) :1261447–1261447

Cette publication (dont je ne suis pas auteur) base une partie importante de ses résultats sur l'utilisation du logiciel Compareads. Il s'agit de l'une des quatre premières publications présentant

les résultats d'analyses des données Tara Oceans. Notons que les résultats basés sur l'analyse plein génome aurait été difficilement envisageable sans l'existence de l'outil Compareads.

► Benoit, G., Peterlongo, P., Lavenier, D., and Lemaitre, C. (2015). Simka : fast kmer-based method for estimating the similarity between numerous metagenomic datasets. JOBIM 2015

Première diffusion de l'outil Simka, via un poster (fourni page 80) présenté à la conférence JOBIM 2015. Ce poster a gagné l'un des deux prix du meilleur poster lors de cette conférence.

Chapitre 4

Mapper les lectures sur des graphes

Contents

4.1 Introduction 65
4.2 Mapping sur graphe 65
4.3 Solution pour le mapping intensif de séquences sur dBG 67
4.4 Perspectives pour l'amélioration du mapping sur graphe 71
4.5 Présentation des publications associées 72

Ce chapitre se situe à mi-chemin entre la présentation de travaux réalisés et une perspective de travaux futurs. Les travaux qui y sont présentés sont motivés par une thématique qui sera présentée plus en détails dans le chapitre suivant, discutant des perspectives de travaux à venir. J'ai cependant fait le choix de considérer les apports des travaux présentés ici comme un chapitre à part entière car il me semble que leur impact est potentiellement important et que les résultats préliminaires sont particulièrement encourageants.

4.1 Introduction

Prenons un peu d'avance sur une partie des perspectives présentées Chapitre 5. Dans la Section 5.1, nous motiverons l'idée de changer les habitudes qui consistent à utiliser un génome de référence sous forme de séquence(s) linéaire(s), pour lui préférer une représentation sous forme de graphe. Ceci présente des avantages techniques (meilleurs assemblages, correction et/ou compression des données, bio-analyse de données de re-séquençage). Laissons pour l'heure planer le mystère sur ces futures avancées pour nous concentrer sur une étape indispensable pour rendre utilisable et populaire une représentation graphique des données génomiques : la possibilité de mapper des séquences sur un tel graphe plutôt que sur des séquences linéaires comme évoqué Section 1.4 page 6.

4.2 Mapping sur graphe

Un prérequis indispensable à la représentation et à l'utilisation de génome de référence sous forme de graphe est que ce graphe de référence soit *requêtable*. La requête la plus évidente est la

suivante : étant donné un tel graphe de référence $\mathcal{G}$ et une séquence requête Q, existe-t-il une séquence R issue de $\mathcal{G}$ telle que Q mappe R avec une distance maximale contrainte ? Une requête similaire consiste non pas à décider de la *mappabilité* de R sur $\mathcal{G}$, mais de fournir le résultat du ou des alignements de R sur $\mathcal{G}$.

Solutions existantes Il existe des méthodes mentionnant le mapping de séquences sur des références représentées sous forme de graphe. Certaines de ces approches, à l'image de [Wang et al., 2012], mappent les données sur des portions du graphe de référence représentées par des concaténations des séquences contenues dans les noeuds de celui-ci. Certaines approches ([Huang et al., 2013 ; Dilthey et al., 2015]), ont été développées dans le cas particulier où le graphe de référence est *suffisamment linéaire* car représentant des ensembles de génomes de références, proches les uns des autres. Ces approches sont particulièrement efficaces mais ne concernent que cette catégorie de graphe qui ne représente pas correctement la variabilité existante dans les données génomiques.

Une librairie appelée GSSW a été développée `http://github.com/ekg/gssw`. Elle permet le mapping de séquences sur graphe, sans contrainte sur ce dernier. Cette librairie est actuellement en cours de développement. Elle est basée sur un mappeur sur séquences linéaires [Zhao et al., 2013]. À l'heure actuelle, il n'existe pas d'outil associé et la documentation de cette librairie indique que l'implémentation fournie permet de mapper des séquences de quelques dizaines de milliers de caractères sur des références du même ordre de grandeur. Ceci ne permet donc pas de passer à l'échelle des jeux de données actuels.

La notion de *read-threading* De nombreux outils d'assemblages basés sur le dBG impliquent une étape dite de *read-threading* aussi connue sous le nom du problème de super-chemin Eulérien [Pevzner et al., 2001]. Il s'agit de détecter et de ne prendre en compte dans le dBG construit à partir d'un ensemble de lectures que les chemins qui correspondent au moins à une lecture, comme défini par [Myers, 2005]. Ceci évite la construction d'assemblages faux dus à la répétition de séquences de taille $\geq k-1$ dans les lectures. Ce problème a été montré NP-difficile [Nagarajan and Pop, 2009] et ne connait pas, à notre connaissance, de solution générique. Les solutions habituellement implémentées, à l'image de l'implémentation de Spades [Bankevich et al., 2012], consistent à conserver au sein du dBG et lors de sa création, l'information des lectures ayant servi à sa construction. Ceci nécessite donc d'importantes ressources mémoires.

* *

*

Nous avons proposé deux outils de mapping de séquences sur des graphes dont les noeuds représentent/contiennent eux-mêmes des séquences. Le premier outil, nommé Blast-Graph et développé lors d'un stage de Guillaume Holley a servi à poser les bases du problème et à proposer une première solution [Holley and Peterlongo, 2012] quand l'alignement est calculé selon la *distance d'édition** et implique donc une phase de *programmation dynamique**. Comme son nom l'indique, Blast-Graph reprend les concepts fondamentaux de l'outil Blast [Altschul et al., 1990] basé sur le principe de l'heuristique *seed-and-extend*. Comme représenté Figure 4.1 page suivante, une subtilité

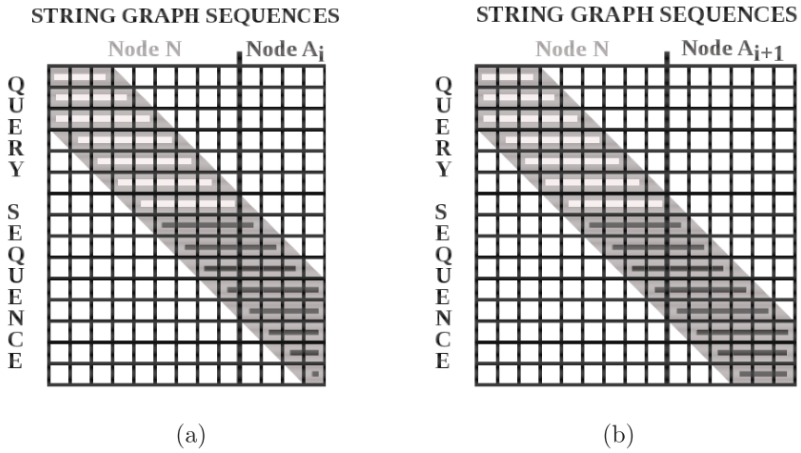


FIGURE 4.1 – Représentation graphique de la matrice de programmation dynamique calculant l'alignement d'une séquence requête (*query*) contre un couple de noeuds connectés $N.A_i$ dans la matrice **(a)** ou un couple de noeuds connectés $N.A_{i+1}$ dans la matrice **(b)**. L'algorithme implémenté dans Blast-Graph permet de factoriser les calculs lors de l'alignement de la requête au graphe contenant, entres autres, les couples de noeuds $N.A_i$ et $N.A_{i+1}$. Les lignes de la matrice spécifique au noeud N (fragments clairs) ne sont pas recalculés entre les deux matrices.

de cette approche consiste à factoriser les étapes de programmation dynamique pour éviter la redondance de calculs lors de l'exploration du graphe.

L'outil Blast-Graph souffre des limitations intrinsèques aux approches impliquant des phases de programmation dynamique. Si optimisée soit-elle, la programmation dynamique a dans ce cadre une complexité temporelle quadratique et lors du traitement de très grosses masses de données à l'image des données NGS, les temps de calcul deviennent rédhibitoires.

4.3 Solution pour le mapping intensif de séquences sur dBG

Nous proposons une étude poussée des possibilités et des limitations du mapping de séquences sur dBG. Cette étude est déposée sur le serveur arXiv Limasset and Peterlongo [2015].

4.3.1 Le mapping de séquences sur dBG, un problème NP-complet

Nous avons proposé une définition formelle du problème, appelé DBGRMP pour *de Bruijn graph Read Mapping Problem*. Dans cette formalisation, le graphe de référence est un graphe de de Bruijn ou un graphe de de Bruijn compacté CdBG. Nous rappelons (voir Section 1.4.2 page 11) qu'un CdBG contient les mêmes informations qu'un dBG.

Dans le problème DBGRMP, la distance autorisée entre une requête et un chemin du graphe est calculée selon une distance de Hamming, où seules les substitutions sont autorisées. De plus dans cette définition du problème le chemin mappé dans le graphe ne doit pas comporter de cycle.

Par réductions successives du problème du *chemin hamiltonien**, nous avons montré la *NP-*

*complétude** du problème DBGRMP. Le problème du chemin hamiltonien a été réduit au *problème du voyageur de commerce de taille fixée**, lui même réduit au problème de mapping sur graphe dont les arrêtes possèdent un caractère, lui même réduit au DBGRMP.

4.3.2 Une solution heuristique au problème

Le problème DBGRMP étant NP-complet, la recherche de toutes les positions de mapping possibles d'une séquence requête sur un CdBG peut demander l'exploration d'un nombre potentiellement insoluble de chemins. C'est pourquoi nous proposons une méthode heuristique pour répondre au problème DBGRMP en un temps limité et avec une consommation de ressources limitée.

FIGURE 4.2 – Image issue de [Limasset and Peterlongo, 2015]. Représentation d'une séquence requête ($CGTAC\ldots$ représentée en haut) mappée sur les noeuds d'un CdBG (avec $k = 6$) représentés en vert, lignes 2, 3 et 4. **Étape 1** : Détection dans la séquence requête des chevauchements entre les noeuds du CdBG (représentés en bleu, ligne 1). **Étape 2** : Les unitigs du CdBG qui mappent le début et la fin de la séquence requête sont détectés (unitigs représentés ligne 2). **Étape 3** : La partie restante de la requête (partie centrale) est mappée sur les unitigs, de la gauche vers la droite (dans cet exemple sur l'unitig représenté ligne 3, puis sur l'unitig représenté ligne 4).

Notre approche est basée sur l'heuristique *seed-and-extend*. Afin d'optimiser l'espace d'indexation ainsi que les temps d'exécution, nous avons décidé d'éviter d'indexer tous les mots d'une taille donnée. Nous avons choisi de n'indexer que les mots chevauchants les unitigs du CdBG, c'est à dire le préfixe de taille $k - 1$ et le suffixe de taille $k - 1$ de chaque unitig du CdBG de référence.

Le mapping d'une séquence de référence sur le graphe se fait en trois étapes, comme représenté Figure 4.2. Les détails de la méthode sont présentés dans [Limasset and Peterlongo, 2015]. Nous insisterons ici sur le fait que cette approche fait intervenir des méthodes heuristiques (utilisation de graines pour l'ancrage des alignements, *approches gloutonnes** pour le le mapping). Aussi les séquences de références peuvent être mal mappées ou pas mappées à tort.

* *

*

Il est particulièrement délicat de modéliser et de prévoir théoriquement l'impact des méthodes heuristiques appliquées. En effet, l'impact dépend de la complexité des données représentées par le graphe ainsi que de celle des données requêtes. Cette constatation a été à l'origine de l'implémentation d'un outil, BGREAT permettant d'estimer l'impact de telles méthodes heuristiques, dans des conditions réelles d'utilisation.

L'outil ainsi que des résultats sur données réelles sont présentés dans la section suivante.

4.3.3 BGREAT, un outil pour le mapping rapide de séquences sur graphe

Soucieux d'estimer les qualités et les défauts de la méthode algorithmique que nous avons proposée pour répondre au problème DBGRMP, nous avons jugé nécessaire de l'implémenter au sein d'un prototype que nous avons appelé BGREAT. L'implémentation a été faite en C++ par Antoine Limasset. Elle est disponible à l'adresse suivante : `github.com/Malfoy/BGREAT`.

Notons qu'il existe une littérature dense et de qualité concernant le mapping sur des séquences linéaires. Ainsi, nous avons fait le choix de nous préoccuper uniquement de séquences ne pouvant mapper que les zones branchantes du graphe. Autrement dit, le prototype BGREAT n'est pas capable de fournir de résultat pour une requête mappant dans son intégralité au sein d'un unique unitig du CdBG de référence. Ainsi dans les tests que nous avons effectués et que nous présentons dans la section suivante, cette étape est laissée à un logiciel de mapping classique, Bowtie 2 [Langmead and Salzberg, 2012a] dans notre cas.

Entrées & sorties de l'outil BGREAT L'outil BGREAT utilise deux fichiers.

L'un des deux fichiers contient le CdBG représenté sous forme d'une suite d'unitigs, eux mêmes stockés sous forme textuelle. En pratique, pour générer un tel fichier à partir de lectures brutes, nous utilisons l'outil DSK [Rizk et al., 2013] pour compter les k-mers et ne conserver que les k-mers dont la couverture est suffisante pour supprimer les erreurs de séquençage (voir Section 2.6 page 28). Les k-mers ainsi générés sont ensuite utilisés par l'outil BCALM [Chikhi et al., 2014] pour générer l'ensemble des unitigs utilisés par BGREAT.

Le second fichier utilisé par BGREAT est un fichier au format FASTA contenant l'ensemble des requêtes à mapper sur le graphe.

À l'heure actuelle, la sortie de BGREAT est composée des informations de mapping. Pour chaque requête, le chemin dans le graphe sur lequel la requête a mappé est indiqué sous la forme d'une suite d'identifiants des unitigs correspondants. Ceci n'est bien entendu valable que pour les requêtes ayant mappé le graphe avec succès.

4.3.4 Présentation de résultats préliminaires

Afin d'estimer les défauts et qualité de l'approche que nous proposons, nous avons effectué quelques tests sur des données issues d'*E.coli*, de *C.elegans* ou plus complexes comme l'humain [1]. Pour ce faire nous avons appliqué le pipeline représenté Figure 4.3 page suivante. Les résultats complets sont présentés dans la publication [Limasset and Peterlongo, 2015]. Nous nous contentons ici d'en tirer les conclusions principales.

Les principaux résultats sont synthétisés Table 4.1 page 73. Ils nous donnent les informations suivantes.

- Les résultats BGREAT des deux premières lignes de ce tableau nous indiquent que les choix algorithmiques effectués à propos des heuristiques sont pertinents : la méthode exacte permet de mapper seulement 0.03% de requêtes supplémentaires alors qu'elle prend plus de 40 fois plus de temps d'exécution. Il semble approprié de perdre 0.03% d'informations de mapping (sachant

1. Référence des jeux de lectures : *E.coli* : SRR959239 ; *C.elegans* : SRR065390 ; humain : SRR345593 et SRR345594. Ces données peuvent être téléchargées à partir du site *European Nucleotide Archive* `http://www.ebi.ac.uk/ena`

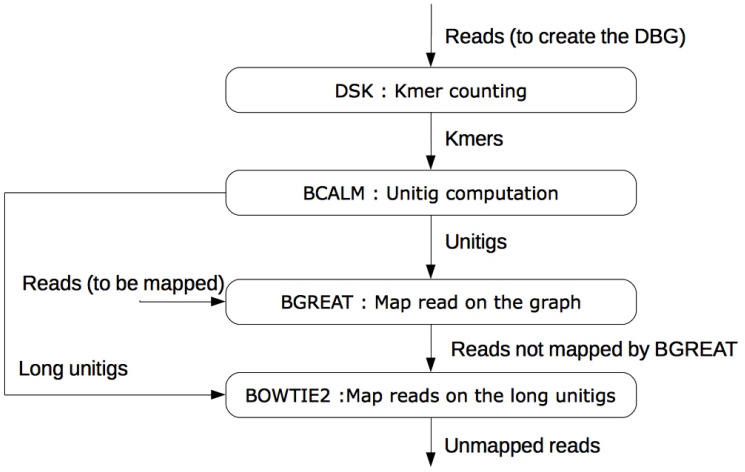


FIGURE 4.3 – Image issue de [Limasset and Peterlongo, 2015]. Pipeline utilisé lors des tests de l'outil BGREAT. Les lectures (*reads*) utilisées pour la création du CdBG peuvent être identiques ou différentes de celles utilisées comme requêtes. Seuls les unitigs plus longs que les lectures mappées sont utilisés par l'outil Bowtie 2.

que près de la totalité (97.17%) des requêtes sont mappées soit sur les parties branchantes avec BGREAT soit sur les unitigs avec Bowtie 2).

- L'idée de ne mapper les requêtes que sur les unitigs serait insuffisante. On constate que ≈13 à ≈20% des requêtes mappent sur des chemins composés de plusieurs noeuds. Sur des graphes plus complexes, ce pourcentage peut monter jusqu'à plus de 66% (voir [Limasset and Peterlongo, 2015])
- La plupart des tests effectués consistaient à considérer les lectures utilisées pour générer le CdBG également comme requêtes pour le mapping. Nous constatons (quatrième ligne du tableau), que les conclusions générales de nos expérimentations restent également valables si les requêtes sont différentes des lectures ayant servi à la construction du graphe.
- La dernière colonne est issue d'un pipeline qui n'est pas représenté Figure 4.3. Il consiste à assembler les lectures pour produire des contigs, puis à tester le mapping des requêtes sur ces assemblages. Nous avons testé deux assembleurs (Minia [Chikhi and Rizk, 2013] et Velvet [Zerbino and Birney, 2008]). Les résultats présentés sur la dernière colonne de la Table 4.1 page 73 ont été obtenus avec l'assembleur Minia. Cette dernière colonne est particulièrement intéressante. Elle permet de montrer que la méthode usuelle de mapping consistant à mapper des séquences sur des résultats d'assemblage n'est pas optimale. Ceci est assez peu marqué sur un génome simple tel qu'*E.coli* mais devient spécialement important lorsque le génome de référence est complexe. En effet, nous pouvons constater que seules 63.16% des requêtes mappent les données d'assemblage alors que 85.51% mappent le CdBG. Autrement dit, le mapping de plus de 20% des requêtes échouent. En outre, ces chiffres n'indiquent pas la quantité de mapping biaisés par les erreurs d'assemblage.

Les résultats préliminaires, en particuliers ceux comparant le taux de mapping sur graphe par rapport au taux de mapping sur données d'assemblage sont particulièrement engageants à poursuivre

nos travaux de recherche dans le mapping de séquences sur des références représentées sous forme de graphe.

Nous en sommes au tout début de l'aventure : de nombreuses applications peuvent être issues des résultats de ces informations de mapping sur graphe. Ces perspectives de travaux sont présentées dans le chapitre suivant alors que la section suivante présente les avancées techniques pouvant être appliquées à la méthode BGREAT.

4.4 Perspectives pour l'amélioration du mapping sur graphe

Comme nous l'avons indiqué en préambule de ce chapitre, ces travaux sont encore préliminaires. De nombreuses améliorations peuvent être apportées.

Amélioration et diffusion du prototype BGREAT L'outil BGREAT a été essentiellement implémenté afin d'offrir une preuve de concept et pour tester choix les algorithmiques possibles. Il nous a permis de constater que cette approche mérite d'être explorée. Ceci signifie également que BGREAT devra évoluer d'un prototype à un outil plus largement utilisable et plus performant. Outre les modifications techniques (structures d'indexation plus efficaces, sorties formatées, utilisation de graines plus appropriées [Vroland et al., 2014]) ceci impose une importante communication et une écoute poussée des utilisateurs pour offrir un outil adapté et diffusé. Cette part de travail n'est pas à négliger.

Cycles et indels La limitation principale de la méthode que nous avons proposée tient dans le fait que la distance autorisée lors du mapping est limitée à une distance de Hamming (substitutions uniquement) et que les chemins mappés ne peuvent pas comporter de cycles.

Comme nous avons pu le constater dans la section précédente, ces limitations n'ont finalement que peu d'impact sur les résultats obtenus en mappant des lectures courtes (100 nucléotides) sur des CdBG construits eux même à partir du même type de données. Cependant, elles peuvent devenir limitantes en fonction de l'application visée. Par exemple s'il s'agit de mapper des données d'espèces éloignées de celle ayant servi à la construction du CdBG, alors la distance de Hamming n'est plus suffisante et il devient nécessaire d'autoriser des insertions et délétions durant le mapping. De même, si le but est de mapper de longues séquences (type TGS par exemple), alors il devient plus probable que ces requêtes contiennent elles mêmes des séquences répétées qui mappent sur des zones cycliques du graphe. Il faut alors pouvoir utiliser ces chemins cycliques pour effectuer un mapping correct.

Nous allons donc chercher à moyen terme les méthodes les plus efficaces possibles pour autoriser des insertions et des délétions dans le mapping des requêtes et, également, adapter notre cadre théorique (formalisme, preuve de complexité) ainsi que notre solution algorithmique au mapping sur les parties cycliques du graphe de référence. Nous porterons bien entendu une attention particulière aux impacts en terme de besoins mémoires et aux temps d'exécutions. Nous étudierons par exemple des méthodes dites *Alignment-free* basées sur un nombre de sous-mots partagés comme première estimation de la similarité de séquences avant d'effectuer des alignements précis impliquant la programmation dynamique.

4.5 Présentation des publications associées

► Holley, G. and Peterlongo, P. (2012). BlastGraph : intensive approximate pattern matching in string graphs and de-Bruijn graphs. In *PSC 2012*, Prague, Czech Republic

Cette publication (fournie en annexe page 82) présente l'outil Blast-Graph.

► Limasset, A. and Peterlongo, P. (2015). Read Mapping on de Bruijn graph. *arXiv preprint arXiv :1505.04911*

Ce travail n'est pas encore publié, mais il est soumis en conférence et déposé sur le serveur ArXiv. La publication est fournie en annexe page 94. Elle présente l'étude théorique sur le mapping de séquences sur graphe et propose une solution implémentée (BGREAT) ainsi que des tests associés.

CdBG	Mêmes Requêtes	BGREAT Heur.	BGREAT Temps BGREAT	BGREAT # sur chemins	Bowtie # sur unitigs	Bowtie+BGREAT ∑# mappés	# sur contigs
E.coli	oui	oui	2m2	687,074 (13.40%)	4,296,710 (83.78%)	4,983,784 (97.17%)	4,933,520 (96.19%)
E.coli	oui	non	1h24	688,929 (13.43%)	4,295,814 (83.76%)	4,984,743 (97.19%)	4,933,520 (96.19%)
C.elegans	oui	oui	1h55	13,994,715 (20.70%)	48,442,146 (71.64%)	62,436,861 (92.34%)	54,383,764 (80.43%)
C.elegans	non$^+$	oui	12m25	3,523,416 (15.65%)	16,682,194 (74.11%)	20,205,610 (89.77)%	NA
Human	oui	oui	11h48*	1,004,182,363 (33.84%)	1,533,456,046 (51.67%)	2,537,638,409 (85.51%)	1,874,368,400 (63.16%)

$^+$données mappées : SRR1522085.
*résultats obtenus sur 12 coeurs.

TABLE 4.1 – Quelques résultats du pipeline (présenté Figure 4.3 page 70) utilisé pour tester les apports de BGREAT. La première colonne indique quel(s) jeu(x) de lectures ont servis à générer le CdBG. La seconde colonne indique si les données utilisées pour générer le CdBG de référence étaient les mêmes que celles mappées sur ce CdBG. La colonne "*Heur.*" indique si la version heuristique ou exacte du mapping a été utilisée. Le symbole '#' indique toujours un nombre de séquences mappées ; dans l'ordre des apparitions dans les colonnes : par BGREAT sur le CdBG, par Bowtie 2 sur le CdBG, la somme des deux, puis enfin par Bowtie 2 sur un assemblage des séquences ayant servi à générer le CdBG (non représenté dans le pipeline Figure 4.3 page 70). Les pourcentages indiquent le pourcentage de requêtes mappées par rapport au pourcentage total de requêtes. Dans le cas où les séquences mappées sont celles ayant servi à construire le graphe, le résultat souhaité est que 100% des séquences soient mappées.

Chapitre 5

Perspectives

Contents

5.1 Représentation des génomes **75**
5.2 Méthodologie pour la (multi-)métagénomique massive **77**
5.3 Perspectives personnelles **80**

Cette section présente les grands axes des travaux de recherche qui me motivent. Cette motivation nait de la rencontre de l'intérêt pour l'algorithmique, des défis techniques, des besoins identifiés des utilisateurs, et puis également de mes intérêts personnels et de la forte motivation pour les domaines en lien avec l'océanographie, comme c'est le cas de la métagénomique que l'on évoquera dans la Section 5.2 page 77.

5.1 Représentation des génomes

Nous l'avons évoqué dans le chapitre précédent. Depuis que nous avons accès à l'information génétique, la communauté scientifique met tout en oeuvre pour représenter les génomes sous forme d'une séquence (par chromosome) parfaitement linéaire du premier au dernier nucléotide. Pour l'oeil humain cette représentation est tout à fait logique et naturelle. Elle permet une bonne représentation "géographique" des différents éléments génomiques d'intérêt et présente de très nombreux avantages algorithmiques (loci uniques, une région donnée correspond à une unique séquence, ...).

Lors de l'assemblage de génomes, tout a donc été mis en oeuvre pour linéariser cette séquence de notre mieux. Les informations structurelles (variants de ploïdie, variants intra-individus, variants inter-individus des données *pool-seq**) sont méticuleusement gommées dans l'assemblage final lorsqu'ils ne découpent pas ce dernier en petites portions "*linéarisables*". L'idée clef est donc qu'une séquence linéaire est nécessairement une représentation biaisée et incomplète de la réalité.

Pour toutes ces raisons, un des points clefs que je souhaite développer dans les années à venir est de militer pour la représentation de référence sous forme de graphe plutôt que sous forme de séquences linéaires. Le graphe qui contient, avant sa linéarisation hasardeuse, toutes les informations de variants et de structures est une alternative puissante à la séquence.

Appelons ces génomes représentés sous forme de graphes des *graphical genomes*.

Nous avons déjà fait une proposition concrète d'utilisation de données génomiques représentées sous forme d'un graphe plutôt que sous forme de séquence. Tout le Chapitre 2 présente effectivement l'exploitation du dBG pour en extraire des variants biologiques d'intérêt. L'idée fondamentale est que ce graphe, hormis le fait que les erreurs de séquençage y sont filtrées, est une représentation parfaite de toutes les données séquencées. Les variants génomiques y sont donc tous présents. Comme nous l'avons évoqué Chapitre 2, une grande étape à franchir à moyen terme sera de proposer toute une série de résultats montrant les avantages (et les défauts) de cette approche par rapport à l'utilisation d'une séquence de référence, et ce, selon divers contextes et diverses qualités de données.

En outre, dans l'esprit de représenter les données sous forme de *graphical genomes*, voici quelques grands axes que j'aimerais explorer dans les années à venir.

5.1.1 Adaptation d'outils basés sur les séquences linéaires

Comme nous venons de l'évoquer, cela fait maintenant quelques dizaines d'années que l'on utilise des séquences linéaires pour représenter des génomes. Ainsi, les outils et les utilisateurs se sont adaptés à ce type de données.

Une étape majeure serait de proposer de nouveaux moyens de faire figurer efficacement (de manière pertinente et compréhensible) de l'information dans les *graphical genomes*. Cela implique notamment d'adapter les méthodes d'annotation et de visualisation d'annotations. Actuellement les *genome browsers* comme celui de l'UCSC `http://genome.ucsc.edu/cgi-bin/hgTracks` représentent à la fois une séquence linéaire et les annotations associées indiquant les variants, les gènes, et d'autres sources d'informations associées au génome. Il faudrait donc proposer une alternative convaincante à cette représentation. Ceci intégrerait un visualiseur de graphe sur lequel les variants seraient représentés par les branchements du graphe et seraient annotés à la manière des *genome browsers* actuels.

Les séquences génomiques sont souvent utilisées comparativement (détection de *groupes de synténies**, création de phylogénies). Ainsi, une condition nécessaire au déploiement des *graphical genomes* sera d'être en mesure de comparer efficacement ce type de génomes et de représenter les résultats de ce type de comparaison de manière compréhensible et exploitable.

À l'image de ce que nous avons présenté dans le chapitre précédent, il sera indispensable de proposer des outils de mapping adaptés aux *graphical genomes* alors utilisés comme références. Les résultats obtenus devront également être facilement visualisables et exploitables. C'est actuellement l'un des points clefs manquant aux résultats présentés dans le précédent chapitre.

Voyons comment les résultats du mapping de séquences sur des *graphical genomes* peuvent être exploités.

5.1.2 Assemblage

Les motivations qui nous ont conduit à proposer des approches de mapping sur *graphical genomes* étaient d'éviter la création de séquences linéaires. Ironiquement, les résultats de ce type de mapping

peuvent aider précisément à l'assemblage et donc à la création de séquences linéaires. Finalement, si l'exploitation de ce type de résultats permet d'obtenir de meilleurs assemblages, il serait malgré tout dommage de s'en passer.

Actuellement, l'une des sources à la fois de création d'assemblages chimériques (création de séquences différentes de celles ayant été séquencées) et de fragmentation des séquences assemblées est liée au découpage en k-mers. L'information de co-occurrences de k-mers dans une même lecture séquencée est perdue dans le dBG. Un $k-1$-mer présent en plusieurs copies sur le génome peut ainsi conduire à la création de séquences chimériques ou à des branchements dans le dBG, fragmentant les séquences assemblées.

L'une des idées que l'on soutient ici est d'utiliser des informations de mapping sur *graphical genomes* pour améliorer les assemblages. En supposant que les données mappées soient liées au génome séquencé (lectures utilisées pour faire le graphe, lectures issues de TGS, données d'assemblage plus sures telles que les BAC^* ou les fosmids), alors les *chemins* qu'elles empruntent dans le *graphical genomes* sont effectivement présents dans le génome à reconstruire. L'étape suivant la publication de BGREAT sera donc de proposer un assembleur utilisant efficacement le dBG dit *routé*, c'est-à-dire un dBG où les chemins validés par des données mappées seront indiqués et pourront être utilisés lors de l'assemblage. Cette approche pourrait permettre d'allier les qualités d'un assemblage par k-mer avec celles d'un assemblage de données plus longues.

5.1.3 Compression + correction

Les résultats du mapping sur *graphical genomes* peuvent être exploités également pour des aspects techniques de correction et/ou de compression des lectures séquencées. Une contribution importante dans ce domaine serait un gros point positif car le stockage des données de séquençage est un réel problème : la capacité de séquençage par euro dépensé double tous les cinq mois alors que la capacité de stockage par euro dépensé double elle tous les 14 mois. Ainsi il est clair qu'il est indispensable de proposer rapidement un moyen efficace de compresser cette information très redondante.

L'idée, déjà exploitée dans le logiciel Bloocoo Benoit et al. [2014], est de mapper les lectures sur des *graphical genomes* les plus similaires possible à ces lectures (voire sur celui obtenu à partir des lectures elles-mêmes). Pour chaque lecture mappée nous pouvons imaginer :

- Stocker la meilleure position de mapping ainsi que les différences entre la lecture et le graphe. Ceci permet de compresser les lectures.
- Si le graphe est lui-même créé avec les lectures mappées, il est possible de ne stocker que les différences entre la lecture et le graphe. Ceci permettrait alors de corriger les lectures (le graphe étant considéré comme parfait, exempt d'erreurs de séquençage).
- De nouveau, si le graphe a été créé avec les mêmes lectures que celles mappées, le simple stockage de la meilleure position de mapping pour chaque lecture permet à la fois la correction et la compression des données. Ceci reste à prouver et à diffuser.

5.2 Méthodologie pour la (multi-)métagénomique massive

La métagénomique offre un changement d'échelle par rapport à la génomique. L'étude de milieux complets et complexes ouvre la porte à de nouvelles découvertes, sur des milieux géographiquement limités (flore intestinale par exemple) jusqu'à l'échelle planétaire (projets d'échantillonnage des océans et de l'intégralité du *microbiome* du globe, voire séquençage extra-terrestre durant les explorations spatiale [Rezzonico, 2014]).

FIGURE 5.1 – Le séquenceur de poche MinION

Cette nouvelle discipline dispose d'un champs d'applications dont les avancées technologiques et algorithmiques futures feront certainement voler en éclat les limitations techniques actuelles. Les avancées technologiques concernent actuellement la quantité d'ADN nécessaire au séquençage, les biais d'amplification et de couverture des génomes en présence, le coût des matériaux et des machines, l'encombrement de ces machines, ou encore la longueur des séquences produites. Cependant, les avancées technologiques sont si rapides que ces caractéristiques évoluent rapidement. L'arrivée par exemple du séquenceur MinION (Figure 5.1) de la taille d'une grosse clef USB permet d'envisager le séquençage directement sur le terrain et non plus au laboratoire comme c'est le cas actuellement.

Finalement, de mon point de vue de *méthodologiste*, il me semble que l'un des défis les plus complexes à résoudre concernera plutôt l'analyse des données massives qui seront produites par les projets métagénomiques. Des protocoles précis et efficaces sont déjà bien établis lorsqu'il s'agit d'analyser des métagénomes dont le contenu est relativement bien connu. Cependant, lorsque l'on sort des sentiers défrichés (je n'ose pas parler de sentiers battus tant nous débutons dans ces domaines), les méthodes de gestion des données (métadonnées, stockage, correction, compression, mise à disposition) et d'analyse de celles-ci (estimation de biodiversité, compréhension des interactions intra ou inter-métagénomes, compréhension des processus biologiques, ...) sont encore balbutiantes.

Dans la lignée des travaux de métagénomique comparative que nous avons entrepris, nous

chercherons à poursuivre nos travaux dédiés à l'analyse comparative de métagénomes. Nous creuserons les questions telles que :

- Comment la présence de multi-métagénomes peut-elle être exploitée pour l'assemblage, l'estimation de la biodiversité et de la variabilité de la biodiversité ? Des idées dans ce sens sont en train de germer. L'assemblage métagénomique peut tirer parti de multiples métagénomes. Ceci peut être envisagé en extrayant les lectures qui sont spécifiques à certains métagénomes ; on se focalise alors sur quelques espèces ciblées *de-novo*. À l'inverse, nous pouvons tenter de n'assembler que les lectures partagées par tout un ensemble de métagénomes. Ces lectures appartiendraient alors aux espèces ubiquitaires, colonne vertébrale des océans par exemple dans le cas des données Tara Oceans.
  Nous pouvons imaginer également des processus d'assemblages itératifs. Les données spécifiques à chaque jeux de données seraient d'abord détectées et assemblées pour chaque métagénome. Dans un second temps, les données spécifiques de clusters de métagénomes seraient à leur tour détectées puis assemblées, et ainsi de suite jusqu'à la détection et l'assemblage des lectures associées aux espèces ubiquitaires, présentes dans tous les métagénomes.
- Comment la comparaison de métagénomes peut renseigner sur la biodiversité ou sur la "méta-biodiversité" (diversité de la biodiversité en fonction des conditions extérieurs géographiques ou physico-chimiques environnementales) ?
  Cette question fera intervenir des spécialistes océanographes et statisticiens qui apporteront un regard spécialisé sur ces questions. Nous proposerons quant à nous les méthodes pour analyser les données et offrir les informations nécessaires pour répondre à ces questions.
- Comment intégrer la présence de données hétérogènes, elles aussi multiples ? Dans les grands projets métagénomiques, chaque échantillon va de pair avec d'autres informations hétérogènes. Dans l'exemple du projet Tara Oceans, il s'agit d'informations géographique, de profondeur sous la surface de l'eau, de taille de filtre, de conditions temporelles, et physico-chimiques (pH, salinité, température, ...). Nous aimerions pouvoir faire intervenir ces différentes sources d'informations directement au sein d'algorithmes comparatifs pour répondre, par exemple, aux questions suivantes : quelles lectures ne sont partagées que par des milieux froids ? acides ?, etc.

En marge de ces questions applicatives, d'autres questions plus techniques se posent. Nous chercherons ainsi à proposer de nouvelles méthodes visant à mieux stocker les métagénomes en factorisant efficacement les redondances intra et inter-métagénomes. Une telle factorisation sera(it) en outre une source importante d'information pouvant être exploitée pour répondre par exemple aux questions suivantes.

- Comparer plus efficacement les métagénomes. Connaitre les données partagées par n métagénomes parmi m est une source d'information utilisable pour comparer les métagénomes. C'est finalement ce que nous avons proposé au travers des outils Commet et Simka, et ce essentiellement à l'échelle des k-mers partagés. Nous devrions aller plus loin sur les avancées algorithmiques de ces méthodes pour organiser (et donc indexer) les données partagées par les métagénomes. Ceci permettrait de changer d'échelle, par exemple pour proposer des résultats comparatifs, non plus à l'échelle du k-mer, mais à celle des lectures, des gènes, ou encore des espèces.
- Proposer une méthode de requêtage sur multi-métagénomes. L'idée est simple et consiste à

offrir une possibilité efficace, c'est à dire dans un temps rapide et accessible à tous, de répondre aux questions suivantes. **Question 1** : étant donnée une requête, dans quel(s) métagénome(s) (parmi un ensemble potentiellement grand de métagénomes) cette requête est-elle présente ? **Question 2** : de quel métagénome cette requête est-elle la plus proche (ce qui impose de proposer un système de score d'alignement) ?
Il semble impensable d'indexer et/ou de parcourir pour chaque requête l'intégralité des données métagénomiques. Dans cette optique, la factorisation des données présentée précédemment jouera un rôle majeur. L'idée envisageable serait d'indexer et/ou de parcourir pour chaque requête les données de multiple métagénomes stockées et indexées de manière non redondante. Nous pourrons alors, à l'image de méthodes telles que Kraken [Wood and Salzberg, 2014], proposer des indexes hiérarchisés permettant d'indiquer dans quel(s) métagénome(s) une requête peut avoir une similarité de séquence. Cela peut représenter une réponse en soi à la question 1, et permet de ne requêter que les bons métagénomes pour répondre à la question 2.

5.3 Perspectives personnelles

Continuer à aimer mon travail. Continuer à profiter au maximum de ce(s) équipe(s) formidable(s) et de celles et ceux qui les composent. Continuer à rester curieux, à l'écoute de tous, tout en rejetant la concurrence, pour lui préférer l'entraide, ce facteur d'évolution [Kropotkin, 1906] mais aussi d'efficacité et de plaisir, tout simplement.

Enfin, rester conscient de la chance de pouvoir apporter une brique, si infime soit-elle, à cette formidable aventure qu'est la découverte du fonctionnement de la vie.

Glossaire

ADN
ADN signifie acide désoxyribonucléique, et constitue la molécule support de l'information génétique héréditaire.

Algorithme génétique
Algorithme d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle.

Alpha-diversité
Mesure de la biodiversité de tout ou partie d'un écosystème. La beta-diversité, quant à elle, mesure la diversité des espèces entre écosystèmes

Amorce
Courte séquence d'ARN ou d'ADN qui s'hybride à sa séquence complémentaire sur un génome. Les amorces ont divers applications : détection de la présence de la séquence dans un génome, définition de séquences à amplifier, etc...

ARN
Molécule qui transporte l'information contenue dans le patrimoine génétique (ADN) jusqu'aux ribosomes qui sont chargés de la "traduire" en protéines ayant des fonctions précises.

Assemblage
Dans le cadre des données NGS ou TGS, l'assemblage consiste à déterminer la séquence génomique ayant généré un ensemble de lectures.

BAC
Les chromosomes artificiels sont des structures qui copient les chromosomes de la cellule hôte eucaryote et se répartissent régulièrement lors des divisions, comme les chromosomes naturels. Ces chromosomes artificiels ont une capacité très augmentée d'accepter de l'ADN étranger. On en a construit pour les bactéries : Bacterial Artificial Chromosome (BAC).

Beta-diversité
cf Alpha-diversité.

Bootstrap
Méthode statistique basée sur l'observation de résultats obtenus sur des ré-échantillonnages successifs

Cartographie génétique
Construction d'une carte soit localisée autour d'un gène, soit à base large portant sur le

génome entier. Plus généralement, c'est la détermination de la position d'un locus (gène ou marqueur génétique) sur un chromosome.

Chemin hamiltonien
Chemin qui passe une fois et une seule par chaque sommet d'un graphe.

Composante bi-connexe
Dans un graphe non orienté, une composante bi-connexe est un sous ensemble maximal de sommet tels qu'il existe au moins deux chemins distincts reliant tout couple de sommets de cet ensemble.

Composante connexe
Dans un graphe non orienté, une composante connexe est un sous ensemble maximal de sommet tels qu'il existe un chemin reliant tout couple de sommets de cet ensemble.

Contigs
Séquence continue et ordonnée générée par l'assemblage de lectures NGS ou TGS. La génération de contig est issue d'un processus heuristique. Les contigs peuvent donc contenir des erreurs.

Correction
Dans le cadre des données NGS ou TGS, la correction consiste à détecter et à corriger les erreurs de séquençage.

dBG
cf "*graphe de De Bruijn*"

Distance d'édition
Distance minimale entre deux séquences textuelles en terme de nombre de substitutions, d'insertions et de délétions. Notons que des *poids* différents peuvent être appliqués à ces différentes opérations.

Electrophorèse
Technique de laboratoire permettant la séparation des protéines ou des acides nucléiques grâce à leur différence de masse en présence d'un champ électrique.

Épissage
Processus par lequel les ARN transcrits à partir de l'ADN génomique peuvent subir des étapes de coupure et ligature qui conduisent à l'élimination de certaines régions dans l'ARN final.

Eucaryote
Ensemble des organismes unicellulaires ou multicellulaires dont les cellules sont dites 'eucaryotes". Elles possèdent un noyau et des organites (réticulum endoplasmique, appareil de Golgi, plastes divers, mitochondries, etc.) délimités par des membranes.

Les eucaryotes se distinguent des *procaryotes* (comme les bactéries) qui sont pour leur part dépourvus de ces structures.

Graphe de De Bruijn
Dans le cadre de l'assemblage des données NGS ou TGS, le graphe de De Bruijn désigne un graphe dont les noeuds contiennent les k-mers issus des lectures à assembler et où une arrête existe entre deux noeuds dont le chevauchement des k-mers est de taille $k-1$.

Gène
Élément génétique correspondant à un segment d'ADN ou d'ARN (virus), situé à un endroit

bien précis (locus) sur un chromosome. Chaque région de l'ADN qui produit une molécule d'ARN fonctionnelle est un gène.

Génome
Ensemble du matériel génétique d'un organisme.

Génotype
Constitution génétique d'un individu.

Génotyper
Définir le génotype.

Hétérotrophe
Qui utilise pour se nourrir les matières organiques constituant ou ayant constitué d'autres organismes

Heuristique
Méthode de calcul qui fournit rapidement (en temps polynomial) une solution réalisable, pas nécessairement optimale.

k-mer
Mot de taille k

KPCA
cf PCA

Lectures (ou *reads*)
Données fournies par les séquenceurs NGS ou TGS. Les lectures sont de courtes séquences issues des génomes ou transcriptomes séquencés.

Mapping
Dans le cadre des données NGS ou TGS, le mapping désigne la détermination de la position de lecture(s) sur un génome de référence et sur leur alignement permettant de détecter les éventuelles différences entre chaque lecture et le génome dont elle est issue.

Marqueur
Un marqueur moléculaire est une séquence permettant d'identifier une région d'un génome (cf Amorce) ou une espèce. Les marqueurs appelés 16S ou 18S sont couramment utilisés pour identifier et distinguer les espèces bactériennes.

Métabolisme
Décrit l'ensemble des réactions chimiques qui se déroulent au sein d'un individu.

Méthode gloutonne / algorithme glouton
algorithme qui suit le principe de faire, étape par étape, un choix optimum local, dans l'espoir d'obtenir un résultat optimum global

NP-complétude ou problème NP-complet
Problème NP complet : problème de décision vérifiant certaines conditions de complexité.

Nucléotide
Molécule composée d'un sucre, d'un phosphate et d'une molécule azotée. Ce sont les unités de base de l'ADN et de l'ARN.

PCA
(“*Principal component analysis*”). Analyse en composantes principales : méthode d’analyse des données qui consiste à transformer des variables corrélées en nouvelles variables décorrélées les unes des autres. Ces nouvelles variables sont nommées ”composantes principales”. La KPCA est une extension de ce type de méthodes : schématiquement, les dépendances entre variables peuvent ne pas être linéaires, mais peuvent dépendre de “*noyaux*” (les “*kernels*”).

Phénotype
Caractéristiques biochimiques, physiologiques et morphologiques d’un individu.

Phylogénie
Étude des relations de parentés entre différents êtres vivants en vue de comprendre l’évolution des organismes vivants.

Pool de données / pool-seq
Des données sont dites “*poolées*” lorsque qu’elles sont mise en commun. Dans le cas des données NGS, il peut s’agir de plusieurs individus séquencés simultanément, ou de plusieurs jeux de données NGS analysés conjointement, sans distinction de leur origine.

Précision
Métrique déterminant pour une méthode de prédiction, le nombre d’éléments correctement prédits par rapport au nombre total d’éléments prédits.

Problème du voyageur de commerce de taille fixée
Étant donné un graphe dont les arêtes possèdent des poids $\in \mathbb{N}^+$, existe-t’il un chemin composé d’un nombre fixé de noeuds dont le poids cumulé des arêtes est inférieur à un seuil donné.

Programmation dynamique
Méthode algorithmique permettant de calculer une solution optimale d’un problème en combinant ses solutions sous-optimales. Le calcul de distane d’édition nécessite l’utilisation de la programmation dynamique.

Protéine
Molécule composée d’une chaîne d’acides aminés.

Recall
Métrique déterminant la capacité d’une méthode de prédiction à détecter effectivement les éléments recherchés.

Seed-and-extend
Principe heuristique classiquement utilisé pour aligner une séquence requête à une référence. Le calcul d’alignement par programmation dynamique (*extend*) de la requête sur une position de la séquence de référence n’est effectué que si à cette position, la requête et la référence partagent exactement des mots communs (les graines/*seeds*).

Séquençage
Le séquençage détermine la séquence textuelle des génomes.

Synténie
Présence simultanée sur le même chromosome de deux ou plusieurs loci, indépendamment de leur liaison génétique.

Transcription
Processus permettant la copie de l'ADN en ARN, ou de l"ARN en ARN messager dans le cas de certains virus

Transcriptome
Ensemble des ARN issus de la transcription du génome.

Unitig
Séquence continue et ordonnée générée par l'assemblage de lectures NGS ou TGS. Les unitigs sont issus de chemins simples et maximaux du graphe de De Bruijn. Si les k-mers représentent effectivement les données à assembler, alors les unitigs sont exempts d'erreurs d'assemblage.

Unité taxonomique opérationnelle (OTU)
Une OTU est un regroupement d'individus d'une même espèce dont les séquences d'ARNr 16S présentent une similitude de plus de 97,5%. Cette standardisation internationale permet à des chercheurs d'une même discipline de comparer leurs résultats, mais elle n'a pas forcément une signification biologique.

Zone photique
zone verticale exposée à une lumière suffisante pour que la photosynthèse se produise

Bibliographie

Alberts, B. M., Bray, D., Johnson, A., and Perelman, S. (1999). *L'essentiel de la biologie cellulaire : introduction à la biologie moléculaire de la cellule.* Médecine-sciences Flammarion.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3) :403–410.

Altshuler, D. M., Durbin, R. M., Abecasis, G. R., Bentley, D. R., Chakravarti, A., Clark, A. G., Donnelly, P., Eichler, E. E., Flicek, P., Gabriel, S. B., Gibbs, R. A., Green, E. D., Hurles, M. E., Knoppers, B. M., Korbel, J. O., Lander, E. S., Lee, C., Lehrach, H., Mardis, E. R., Marth, G. T., Mcvean, G. A., Nickerson, D. A., Schmidt, J. P., Sherry, S. T., Wang, J., Wilson, R. K., Dinh, H., Kovar, C., Lee, S., Lewis, L., Muzny, D., Reid, J., Wang, M., Fang, X. D., Guo, X. S., Jian, M., Jiang, H., Jin, X., Li, G. Q., Li, J. X., Li, Y. R., Li, Z., Liu, X., Lu, Y., Ma, X. D., Su, Z., Tai, S. S., Tang, M. F., Wang, B., Wang, G. B., Wu, H. L., Wu, R. H., Yin, Y., Zhang, W. W., Zhao, J., Zhao, M. R., Zheng, X. L., Zhou, Y., Gupta, N., Clarke, L., Leinonen, R., Smith, R. E., Zheng-Bradley, X., Grocock, R., Humphray, S., James, T., Kingsbury, Z., Sudbrak, R., Albrecht, M. W., Amstislavskiy, V. S., Borodina, T. A., Lienhard, M., Mertes, F., Sultan, M., Timmermann, B., Yaspo, M. L., Fulton, L., Fulton, R., Weinstock, G. M., Balasubramaniam, S., Burton, J., Danecek, P., Keane, T. M., Kolb-Kokocinski, A., McCarthy, S., Stalker, J., Quail, M., Davies, C. J., Gollub, J., Webster, T., Wong, B., Zhan, Y. P., Auton, A., Yu, F., Bainbridge, M., Challis, D., Evani, U. S., Lu, J., Nagaswamy, U., Sabo, A., Wang, Y., Yu, J., Coin, L. J. M., Fang, L., Li, Q. B., Li, Z. Y., Lin, H. X., Liu, B. H., Luo, R. B., Qin, N., Shao, H. J., Wang, B. Q., Xie, Y. L., Ye, C., Yu, C., Zhang, F., Zheng, H. C., Zhu, H. M., Garrison, E. P., Kural, D., Lee, W. P., Leong, W. F., Ward, A. N., Wu, J. T., Zhang, M. Y., Griffin, L., Hsieh, C. H., Mills, R. E., Shi, X. H., von Grotthuss, M., Zhang, C. S., Daly, M. J., DePristo, M. A., Banks, E., Bhatia, G., Carneiro, M. O., del Angel, G., Genovese, G., Handsaker, R. E., Hartl, C., McCarroll, S. A., Nemesh, J. C., Poplin, R. E., Schaffner, S. F., Shakir, K., Yoon, S. C., Lihm, J., Makarov, V., Jin, H. J., Kim, W., Kim, K. C., Rausch, T., Beal, K., Cunningham, F., Herrero, J., McLaren, W. M., Ritchie, G. R. S., Gottipati, S., Keinan, A., Rodriguez-Flores, J. L., Sabeti, P. C., Grossman, S. R., Tabrizi, S., Tariyal, R., Cooper, D. N., Ball, E. V., Stenson, P. D., Barnes, B., Bauer, M., Cheetham, R. K., Cox, T., Eberle, M., Kahn, S., Murray, L., Peden, J., Shaw, R., Ye, K., Batzer, M. A., Konkel, M. K., Walker, J. A., MacArthur, D. G., Lek, M., Herwig, R., Shriver, M. D., Bustamante, C. D., Byrnes, J. K., De la Vega, F. M., Gravel, S., Kenny, E. E., Kidd, J. M., Lacroute, P., Maples, B. K., Moreno-Estrada, A., Zakharia, F., Halperin, E., Baran, Y., Craig, D. W., Christoforides, A., Homer, N., Izatt, T., Kurdoglu, A. A., Sinari, S. A., Squire, K., Xiao,

C. L., Sebat, J., Bafna, V., Burchard, E. G., Hernandez, R. D., Gignoux, C. R., Haussler, D., Katzman, S. J., Kent, W. J., Howie, B., Ruiz-Linares, A., and Dermitzakis, E. T. (2012). An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491 :56–65.

Amann, R. I., Ludwig, W., and Schleifer, K. H. (1995). Phylogenetic identification and in situ detection of individual microbial cells without cultivation. *Microbiological reviews*, 59(1) :143–169.

Anisimova, M., Pecerska, J., and Schaper, E. (2015). Statistical Approaches to Detecting and Analyzing Tandem Repeats in Genomic Sequences. *Frontiers in Bioengineering and Biotechnology*, 3(March) :1–6.

Antoniou, P., Crochemore, M., Iliopoulos, C., and Peterlongo, P. (2007). Application of suffix trees for the acquisition of common motifs with gaps in a set of strings. In *International Conference on Language and Automata Theory and Applications*, Tarragona, Spain.

Antoniou, P., Holub, J., Iliopoulos, C., Melichar, B., and Peterlongo, P. (2006). Finding Common Motifs with Gaps Using Finite Automata. In *Implementation and Application of Automata*, pages 69–77, Taipei, Taiwan.

Ball, E. V., Stenson, P. D., Abeysinghe, S. S., Krawczak, M., Cooper, D. N., and Chuzhanova, N. A. (2005). Microdeletions and microinsertions causing human genetic disease : common mechanisms of mutagenesis and the role of local dna sequence complexity. *Human mutation*, 26(3) :205–213.

Bankevich, A., Nurk, S., Antipov, D., Gurevich, A. A., Dvorkin, M., Kulikov, A. S., Lesin, V. M., Nikolenko, S. I., Pham, S., Prjibelski, A. D., Pyshkin, A. V., Sirotkin, A. V., Vyahhi, N., Tesler, G., Alekseyev, M. A., and Pevzner, P. A. (2012). SPAdes : A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. *Journal of Computational Biology*, 19(5) :455–477.

Barrett, B. S., Smith, D. S., Li, S. X., Guo, K., Hasenkrug, K. J., and Santiago, M. L. (2012). A single nucleotide polymorphism in ¡italic¿tetherin¡/italic¿ promotes retrovirus restriction ¡italic¿in vivo¡/italic¿. *PLoS Pathog*, 8(3) :e1002596.

Batzoglou, S., Jaffe, D. B., Stanley, K., Butler, J., Gnerre, S., Mauceli, E., Berger, B., Mesirov, J. P., and Lander, E. S. (2002). ARACHNE : A whole-genome shotgun assembler. *Genome Research*, 12(1) :177–189.

Benoit, G., Lavenier, D., Lemaitre, C., and Rizk, G. (2014). Bloocoo, a memory efficient read corrector. In *European Conference on Computational Biology (ECCB)*.

Benoit, G., Peterlongo, P., Lavenier, D., and Lemaitre, C. (2015). Simka : fast kmer-based method for estimating the similarity between numerous metagenomic datasets. JOBIM 2015.

Bianchi, J. S. and Kersten, R. d. A. (2014). Edge effect on vascular epiphytes in a subtropical Atlantic Forest. *Acta Botanica Brasilica*, 28(1) :120–126.

Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7) :422–426.

Boisvert, S., Raymond, F., Godzaridis, E., Laviolette, F., and Corbeil, J. (2012). Ray Meta : scalable de novo metagenome assembly and profiling. *Genome Biology*, 13(12) :R122.

Broder, A. and Mitzenmacher, M. (2004). Network applications of bloom filters : A survey. *Internet Mathematics*, 1(4) :485–509.

Cardoso, A. M., Cavalcante, J. J. V., Cantão, M. E., Thompson, C. E., Flatschart, R. B., Glogauer, A., Scapin, S. M. N., Sade, Y. B., Beltrão, P. J. M. S. I., Gerber, A. L., Martins, O. B., Garcia, E. S., de Souza, W., and Vasconcelos, A. T. R. (2012). Metagenomic Analysis of the Microbiota from the Crop of an Invasive Snail Reveals a Rich Reservoir of Novel Genes. *PLoS ONE*, 7(11).

Chikhi, R., Limasset, A., et al. (2014). On the representation of de bruijn graphs. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8394 LNBI, pages 35–55.

Chikhi, R. and Medvedev, P. (2014). Informed and automated k-mer size selection for genome assembly. *Bioinformatics*, 30(1) :31–37.

Chikhi, R. and Rizk, G. (2013). Space-efficient and exact de Bruijn graph representation based on a Bloom filter. *Algorithms Mol Biol*, 8(1) :22.

Collard, B. C. Y., Jahufer, M. Z. Z., Brouwer, J. B., and Pang, E. C. K. (2005). An introduction to markers, quantitative trait loci (QTL) mapping and marker-assisted selection for crop improvement : The basic concepts.

de la Bastide, M. and McCombie, W. R. (2007). Assembling genomic DNA sequences with PHRAP. *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]*, Chapter 11 :Unit11.4.

de Vargas, C., Audic, S., Henry, N., Decelle, J., Mahe, F., Logares, R., Lara, E., Berney, C., Le Bescot, N., Probert, I., Carmichael, M., Poulain, J., Romac, S., Colin, S., Aury, J.-M., Bittner, L., Chaffron, S., Dunthorn, M., Engelen, S., Flegontova, O., Guidi, L., Horak, A., Jaillon, O., Lima-Mendez, G., Luke, J., Malviya, S., Morard, R., Mulot, M., Scalco, E., Siano, R., Vincent, F., Zingone, A., Dimier, C., Picheral, M., Searson, S., Kandels-Lewis, S., Acinas, S. G., Bork, P., Bowler, C., Gorsky, G., Grimsley, N., Hingamp, P., Iudicone, D., Not, F., Ogata, H., Pesant, S., Raes, J., Sieracki, M. E., Speich, S., Stemmann, L., Sunagawa, S., Weissenbach, J., Wincker, P., Karsenti, E., Boss, E., Follows, M., Karp-Boss, L., Krzic, U., Reynaud, E. G., Sardet, C., Sullivan, M. B., and Velayoudon, D. (2015). Eukaryotic plankton diversity in the sunlit ocean. *Science*, 348(6237) :1261605–1261605.

Delmont, T. O., Malandain, C., Prestat, E., Larose, C., Monier, J.-M., Simonet, P., and Vogel, T. M. (2011). Metagenomic mining for microbiologists. *The ISME Journal*, 5(12) :1837–1843.

Deorowicz, S., Kokot, M., and Grabowski, S. (2014). KMC 2 : Fast and resource-frugal k-mer counting. *arXiv*, 1 :1–21.

Dilthey, A., Cox, C., Iqbal, Z., Nelson, M. R., and McVean, G. (2015). Improved genome inference in the MHC using a population reference graph. *Nature Genetics*, 47(6) :682–688.

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., and Gingeras, T. R. (2013). STAR : Ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1) :15–21.

Drezen, E., Rizk, G., Chikhi, R., Deltel, C., Lemaitre, C., Peterlongo, P., and Lavenier, D. (2014). GATB : Genome Assembly & Analysis Tool Box. *Bioinformatics (Oxford, England)*, pages 1–3.

Dutilh, B. E., Schmieder, R., Nulton, J., Felts, B., Salamon, P., Edwards, R. a., and Mokili, J. L. (2012). Reference-independent comparative metagenomics using cross-assembly : crAss. *Bioinformatics*, 28(24) :3225–3231.

Federico, M., Peterlongo, P., and Pisanti, N. (2010). An optimized filter for finding multiple repeats in DNA sequences. In *2010 ACS/IEEE International Conference on Computer Systems and Applications*, HAMMAMET, Tunisia.

Federico, M., Peterlongo, P., Pisanti, N., and Sagot, M.-F. (2011). Finding Long and Multiple Repeats with Edit Distance. In *The Prague Stringology Conference 2011*, Prague, Czech Republic.

Federico, M., Peterlongo, P., Pisanti, N., and Sagot, M.-F. (2014). Rime : Repeat identification. *Discrete Applied Mathematics*, 163(3) :275–286.

Fimereli, D., Detours, V., and Konopka, T. (2013). TriageTools : tools for partitioning and prioritizing analysis of high-throughput sequencing data. *Nucleic Acids Research*, 41(7) :e86–e86.

Gallé, M., Peterlongo, P., and Coste, F. (2009). In-place update of suffix array while recoding words. *International Journal of Foundations of Computer Science*, 20(6) :1025–1045.

Gemayel, R., Vinces, M. D., Legendre, M., and Verstrepen, K. J. (2010). Variable tandem repeats accelerate evolution of coding and regulatory sequences. *Annual review of genetics*, 44 :445–477.

Gilbert, J. A., Jansson, J. K., and Knight, R. (2014). The Earth Microbiome project : successes and aspirations. *BMC Biology*, 12(1) :69.

Gonzalez, K. D., Hill, K. A., Li, K., Li, W., Scaringe, W. A., Wang, J. C., Gu, D., and Sommer, S. S. (2007). Somatic microindels : Analysis in mouse soma and comparison with the human germline. *Human Mutation*, 28(1) :69–80.

Hoffmann, A. A., Sgrò, C. M., and Weeks, A. R. (2004). Chromosomal inversion polymorphisms and adaptation.

Holley, G. and Peterlongo, P. (2012). BlastGraph : intensive approximate pattern matching in string graphs and de-Bruijn graphs. In *PSC 2012*, Prague, Czech Republic.

Huang, L., Popic, V., and Batzoglou, S. (2013). Short read alignment with populations of genomes. *Bioinformatics*, 29(13) :i361–i370.

Huang, X., Wang, J., Aluru, S., Yang, S. P., and Hillier, L. (2003). PCAP : A whole-genome assembly program. *Genome Research*, 13(9) :2164–2170.

Iliopoulos, C., Mchugh, J., Peterlongo, P., Pisanti, N., Rytter, W., and Sagot, M.-F. (2005). A first approach to finding common motifs with gaps,. *International Journal of Foundations of Computer Science*, 16(6) :1145–1155.

Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., and McVean, G. (2012). De novo assembly and genotyping of variants using colored de bruijn graphs. *Nature genetics*, 44(2) :226–232.

Ishii, S., Yamamoto, M., Kikuchi, M., Oshima, K., Hattori, M., Otsuka, S., and Senoo, K. (2009). Microbial populations responsive to denitrification-inducing conditions in rice paddy soil, as revealed by comparative 16S rRNA gene analysis. *Applied and Environmental Microbiology*, 75(22) :7070–7078.

Jacques Nicolas, Sébastien Tempel, Pierre Peterlongo (In press). Finding and characterizing repeats in plant genomes. In Dave Edwards, editor, *Plant Bioinformatics : Methods and Protocols, Second Edition*. Humana Press.

Jaenicke, S., Ander, C., Bekel, T., Bisdorf, R., Dröge, M., Gartemann, K.-H., Jünemann, S., Kaiser, O., Krause, L., Tille, F., Zakrzewski, M., Pühler, A., Schlüter, A., and Goesmann, A. (2011). Comparative and Joint Analysis of Two Metagenomic Datasets from a Biogas Fermenter Obtained by 454-Pyrosequencing. *PLoS ONE*, 6(1) :e14519.

Jurka, J. (1998). Repeats in genomic DNA : Mining and meaning.

Karsenti, E. (2012). Towards an 'Oceans Systems Biology'. *Molecular Systems Biology*, 8.

Karsenti, E., Acinas, S. G., Bork, P., Bowler, C., de Vargas, C., Raes, J., Sullivan, M., Arendt, D., Benzoni, F., Claverie, J. M., Follows, M., Gorsky, G., Hingamp, P., Iudicone, D., Jaillon, O., Kandels-Lewis, S., Krzic, U., Not, F., Ogata, H., Pesant, S., Reynaud, E. G., Sardet, C., Sieracki, M. E., Speich, S., Velayoudon, D., Weissenbach, J., and Wincker, P. (2011). A holistic approach to marine Eco-systems biology. *PLoS Biology*, 9(10).

Kent, W. J. (2002). BLAT—The BLAST-Like Alignment Tool. *Genome Research*, 12(4) :656–664.

Kim, J. J., Vaziri, S. A., Rini, B. I., Elson, P., Garcia, J. A., Wirka, R., Dreicer, R., Ganapathi, M. K., and Ganapathi, R. (2012). Association of vegf and vegfr2 single nucleotide polymorphisms with hypertension and clinical outcome in metastatic clear cell renal cell carcinoma patients treated with sunitinib. *Cancer*, 118(7) :1946–1954.

Kropotkin, P. A. (1906). *L'entr'aide : un facteur de l'évolution*. Hachette.

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W., Funke, R., Gage, D., Harris, K., Heaford, A., Howland, J., Kann, L., Lehoczky, J., LeVine, R., McEwan, P., McKernan, K., Meldrim, J., Mesirov, J. P., Miranda, C., Morris, W., Naylor, J., Raymond, C., Rosetti, M., Santos, R., Sheridan, A., Sougnez, C., Stange-Thomann, N., Stojanovic, N., Subramanian, A., Wyman, D., Rogers, J., Sulston, J., Ainscough, R., Beck, S., Bentley, D., Burton, J., Clee, C., Carter, N., Coulson, A., Deadman, R., Deloukas, P., Dunham, A., Dunham, I., Durbin, R., French, L., Grafham, D., Gregory, S., Hubbard,

T., Humphray, S., Hunt, A., Jones, M., Lloyd, C., McMurray, A., Matthews, L., Mercer, S., Milne, S., Mullikin, J. C., Mungall, A., Plumb, R., Ross, M., Shownkeen, R., Sims, S., Waterston, R. H., Wilson, R. K., Hillier, L. W., McPherson, J. D., Marra, M. A., Mardis, E. R., Fulton, L. A., Chinwalla, A. T., Pepin, K. H., Gish, W. R., Chissoe, S. L., Wendl, M. C., Delehaunty, K. D., Miner, T. L., Delehaunty, A., Kramer, J. B., Cook, L. L., Fulton, R. S., Johnson, D. L., Minx, P. J., Clifton, S. W., Hawkins, T., Branscomb, E., Predki, P., Richardson, P., Wenning, S., Slezak, T., Doggett, N., Cheng, J. F., Olsen, A., Lucas, S., Elkin, C., Uberbacher, E., Frazier, M., Gibbs, R. A., Muzny, D. M., Scherer, S. E., Bouck, J. B., Sodergren, E. J., Worley, K. C., Rives, C. M., Gorrell, J. H., Metzker, M. L., Naylor, S. L., Kucherlapati, R. S., Nelson, D. L., Weinstock, G. M., Sakaki, Y., Fujiyama, A., Hattori, M., Yada, T., Toyoda, A., Itoh, T., Kawagoe, C., Watanabe, H., Totoki, Y., Taylor, T., Weissenbach, J., Heilig, R., Saurin, W., Artiguenave, F., Brottier, P., Bruls, T., Pelletier, E., Robert, C., Wincker, P., Smith, D. R., Doucette-Stamm, L., Rubenfield, M., Weinstock, K., Lee, H. M., Dubois, J., Rosenthal, A., Platzer, M., Nyakatura, G., Taudien, S., Rump, A., Yang, H., Yu, J., Wang, J., Huang, G., Gu, J., Hood, L., Rowen, L., Madan, A., Qin, S., Davis, R. W., Federspiel, N. A., Abola, A. P., Proctor, M. J., Myers, R. M., Schmutz, J., Dickson, M., Grimwood, J., Cox, D. R., Olson, M. V., Kaul, R., Raymond, C., Shimizu, N., Kawasaki, K., Minoshima, S., Evans, G. A., Athanasiou, M., Schultz, R., Roe, B. A., Chen, F., Pan, H., Ramser, J., Lehrach, H., Reinhardt, R., McCombie, W. R., de la Bastide, M., Dedhia, N., Blöcker, H., Hornischer, K., Nordsiek, G., Agarwala, R., Aravind, L., Bailey, J. A., Bateman, A., Batzoglou, S., Birney, E., Bork, P., Brown, D. G., Burge, C. B., Cerutti, L., Chen, H. C., Church, D., Clamp, M., Copley, R. R., Doerks, T., Eddy, S. R., Eichler, E. E., Furey, T. S., Galagan, J., Gilbert, J. G., Harmon, C., Hayashizaki, Y., Haussler, D., Hermjakob, H., Hokamp, K., Jang, W., Johnson, L. S., Jones, T. A., Kasif, S., Kaspryzk, A., Kennedy, S., Kent, W. J., Kitts, P., Koonin, E. V., Korf, I., Kulp, D., Lancet, D., Lowe, T. M., McLysaght, A., Mikkelsen, T., Moran, J. V., Mulder, N., Pollara, V. J., Ponting, C. P., Schuler, G., Schultz, J., Slater, G., Smit, A. F., Stupka, E., Szustakowski, J., Thierry-Mieg, D., Thierry-Mieg, J., Wagner, L., Wallis, J., Wheeler, R., Williams, A., Wolf, Y. I., Wolfe, K. H., Yang, S. P., Yeh, R. F., Collins, F., Guyer, M. S., Peterson, J., Felsenfeld, A., Wetterstrand, K. A., Patrinos, A., Morgan, M. J., de Jong, P., Catanese, J. J., Osoegawa, K., Shizuya, H., Choi, S., Chen, Y. J., and Szustakowki, J. (2001). Initial sequencing and analysis of the human genome. *Nature*, 409(6822) :860–921.

Langmead, B. and Salzberg, S. L. (2012a). Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4) :357–359.

Langmead, B. and Salzberg, S. L. (2012b). Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4) :357–359.

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biology*, 10(3) :R25.

Lemaitre, C., Ciortuz, L., and Peterlongo, P. (2014). Mapping-free and assembly-free discovery of inversion breakpoints from raw NGS reads. In Dediu, A.-H., Martín-Vide, C., and Truthe, B., editors, *Algorithms for Computational Biology*, volume 8542, pages 119–130, Tarragona, Spain.

Li, D., Liu, C.-M., Luo, R., Sadakane, K., and Lam, T.-W. (2015). MEGAHIT : an ultra-fast

single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics*, 31(10) :1674–1676.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14) :1754–1760.

Lieber, M. R., Ma, Y., Pannicke, U., and Schwarz, K. (2003). Mechanism and regulation of human non-homologous DNA end-joining. *Nature reviews. Molecular cell biology*, 4(9) :712–720.

Lima-Mendez, G., Faust, K., Henry, N., Decelle, J., Colin, S., Carcillo, F., Chaffron, S., Ignacio-Espinosa, J. C., Roux, S., Vincent, F., Bittner, L., Darzi, Y., Wang, J., Audic, S., Berline, L., Bontempi, G., Cabello, A. M., Coppola, L., Cornejo-Castillo, F. M., D'Ovidio, F., De Meester, L., Ferrera, I., Garet-Delmas, M.-J., Guidi, L., Lara, E., Pesant, S., Royo-Llonch, M., Salazar, G., Sanchez, P., Sebastian, M., Souffreau, C., Dimier, C., Picheral, M., Searson, S., Kandels-Lewis, S., Gorsky, G., Not, F., Ogata, H., Speich, S., Stemmann, L., Weissenbach, J., Wincker, P., Acinas, S. G., Sunagawa, S., Bork, P., Sullivan, M. B., Karsenti, E., Bowler, C., de Vargas, C., and Raes, J. (2015). Determinants of community structure in the global plankton interactome. *Science*, 348(6237) :1262073–1262073.

Limasset, A. and Peterlongo, P. (2015). Read Mapping on de Bruijn graph. *arXiv preprint arXiv :1505.04911*.

Lodish, H. F., Berk, A., Zipursky, S. L., Matsudaira, P., Baltimore, D., Darnell, J., et al. (2000). *Molecular cell biology*, volume 4. Citeseer.

Loux, V., Mariadassou, M., Almeida, S., Chiapello, H., Hammani, A., Buratti, J., Gendrault, A., Barbe, V., Aury, J.-M., Deutsch, S.-M., Parayre, S., Madec, M.-N., Chuat, V., Jan, G., Peterlongo, P., Azevedo, V., Le Loir, Y., and Falentin, H. (2015). Mutations and genomic islands can explain the strain dependency of sugar utilization in 21 strains of Propionibacterium freudenreichii. *BMC Genomics*, 16(1) :35.

Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., Yuan, J., He, G., Chen, Y., Pan, Q., Liu, Y., et al. (2012). Soapdenovo2 : an empirically improved memory-efficient short-read de novo assembler. *GigaScience*, 1(1) :18.

Magurran, A. E. and Henderson, P. A. (2012). How selection structures species abundance distributions. *Proceedings of the Royal Society B : Biological Sciences*, 279(1743) :3722–3726.

Maillet, N. (2013). *Comparaison de novo de données de séquençage issues de très grands échantillons métagénomiques : application sur le projet Tara Oceans*. PhD thesis, Rennes 1.

Maillet, N., Collet, G., Vannier, T., Lavenier, D., and Peterlongo, P. (2014). COMMET : comparing and combining multiple metagenomic datasets. In *IEEE BIBM 2014*, Belfast, United Kingdom.

Maillet, N., Lemaitre, C., Chikhi, R., Lavenier, D., and Peterlongo, P. (2012). Compareads : comparing huge metagenomic experiments. In *RECOMB Comparative Genomics 2012*, Niterói, Brazil.

Marçais, G. and Kingsford, C. (2011). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6) :764–770.

Mardis, E. R. (2011). A decade's perspective on DNA sequencing technology. *Nature*, 470(7333) :198–203.

Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y.-J., Chen, Z., Dewell, S. B., Du, L., Fierro, J. M., Gomes, X. V., Godwin, B. C., He, W., Helgesen, S., Ho, C. H., Irzyk, G. P., Jando, S. C., Alenquer, M. L. I., Jarvie, T. P., Jirage, K. B., Kim, J.-B., Knight, J. R., Lanza, J. R., Leamon, J. H., Lefkowitz, S. M., Lei, M., Li, J., Lohman, K. L., Lu, H., Makhijani, V. B., McDade, K. E., McKenna, M. P., Myers, E. W., Nickerson, E., Nobile, J. R., Plant, R., Puc, B. P., Ronan, M. T., Roth, G. T., Sarkis, G. J., Simons, J. F., Simpson, J. W., Srinivasan, M., Tartaro, K. R., Tomasz, A., Vogt, K. A., Volkmer, G. A., Wang, S. H., Wang, Y., Weiner, M. P., Yu, P., Begley, R. F., and Rothberg, J. M. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057) :376–380.

McKenna, A. H., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., Garimella, K., Altshuler, D., Gabriel, S., Daly, M., and Depristo, M. (2010). The Genome Analysis Toolkit : A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, 20(9) :1297–1303.

Modrek, B. and Lee, C. J. (2003). Alternative splicing in the human, mouse and rat genomes is associated with an increased frequency of exon creation and/or loss. *Nature genetics*, 34(2) :177–180.

Myers, E. W. (2005). The fragment assembly string graph. *Bioinformatics*, 21(Suppl 2) :ii79–ii85.

Myers, E. W., Sutton, G. G., Delcher, A. L., Dew, I. M., Fasulo, D. P., Flanigan, M. J., Kravitz, S. A., Mobarry, C. M., Reinert, K. H., Remington, K. A., Anson, E. L., Bolanos, R. A., Chou, H. H., Jordan, C. M., Halpern, A. L., Lonardi, S., Beasley, E. M., Brandon, R. C., Chen, L., Dunn, P. J., Lai, Z., Liang, Y., Nusskern, D. R., Zhan, M., Zhang, Q., Zheng, X., Rubin, G. M., Adams, M. D., and Venter, J. C. (2000). A whole-genome assembly of Drosophila. *Science (New York, N.Y.)*, 287(5461) :2196–2204.

Nagarajan, N. and Pop, M. (2009). Parametric Complexity of Sequence Assembly : Theory and Applications to Next Generation Sequencing. *Journal of Computational Biology*, 16(7) :897–908.

Namiki, T., Hachiya, T., Tanaka, H., and Sakakibara, Y. (2012). MetaVelvet : an extension of Velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic Acids Research*, 40(20) :e155–e155.

O'Driscoll, M. and Jeggo, P. A. (2006). The role of double-strand break repair - insights from human genetics. *Nature reviews. Genetics*, 7(1) :45–54.

on Biochem. Nomenclature (CBN), I.-I. C. (1970). Abbreviations and symbols for nucleic acids, polynucleotides, and their constituents. *Biochemistry*, 9(20) :4022–4027.

Peng, Y., Leung, H. C. M., Yiu, S. M., and Chin, F. Y. L. (2011). Meta-IDBA : a de Novo assembler for metagenomic data. *Bioinformatics*, 27(13) :i94–i101.

Peterlongo, P. (2006). *DNA sequence filtration for the problem of finding long multiple repetitions*. Theses, Université de Marne la Vallée.

Peterlongo, P., Allali, J., and Sagot, M. (2006). The gapped-factor tree. *The Prague Stringology Conference*.

Peterlongo, P., Allali, J., and Sagot, M.-F. (2007a). Indexing gapped-factors using a tree. *International Journal of Foundations of Computer Science*.

Peterlongo, P. and Chikhi, R. (2012). Mapsembler, targeted and micro assembly of large NGS datasets on a desktop computer. *BMC Bioinformatics*, 13(1) :48.

Peterlongo, P., Nicolas, J., Lavenier, D., Vorc'H, R., and Querellou, J. (2009a). c-GAMMA : Comparative Genome Analysis of Molecular Markers. In LNCS, editor, *Pattern Recognition in Bioinformatics*, volume 5780/2009 of *Pattern Recognition in Bioinformatics*, pages 255–269, Sheffield, United Kingdom. SpringerLink.

Peterlongo, P., Noé, L., Lavenier, D., Georges, G., Jacques, J., Kucherov, G., and Giraud, M. (2007b). Protein similarity search with subset seeds on a dedicated reconfigurable hardware. In *Parallel Bio-Computing*, Gdansk,, Poland.

Peterlongo, P., Noé, L., Lavenier, D., Nguyen, V. H., Kucherov, G., and Giraud, M. (2008a). Optimal neighborhood indexing for protein similarity search. *BMC bioinformatics*, 9(1) :534.

Peterlongo, P., Pisanti, N., Boyer, F., Pereira Do Lago, A., and Sagot, M.-F. (2008b). Lossless filter for multiple repetitions with Hamming distance. *Journal of Discrete Algorithms*, 6(3) :497–509.

Peterlongo, P., Sacomoto, Gustavo, A. T., Do Lago, Alair, P., Pisanti, N., and Sagot, M.-F. (2009b). Lossless filter for multiple repeats with bounded edit distance. *Algorithms for Molecular Biology*, 4(1) :3.

Peterlongo, P., Schnel, N., Pisanti, N., Sagot, M.-F., and Lacroix, V. (2010). Identifying snps without a reference genome by comparing raw reads. In *String Processing and Information Retrieval*, pages 147–158. Springer.

Pevzner, P. A., Tang, H., and Waterman, M. S. (2001). An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17) :9748–9753.

Pisanti, N., Giraud, M., and Peterlongo, P. (2010). Filters and seeds approaches for fast homology searches in large datasets. In Elloumi, M. and Zomaya, A. Y., editors, *Algorithms in computational molecular biology*. John Wiley & sons.

Port, J. A., Wallace, J. C., Griffith, W. C., and Faustman, E. M. (2012). Metagenomic Profiling of Microbial Composition and Antibiotic Resistance Determinants in Puget Sound. *PLoS ONE*, 7(10).

Proctor, L. M. (2015). Overview of the Phase One (2007-2012) of the NIH Human Microbiome Project. In *Encyclopedia of Metagenomics*, pages 488–494. Springer US, Boston, MA.

Qin, J., Li, R., Raes, J., Arumugam, M., Burgdorf, K. S., Manichanh, C., Nielsen, T., Pons, N., Levenez, F., Yamada, T., Mende, D. R., Li, J., Xu, J., Li, S., Li, D., Cao, J., Wang, B., Liang, H., Zheng, H., Xie, Y., Tap, J., Lepage, P., Bertalan, M., Batto, J.-M., Hansen, T., Le Paslier, D., Linneberg, A., Nielsen, H. B. r., Pelletier, E., Renault, P., Sicheritz-Ponten, T., Turner, K., Zhu, H., Yu, C., Li, S., Jian, M., Zhou, Y., Li, Y., Zhang, X., Li, S., Qin, N., Yang, H., Wang, J., Brunak, S. r., Doré, J., Guarner, F., Kristiansen, K., Pedersen, O., Parkhill, J., Weissenbach, J., Antolin, M., Artiguenave, F., Blottiere, H., Borruel, N., Bruls, T., Casellas, F., Chervaux, C., Cultrone, A., Delorme, C., Denariaz, G., Dervyn, R., Forte, M., Friss, C., van de Guchte, M., Guedon, E., Haimet, F., Jamet, A., Juste, C., Kaci, G., Kleerebezem, M., Knol, J., Kristensen, M., Layec, S., Le Roux, K., Leclerc, M., Maguin, E., Melo Minardi, R., Oozeer, R., Rescigno, M., Sanchez, N., Tims, S., Torrejon, T., Varela, E., de Vos, W., Winogradsky, Y., Zoetendal, E., Bork, P., Ehrlich, S. D., and Wang, J. (2010). A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, 464(7285) :59–65.

Quillery, E., Quenez, O., Peterlongo, P., and Plantard, O. (2014). Development of genomic resources for the tick Ixodes ricinus : isolation and characterization of single nucleotide polymorphisms. *Molecular ecology resources*, 14(2) :393–400.

Rezzonico, F. (2014). Nanopore-Based Instruments as Biosensors for Future Planetary Missions. *Astrobiology*, 14(4) :344–351.

Riou, C., Lemaitre, C., and Peterlongo, P. (2015). VCF_creator : Mapping and VCF Creation features in DiscoSnp++ . JOBIM 2015.

Rizk, G., Lavenier, D., and Chikhi, R. (2013). DSK : K-mer counting with very low memory usage. *Bioinformatics*, 29(5) :652–653.

Rusch, D. B., Halpern, A. L., Sutton, G., Heidelberg, K. B., Williamson, S., Yooseph, S., Wu, D., Eisen, J. A., Hoffman, J. M., Remington, K., Beeson, K., Tran, B., Smith, H., Baden-Tillson, H., Stewart, C., Thorpe, J., Freeman, J., Andrews-Pfannkoch, C., Venter, J. E., Li, K., Kravitz, S., Heidelberg, J. F., Utterback, T., Rogers, Y.-H., Falcón, L. I., Souza, V., Bonilla-Rosso, G., Eguiarte, L. E., Karl, D. M., Sathyendranath, S., Platt, T., Bermingham, E., Gallardo, V., Tamayo-Castillo, G., Ferrari, M. R., Strausberg, R. L., Nealson, K., Friedman, R., Frazier, M., and Venter, J. C. (2007). The Sorcerer II Global Ocean Sampling Expedition : Northwest Atlantic through Eastern Tropical Pacific. *PLoS Biology*, 5(3) :e77.

Sacomoto, G. A., Kielbassa, J., Chikhi, R., Uricaru, R., Antoniou, P., Sagot, M.-F., Peterlongo, P., and Lacroix, V. (2012). Kissplice : de-novo calling alternative splicing events from rna-seq data. *BMC bioinformatics*, 13(Suppl 6) :S5.

Sanger, F., Nicklen, S., and Coulson, A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, 74(12) :5463–5467.

Shakya, M., Quince, C., Campbell, J. H., Yang, Z. K., Schadt, C. W., and Podar, M. (2013). Comparative metagenomic and rRNA microbial diversity characterization using archaeal and bacterial synthetic communities. *Environmental Microbiology*, 15(6) :1882–1899.

Sheridan, C. (2014). Illumina claims $1,000 genome win. *Nature biotechnology*, 32(2) :115.

Snir, S. and Pachter, L. (2006). Phylogenetic profiling of insertions and deletions in vertebrate genomes. In *Research in Computational Molecular Biology*, pages 265–280. Springer.

Snir, S. and Pachter, L. (2011). Tracing the most parsimonious indel history. *Journal of Computational Biology*, 18(8) :967–986.

Thorvaldsdóttir, H., Robinson, J. T., and Mesirov, J. P. (2012). Integrative genomics viewer (igv) : high-performance genomics data visualization and exploration. *Briefings in bioinformatics*, page bbs017.

Uricaru, R., Rizk, G., Lacroix, V., Quillery, E., Plantard, O., Chikhi, R., Lemaitre, C., and Peterlongo, P. (2015). Reference-free detection of isolated snps. *Nucleic acids research*, 43(2) :e11–e11.

Van Doorn, G. and Kirkpatrick, M. (2007). Turnover of sex chromosomes induced by sexual conflict. *Nature*, 449(7164) :909–912.

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., Smith, H. O., Yandell, M., Evans, C. A., Holt, R. A., Gocayne, J. D., Amanatides, P., Ballew, R. M., Huson, D. H., Wortman, J. R., Zhang, Q., Kodira, C. D., Zheng, X. H., Chen, L., Skupski, M., Subramanian, G., Thomas, P. D., Zhang, J., Gabor Miklos, G. L., Nelson, C., Broder, S., Clark, A. G., Nadeau, J., McKusick, V. A., Zinder, N., Levine, A. J., Roberts, R. J., Simon, M., Slayman, C., Hunkapiller, M., Bolanos, R., Delcher, A., Dew, I., Fasulo, D., Flanigan, M., Florea, L., Halpern, A., Hannenhalli, S., Kravitz, S., Levy, S., Mobarry, C., Reinert, K., Remington, K., Abu-Threideh, J., Beasley, E., Biddick, K., Bonazzi, V., Brandon, R., Cargill, M., Chandramouliswaran, I., Charlab, R., Chaturvedi, K., Deng, Z., Di Francesco, V., Dunn, P., Eilbeck, K., Evangelista, C., Gabrielian, A. E., Gan, W., Ge, W., Gong, F., Gu, Z., Guan, P., Heiman, T. J., Higgins, M. E., Ji, R. R., Ke, Z., Ketchum, K. A., Lai, Z., Lei, Y., Li, Z., Li, J., Liang, Y., Lin, X., Lu, F., Merkulov, G. V., Milshina, N., Moore, H. M., Naik, A. K., Narayan, V. A., Neelam, B., Nusskern, D., Rusch, D. B., Salzberg, S., Shao, W., Shue, B., Sun, J., Wang, Z., Wang, A., Wang, X., Wang, J., Wei, M.-h., Wides, R., Xiao, C., Yan, C., Yao, A., Ye, J., Zhan, M., Zhang, W., Zhang, H., Zhao, Q., Zheng, L., Zhong, F., Zhong, W., Zhu, S., Zhao, S., Gilbert, D., Baumhueter, S., Spier, G., Carter, C., Cravchik, A., Woodage, T., Ali, F., An, H., Awe, A., Baldwin, D., Baden, H., Barnstead, M., Barrow, I., Beeson, K., Busam, D., Carver, A., Center, A., Cheng, M. L., Curry, L., Danaher, S., Davenport, L., Desilets, R., Dietz, S., Dodson, K., Doup, L., Ferriera, S., Garg, N., Gluecksmann, A., Hart, B., Haynes, J., Haynes, C., Heiner, C., Hladun, S., Hostin, D., Houck, J., Howland, T., Ibegwam, C., Johnson, J., Kalush, F., Kline, L., Koduru, S., Love, A., Mann, F., May, D., McCawley, S., McIntosh, T., McMullen, I., Moy, M., Moy, L., Murphy, B., Nelson, K., Pfannkoch, C., Pratts, E., Puri, V., Qureshi, H., Reardon, M., Rodriguez, R., Rogers, Y. H., Romblad, D., Ruhfel, B., Scott, R., Sitter, C., Smallwood, M., Stewart, E., Strong, R., Suh, E., Thomas, R., Tint, N. N., Tse, S., Vech, C., Wang, G., Wetter, J., Williams, S.,

Williams, M., Windsor, S., Winn-Deen, E., Wolfe, K., Zaveri, J., Zaveri, K., Abril, J. F., Guigó, R., Campbell, M. J., Sjolander, K. V., Karlak, B., Kejariwal, A., Mi, H., Lazareva, B., Hatton, T., Narechania, A., Diemer, K., Muruganujan, A., Guo, N., Sato, S., Bafna, V., Istrail, S., Lippert, R., Schwartz, R., Walenz, B., Yooseph, S., Allen, D., Basu, A., Baxendale, J., Blick, L., Caminha, M., Carnes-Stine, J., Caulk, P., Chiang, Y. H., Coyne, M., Dahlke, C., Mays, A., Dombroski, M., Donnelly, M., Ely, D., Esparham, S., Fosler, C., Gire, H., Glanowski, S., Glasser, K., Glodek, A., Gorokhov, M., Graham, K., Gropman, B., Harris, M., Heil, J., Henderson, S., Hoover, J., Jennings, D., Jordan, C., Jordan, J., Kasha, J., Kagan, L., Kraft, C., Levitsky, A., Lewis, M., Liu, X., Lopez, J., Ma, D., Majoros, W., McDaniel, J., Murphy, S., Newman, M., Nguyen, T., Nguyen, N., Nodell, M., Pan, S., Peck, J., Peterson, M., Rowe, W., Sanders, R., Scott, J., Simpson, M., Smith, T., Sprague, A., Stockwell, T., Turner, R., Venter, E., Wang, M., Wen, M., Wu, D., Wu, M., Xia, A., Zandieh, A., and Zhu, X. (2001). The sequence of the human genome. *Science (New York, N.Y.)*, 291(5507) :1304–51.

Villar, E., Farrant, G. K., Follows, M., Garczarek, L., Speich, S., Audic, S., Bittner, L., Blanke, B., Brum, J. R., Brunet, C., Casotti, R., Chase, A., Dolan, J. R., D'Ortenzio, F., Gattuso, J.-P., Grima, N., Guidi, L., Hill, C. N., Jahn, O., Jamet, J.-L., Le Goff, H., Lepoivre, C., Malviya, S., Pelletier, E., Romagnan, J.-B., Roux, S., Santini, S., Scalco, E., Schwenck, S. M., Tanaka, A., Testor, P., Vannier, T., Vincent, F., Zingone, A., Dimier, C., Picheral, M., Searson, S., Kandels-Lewis, S., Acinas, S. G., Bork, P., Boss, E., de Vargas, C., Gorsky, G., Ogata, H., Pesant, S., Sullivan, M. B., Sunagawa, S., Wincker, P., Karsenti, E., Bowler, C., Not, F., Hingamp, P., and Iudicone, D. (2015). Environmental characteristics of Agulhas rings affect interocean plankton transport. *Science*, 348(6237) :1261447–1261447.

Vroland, C., Salson, M., and Touzet, H. (2014). Lossless seeds for searching short patterns with high error rates. In *Combinatorial Algorithms - 25th International Workshop, IWOCA 2014, Duluth, MN, USA, October 15-17, 2014, Revised Selected Papers*, pages 364–375.

Wang, M., Ye, Y., and Tang, H. (2012). A de Bruijn Graph Approach to the Quantification of Closely-Related Genomes in a Microbial Community. *Journal of Computational Biology*, 19(6) :814–825.

Whon, T. W., Kim, M.-S., Roh, S. W., Shin, N.-R., Lee, H.-W., and Bae, J.-W. (2012). Metagenomic characterization of airborne viral dna diversity in the near-surface atmosphere. *Journal of virology*, 86(15) :8221–8231.

Wong, K., Bumpstead, S., Van Der Weyden, L., Reinholdt, L. G., Wilming, L. G., Adams, D. J., and Keane, T. M. (2012). Sequencing and characterization of the FVB/NJ mouse genome. *Genome biology*, 13(8) :R72.

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

Wooley, J. C., Godzik, A., and Friedberg, I. (2010). A Primer on Metagenomics. *PLoS Computational Biology*, 6(2) :e1000667.

Wright, S. (1949). Adaptation and selection. *Genetics, paleontology and evolution*, pages 365–389.

Zerbino, D. R. and Birney, E. (2008). Velvet : Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, 18(5) :821–829.

Zhao, M., Lee, W.-P., Garrison, E. P., and Marth, G. T. (2013). SSW Library : An SIMD Smith-Waterman C/C++ Library for Use in Genomic Applications. *PLoS ONE*, 8(12) :e82138.

Curriculum Vitæ
Pierre Peterlongo

État civil

Née le	2 juillet 1978 **à** Clermont-Ferrand, France
Contacts	
	Téléphone +33.2.99.84.74.59
	Fax +33.2.99.84.71.71
	email pierre.peterlongo@inria.fr
	page personnelle http ://people.rennes.inria.fr/Pierre.Peterlongo
	adresse Inria, Campus de Beaulieu, 35042 Rennes Cedex
Situation actuelle	Chargé de recherche Inria, Équipe-projet Genscale, Rennes.

Formation scientifique

- **2003-2006** – Doctorat d'informatique de l'Université de Marne-la-Vallée, institut Gaspard-Monge, défendue le 30 septembre 2006.
  - **Titre** : Filtrage de séquences d'ADN pour la recherche de longues répétitions multiples.
  - **Directeurs de thèse**
    - Marie-France Sagot
    - Maxime Crochemore
  - **Rapporteurs**
    - Mathieu Blanchette
    - Christian Gautier
    - Gregory Kucherov
  - **Membres du jury additionnels**
    - Nadia Pisanti
    - Jean Berstel
    - Thierry Lecroq
- **2002-2003** – Master (DEA) d'informatique de l'Institut Gaspard-Monge, Université de Marne-la-Vallée.
  - **Titre** Détection de motifs répétés et de gènes par alignement multiple. Filtrage des données
  - **Encadrants**

  - ▹ Marie-France Sagot
  - ▹ Maxime Crochemore
- ▶ **1999-2002** – DEUG & Licence d'informatique de l'Institut Gaspard-Monge, Université de Marne-la-Vallée.

Expériences professionnelles

- ▶ **Depuis 2008** – Chargé de recherche à plein temps à Inria, Rennes Bretagne Atlantique
- ▶ **Depuis 2009** – Enseignant vacataire à l'Université de Rennes 1 (Master Bio-info) et à l'Istic (précédemment Ifsic). Niveau Master
- ▶ **2006-2008** – Post-doctorat à Inria, Rennes Bretagne Atlantique
- ▶ **2003-2006** – Doctorat d'informatique et enseignant Moniteur à l'Université de Marne-la-Vallée
- ▶ **2002-2003** – Enseignant vacataire à l'Université de Marne-la-Vallée

Principales contributions à l'animation et à la diffusion de l'information scientifique

Enseignement J'ai été responsable de deux Unités d'Enseignement en Master1 Info et en Master2 Bio-Informatique de Génomes (BIG). Ces UEs existent encore à l'heure actuelle et je suis responsable de l'UE en Master 1. Ces UEs explorent les techniques algorithmiques associées à la recherche dans de grandes masses de données textuelles ou structurelles. Elles présentent des techniques de Pattern Matching classiques jusqu'aux algorithmes de mapping ou d'assemblage utilisant des structures de données poussées comme le FM-index. Ces UEs sont adaptées au public (plus algorithmique en M1 info et plus appliqué en M2 BIG). J'ai orchestré ces deux UEs avec d'autres personnels enseignants, ce qui permet d'offrir un plus large point de vue aux étudiants.

Formation J'ai participé en 2015 à la formation "*CNRS Formation Entreprise*" appelée "*Bioinformatique pour le traitement de données de séquençage (NGS)*". Celle-ci m'a permis de présenter l'outil discoSnp++ et d'organiser une session de travaux pratiques avec de futurs utilisateurs. Cette formation sera reconduite au moins en 2016.

Médiation Je considère que l'une de mes missions est de participer à démocratiser les savoirs et les outils du numérique. Dans cette optique j'ai participé 6 années de suite à l'initiative "*À la découverte de la recherche*", consistant à exposer en classes de lycée, le domaine de la bio-informatique et la vie de chercheur. Dans la même optique, j'ai assuré des conférences en lycée et pour les Olympiades des mathématiques. J'ai également été impliqué dans la médiation des contenus scientifique (mécsi) en tant que représentant du centre Rennes, Bretagne-Atlantique.

Séminaire interne Symbiose J'organise depuis 2011 le séminaire Symbiose réunissant les équipes GenOuest, Dyliss et GenScale. Il s'agit d'un rendez-vous hebdomadaire où un invité extérieur présente ses travaux. Ce séminaire est un espace d'échange riche et d'ouverture de collaborations.

Structuration de la communauté

- J'ai organisé plusieurs groupes de travail liés à la bio-informatique. Il s'agissait de mettre en relation biologistes et informaticiens pour permettre une mise à niveau dans le domaine de la biologie ou autour du sujet central des NGS. Dans ce domaine évoluant extrêmement rapidement, la demande des utilisateurs est particulièrement forte. Ce groupe de travail a connu un important succès (une cinquantaine de participants locaux et en visio).
- Jobim 2012 http://jobim2012.inria.fr/ : De concert avec Claire Lemaitre, j'ai co-organisé la treizième conférence Jobim 2012. Il s'agit du rendez-vous annuel francophone de la recherche en bio-informatique. Cette conférence est particulièrement importante puisqu'elle réunit près de 400 chercheurs et ingénieurs, ainsi que les principaux acteurs industriels du domaine. Elle permet à la communauté d'échanger (en 2012 45 présentations plénières ou parallèles et 110 posters ont été présentés) et de se structurer ou d'interagir avec les industriels du domaine (nous y avons accueilli *Korilog*, *Fasteris*, *Microsoft-Inria*, *Panassas*, *NVidia*, *Genostar*, et *Ingenuity Systems*). Cette manifestation permet bien évidemment également d'accroître la visibilité des équipes de recherches GenScale et Dyliss et du centre Inria Rennes-Atlantique.
- Je suis animateur de l'axe scientifique "*Analyse des séquences*" du GdR BIM (Bio-Informatique Moléculaire).
- En marge de ces aspects purement scientifiques, je suis secrétaire de l'AGOS (Association pour la Gestion des Oeuvres Sociales de l'Inria).

Activités scientifiques

Comités de recrutement

J'ai fait partie de divers comité de recrutements, dont en particulier :

- 2014&2015 - Jury aux concours d'attribution à l'IRISA de bourses ministérielles d'allocation de recherche MESR
- 2014 - Jury au concours de maître de conférences 27MCF1570 à Toulouse
- 2012 - Jury aux concours de maître de conférences 27MCF1107 à Bordeaux et 6400MCF1193 à Brest
- 2011 - Jury pour deux postes CDI d'ingénieurs / chercheurs au Génoscope, Evry
- 2011 - Jury au concours de maître de conférences 27MCF0923 à Nantes

Suivi et encadrement d'étudiants

Encadrement d'étudiants en thèse

- **Antoine Limasset** - Thèse de doctorat, en cours débutée en 2014. "*Nouvelles approches pour l'exploitation des données de séquençage génomiques haut débit*". Cette thèse est dirigée par Dominique Lavenier. Je la co-encadre avec Claire Lemaitre. Le sujet est lié au travaux présentés Chapitre 4 page 65 visant à offrir de nouvelles méthodes pour exploiter au mieux les données de séquençages pour proposer de meilleurs assemblages et une meilleure détection des variants génomiques.

- **Camille Marchet** - Thèse de doctorat, en cours débutée en 2015. "*Nouvelles méthodologies pour l'assemblage de données de séquençage polymorphes*". Une dérogation me permet d'encadrer Camille Marchet à100%. Le sujet est en lien avec l'analyse de données des séquenceurs de troisième génération.
- **Nicolas Maillet** - Thèse de doctorat 2010-2013. "*Algorithmes pour l'assemblage des données NGS*". J'ai co-encadré cette thèse avec Dominique Lavenier. Il s'agissait de proposer de nouveaux algorithmes pour extraire de novo de l'information biologique dans les données de métagénomique générées par le projet TARA-ocean. Les travaux issus de cette thèse sont le point d'origine de ce qui est présenté dans le Chapitre 3 page 41. Cette thèse est financée par le projet ANR Mappi.

Comité de thèse

- Je participe au comité de thèse de **Léa Siegwald** sur le sujet "*Évaluation, optimisation, développement d'outils informatiques permettant l'analyse de données de métagénomique ciblée*", encadrée par Yves Lemoine et Hélène Touzet.
- Je participe au comité de thèse d'**Alix Mas** sur le sujet "*Évolution et traits fonctionnels en milieux hétérogènes.*", encadrée par Yvan Lagadeuc et de Philippe Vandenkoornhuyse à Écobio à Rennes.
- Je participe au comité de thèse d'**Andrea Radulescu** sur le sujet "*Traitement algorithmique de données NGS*", encadrée au Lina par Géraldine Jean et Irena Rusu.
- J'ai participé au comité de thèse de **Tiayyba Riaz** sur le sujet "*Approches bioinformatiques pour l'évaluation de la biodiversité*", encadrée au LECA à Grenoble par Éric Coissac. J'ai également participé au jury de sa soutenance de thèse en novembre 2011.

Jury de thèse En plus de faire partie du jury de soutenance de thèse de Nicolas Maillet, j'ai participé au jury de soutenance de Tiayyba Riaz.

Encadrement de stages, d'ingénieurs et de post-doctorats

- **Antoine Limasset** - 2015 - Stage M2 - "Amélioration d'assemblage par mapping de reads et phasing." Antoine à travaillé sur une première version de méthodes de mapping de lectures sur dBG. Ces travaux sont en cours de soumission pour une publication en conférence. Antoine a obtenu une bourse de thèse ministèrielle. Je co-encadre avec Claire Lemaitre ses travaux de thèse.
- **Chloé Riou** - Depuis Octobre 2014 - Ingénieur projet ANR "Colib'read". Chloé est co-encadrée par Claire Lemaitre et moi-même. Elle travaille sur les nouveaux développements de l'outil discoSnp++ et en particulier la mise en oeuvre de l'outil VCF_creator.
- **Estelle Lecluze** - Mars 2014 à Juin 2014 - Stage M1 sur le sujet "Nouvelles méthodes de traitement et d'assemblage de métagénomes de novo appliquées au projet TaraOcean".
- **Alexan Andrieux** – Septembre 2012 à Septembre 2014 – Ingénieur jeune diplômé. Alexan a eu pour mission de finaliser le code du logiciel Mapsembler, ainsi que d'en proposer une interface graphique pour rendre le logiciel plus accessible et le préparer à un transfert.

- **Erwan Scaon** – Mars 2012 à Juin 2012 – À la suite d'un stage que j'ai encadré, Erwan a été sélectionné pour obtenir une bourse ministérielle pour travailler sur l'assemblage de données polyploïdes. Erwan n'a pas terminé sa thèse.
- **Liviu Ciortuz** – Octobre 2012 - Septembre 2013 - Conjointement avec Claire Lemaitre, nous avons encadré Liviu, post doctorant travaillant sur la détection d'évènements d'inversions dans les données de séquençage haut débit. Ses travaux sont à l'origine de l'outil TakeABreak, présenté Chapitre 2 page 15.
- **Raluca Uricaru** - Septembre 2011 à septembre 2012 - Post Doctorante – Modélisation et création d'algorithmes pour la détection de SNP dans les données de séquençage haut débit. Raluca, maintenant maître de conférences au LaBri à Bordeaux, poursuit entre autres son travail dans ces thématiques.
- **Guillaume Holley** - Mai à Septembre 2011 et janvier à juin 2013 – Stage L3 et M2 Bio-informatique. Stage L3 : "Développement d'un plugin cytoscape pour la recherche d'éléments génomiques". J'ai été l'encadrant unique de ce stage adaptant les méthodes de pattern matching aux graphes utilisés pour l'assemblage des données NGS. Ce stage a été un succès, donnant lieux à un article présenté lors de la conférence PSC 2012. Stage M2 : "Comparaison massive et multiple de métagénomes non assemblés". Durant ce stage, les premiers outils de comparaisons statistiques de métagénomes non assemblés ont été développés. Ce stage a réuni les acteurs du projet ANR Hydrogen et en a été l'un des fondements qui ont motivé sa rédaction. Guillaume est actuellement en thèse sous la direction de Jens Stoye à l'université de Bielefeld, en Allemagne.
- **Pavlos Antoniou** - Septembre 2010 à septembre 2011 - Post Doctorant – "Extraction d'information provenant de données non assemblées issues de séquenceurs nouvelle génération". Pavlos a développé une nouvelle approche pour répondre à des questions biologiques précises dans les données de séquençage d'ARN (RNA-seq) en analysant des données générées par des nouvelles générations de séquenceurs. Pavlos a obtenu un poste permanent de chercheur à Ciprus à Chypre.
- **Nicolas Schnell** - Janvier à juin 2010 - Stage Master 2 Bio-informatique – "Approche méthodologique de détection de SNP par comparaison de données non assemblées". J'ai été encadrant unique de ce stage mêlant modélisation d'un problème biologique, création de l'algorithme associé, rédaction de article présenté à la conférence SPIRE 2010.
- **Jordan Thefaut** - Mai à juin 2010 - Stage Master 1 Information spécialité Systèmes et Réseaux – "Parallélisation d'un algorithme de bioinformatique : étude et mise en œuvre sur GPU". J'ai co-encadré avec Jacques Nicolas ce stage qui consistait à optimiser un code existant tout en améliorant le modèle biologique associé. Jordan est maintenant Développeur chez France Telecom R&D.

Comités de relecture

Durant ces dernières années j'ai participé aux comités scientifiques de **conférences** telles que *JOBIM*, *Recomb-Seq*, *SeqBio* ou *Wabi*. J'ai également été relecteur d'autres conférences à l'image de *IWOCA*, de *Spire*, et de *BIOKDD*, et d'**articles pour les journaux** *BMC-Bioinformatics*, *BMC-Algorithm for Molecular Biology*, *Bioinformatics*, *Frontiers*, *Nucleic Acid Research*, *International Journal of Foundations of Computer Science*, *PlosOne*, *Journal of Experimental algorithms*, *Recent*

Patents on DNA and Gene Sequence, ou *Algorithms for Molecular Biology*. Enfin j'ai participé au comité de relecture de l'**ANR "*emergence*"**.

Projets ANR

- **ANR Mappi**. J'ai été responsable de l'un des axes de ce projet ANR dirigé par Mathieu Raffinot. Ce projet était dédié à la mise en oeuvre de solutions algorithmiques pour le traitement de données métagénomiques à grande échelle. La thèse de Nicolas Maillet était financée par ce projet.
- **ANR Colib'read**. Je dirige l'ANR Colib'read depuis bientôt trois années. Les travaux présentés dans le second chapitre s'inscrivent dans ce projet ANR.
- **ANR Hydrogen**. Ce projet ANR, débuté fin 2014, dirigé par Dominique Lavenier vise à proposer de nouvelles méthodes performantes pour l'analyse de données métagénomiques à grande échelle, en particulier celles issues du projet TARA Oceans. Je suis responsable de l'un des axes de ce projet.

Liste de mes publications

Les publications de conférences ayant donné lieu à une publication journal ne sont pas reportées ici. Les communications sous forme de posters ne sont pas non plus reportées dans ce qui suit.

Chapitres de livre

▶ Jacques Nicolas, Sébastien Tempel, Pierre Peterlongo (In press). Finding and characterizing repeats in plant genomes. In Dave Edwards, editor, *Plant Bioinformatics : Methods and Protocols, Second Edition*. Humana Press

▶ Pisanti, N., Giraud, M., and Peterlongo, P. (2010). Filters and seeds approaches for fast homology searches in large datasets. In Elloumi, M. and Zomaya, A. Y., editors, *Algorithms in computational molecular biology*. John Wiley & sons

Journaux

▶ Loux, V., Mariadassou, M., Almeida, S., Chiapello, H., Hammani, A., Buratti, J., Gendrault, A., Barbe, V., Aury, J.-M., Deutsch, S.-M., Parayre, S., Madec, M.-N., Chuat, V., Jan, G., Peterlongo, P., Azevedo, V., Le Loir, Y., and Falentin, H. (2015). Mutations and genomic islands can explain the strain dependency of sugar utilization in 21 strains of Propionibacterium freudenreichii. *BMC Genomics*, 16(1) :35

▶ Uricaru, R., Rizk, G., Lacroix, V., Quillery, E., Plantard, O., Chikhi, R., Lemaitre, C., and Peterlongo, P. (2015). Reference-free detection of isolated snps. *Nucleic acids research*, 43(2) :e11–e11

▶ Federico, M., Peterlongo, P., Pisanti, N., and Sagot, M.-F. (2014). Rime : Repeat identification. *Discrete Applied Mathematics*, 163(3) :275–286

▶ Quillery, E., Quenez, O., Peterlongo, P., and Plantard, O. (2014). Development of genomic resources for the tick Ixodes ricinus : isolation and characterization of single nucleotide polymorphisms. *Molecular ecology resources*, 14(2) :393–400

▶ Drezen, E., Rizk, G., Chikhi, R., Deltel, C., Lemaitre, C., Peterlongo, P., and Lavenier, D. (2014). GATB : Genome Assembly & Analysis Tool Box. *Bioinformatics (Oxford, England)*, pages 1–3
▶ Maillet, N., Lemaitre, C., Chikhi, R., Lavenier, D., and Peterlongo, P. (2012). Compareads : comparing huge metagenomic experiments. In *RECOMB Comparative Genomics 2012*, Niterói, Brazil
▶ Sacomoto, G. A., Kielbassa, J., Chikhi, R., Uricaru, R., Antoniou, P., Sagot, M.-F., Peterlongo, P., and Lacroix, V. (2012). Kissplice : de-novo calling alternative splicing events from rna-seq data. *BMC bioinformatics*, 13(Suppl 6) :S5
▶ Peterlongo, P. and Chikhi, R. (2012). Mapsembler, targeted and micro assembly of large NGS datasets on a desktop computer. *BMC Bioinformatics*, 13(1) :48
▶ Gallé, M., Peterlongo, P., and Coste, F. (2009). In-place update of suffix array while recoding words. *International Journal of Foundations of Computer Science*, 20(6) :1025–1045
▶ Peterlongo, P., Sacomoto, Gustavo, A. T., Do Lago, Alair, P., Pisanti, N., and Sagot, M.-F. (2009b). Lossless filter for multiple repeats with bounded edit distance. *Algorithms for Molecular Biology*, 4(1) :3
▶ Peterlongo, P., Noé, L., Lavenier, D., Nguyen, V. H., Kucherov, G., and Giraud, M. (2008a). Optimal neighborhood indexing for protein similarity search. *BMC bioinformatics*, 9(1) :534
▶ Peterlongo, P., Pisanti, N., Boyer, F., Pereira Do Lago, A., and Sagot, M.-F. (2008b). Lossless filter for multiple repetitions with Hamming distance. *Journal of Discrete Algorithms*, 6(3) :497–509
▶ Peterlongo, P., Allali, J., and Sagot, M.-F. (2007a). Indexing gapped-factors using a tree. *International Journal of Foundations of Computer Science*
▶ Iliopoulos, C., Mchugh, J., Peterlongo, P., Pisanti, N., Rytter, W., and Sagot, M.-F. (2005). A first approach to finding common motifs with gaps,. *International Journal of Foundations of Computer Science*, 16(6) :1145–1155

Conférences internationales

▶ Maillet, N., Collet, G., Vannier, T., Lavenier, D., and Peterlongo, P. (2014). COMMET : comparing and combining multiple metagenomic datasets. In *IEEE BIBM 2014*, Belfast, United Kingdom
▶ Lemaitre, C., Ciortuz, L., and Peterlongo, P. (2014). Mapping-free and assembly-free discovery of inversion breakpoints from raw NGS reads. In Dediu, A.-H., Martín-Vide, C., and Truthe, B., editors, *Algorithms for Computational Biology*, volume 8542, pages 119–130, Tarragona, Spain
▶ Holley, G. and Peterlongo, P. (2012). BlastGraph : intensive approximate pattern matching in string graphs and de-Bruijn graphs. In *PSC 2012*, Prague, Czech Republic
▶ Federico, M., Peterlongo, P., Pisanti, N., and Sagot, M.-F. (2011). Finding Long and Multiple Repeats with Edit Distance. In *The Prague Stringology Conference 2011*, Prague, Czech Republic
▶ Peterlongo, P., Schnel, N., Pisanti, N., Sagot, M.-F., and Lacroix, V. (2010). Identifying snps without a reference genome by comparing raw reads. In *String Processing and Information Retrieval*, pages 147–158. Springer
▶ Federico, M., Peterlongo, P., and Pisanti, N. (2010). An optimized filter for finding multiple repeats in DNA sequences. In *2010 ACS/IEEE International Conference on Computer Systems and Applications*, HAMMAMET, Tunisia
▶ Peterlongo, P., Nicolas, J., Lavenier, D., Vorc'H, R., and Querellou, J. (2009a). c-GAMMA :

Comparative Genome Analysis of Molecular Markers. In LNCS, editor, *Pattern Recognition in Bioinformatics*, volume 5780/2009 of *Pattern Recognition in Bioinformatics*, pages 255–269, Sheffield, United Kingdom. SpringerLink

▶ Peterlongo, P., Noé, L., Lavenier, D., Georges, G., Jacques, J., Kucherov, G., and Giraud, M. (2007b). Protein similarity search with subset seeds on a dedicated reconfigurable hardware. In *Parallel Bio-Computing*, Gdansk,, Poland

▶ Antoniou, P., Crochemore, M., Iliopoulos, C., and Peterlongo, P. (2007). Application of suffix trees for the acquisition of common motifs with gaps in a set of strings. In *International Conference on Language and Automata Theory and Applications*, Tarragona, Spain

▶ Antoniou, P., Holub, J., Iliopoulos, C., Melichar, B., and Peterlongo, P. (2006). Finding Common Motifs with Gaps Using Finite Automata. In *Implementation and Application of Automata*, pages 69–77, Taipei, Taiwan

▶ Peterlongo, P., Allali, J., and Sagot, M. (2006). The gapped-factor tree. *The Prague Stringology Conference*

Manuscrit de thèse

▶ Peterlongo, P. (2006). *DNA sequence filtration for the problem of finding long multiple repetitions*. Theses, Université de Marne la Vallée