Méthodes sans alignement et indexation pour l'analyse de données nucléiques massives

Mikaël Salson

Méthodes sans alignement et indexation pour l'analyse de données nucléiques massives

HABILITATION À DIRIGER LES RECHERCHES
en informatique

présentée et soutenue le 21 novembre 2023 par

Mikaël SALSON

Présidente		
	Anne ETIEN	Professeure des universités, CRIStAL, Lille
Rapporteurs		
	Jérémie BOURDON	Professeur des universités, LS2N, Nantes
	Dominique LAVENIER	Directeur de recherche CNRS, IRISA, Rennes
	Yann PONTY	Directeur de recherche CNRS, LIX, Palaiseau
Examinatrice		
	Céline SCORNAVACCA	Directrice de recherche CNRS, ISEM, Montpellier
Garante		
	Hélène TOUZET	Directrice de recherche CNRS, CRIStAL, Lille

Centre de recherche en informatique signal et automatique de Lille — UMR 9189
Université de Lille

Remerciements

Merci à tous les membres du jury de cette HDR d'avoir accepté d'en faire partie et d'avoir subi le décalage temporel de la soutenance, indépendant de ma volonté (voir ci-dessous). Merci donc à Jérémie Bourdon, Dominique Lavenier et Yann Ponty d'avoir bien voulu endosser le rôle de rapporteurs. Merci à Céline Scornavacca d'être examinatrice. Merci Anne Etien d'assumer le rôle de présidente.

Merci bien évidemment à Hélène Touzet. D'une part pour avoir réussi à conduire l'équipe où elle en est à l'heure actuelle. Merci également de progressivement m'avoir amené à l'idée de rédiger cette HDR. Merci pour ta confiance. Bravo et merci à Camille Marchet d'avoir asséné le coup final, en me proposant de diriger la thèse dont elle venait d'avoir un financement.

Merci aux doctorants et doctorante que j'ai co-encadrés et sans qui une grande partie de tous ces travaux n'auraient pas été possibles. Merci à Christophe, Tatiana, Augustin et Thomas.

Merci à toute l'équipe Bonsai pour les nombreux bons moments malgré les périodes parfois plus difficiles. Merci à Antoine, Areski, Bastien, Coralie, Florian, Igor, Jean-Stéphane, Laurent, Léa, Lilian, Maude, Pierre, Stéphane, Thomas pour citer les membres (à peu près) actuels. Je ne m'aventurerai pas à citer tous les membres passés, de peur d'en oublier certains ou certaines, mais le cœur y est. Le plus sûr est donc de ne citer personne.

Merci à Martin Figeac qui, en venant nous proposer un projet de recherche en 2010, à ouvert une sacrée porte, encore largement ouverte. Merci à toute l'équipe (Aurélie, Claude, Nathalie, Nicolas, Stéphanie) du département d'hématologie du CHU de Lille pour leur patience, leur confiance et avoir amené le projet là où il en est. Merci Mathieu Giraud pour toutes ces années ensemble et cette belle aventure « Vidjil ». Merci également à toute l'équipe Vidjil, passée, présente et future (merci Aurélien, Clément, Marc, Florian, Ryan, Tatiana) pour votre travail indispensable sur ce projet.

Merci à Thérèse Commes et toute son équipe à Montpellier, pour cette longue collaboration, débutée en 2009, qui se poursuit toujours, de collaborations informelles en projets ANR. C'est notamment avec toi que j'ai commencé la bioinformatique, en travaillant sur des données de séquençage en santé… sujet que je n'ai pas quitté. Merci évidemment à Nicolas, pour cette fameuse discussion d'un soir de décembre 2008, qui m'a mis le pied à l'étrier de cette thématique et pour les nombreux projets menés ensuite.

Merci Rayan, comme ancien de Bonsai et comme actuel collaborateur. On n'aura pas réussi sur la k-bwt, mais par la suite, nos projets communs ont été plus fructueux !

Merci spécial à Sasha, stagiaire en temps de Covid que je n'ai jamais eu la chance de croiser, mais merci pour ton efficacité sur le projet FiLT3r. Merci également et bravo à Augustin pour ton efficacité et ta capacité d'apprentissage.

Merci aux collègues du département informatique pour l'ambiance de travail malgré les conditions d'exercice pas toujours… optimales.

Merci à l'université, qui a inflaté de manière inattendue la durée nécessaire pour pouvoir s'inscire en HDR, me donnant une excuse commode (mais fondée !) pour justifier ma procrastination.

Merci Aurélie pour ta relecture scrupuleuse.

Je ne remercie pas ChatGPT, ni les autres alternatives, puisqu'il n'a pas été utilisé pour la rédaction de ce document [1].

1. J'ai bien tenté de l'utiliser pour trouver un titre, celui-ci étant dramatiquement plat, mais force est de constater que les résultats n'étaient pas probants. Vous avez échappé à « *Naviguer dans l'océan des données : des méthodes d'indexation funky pour l'analyse de séquences sans alignement* » ou encore à « *Les super-pouvoirs de l'indexation : comment la bioinformatique utilise des méthodes sans alignement pour déchiffrer les séquences* ».

Résumé

Alors que les technologies de séquençage de l'ADN et de l'ARN se démocratisent, les données produites par les laboratoires se démultiplient. Les besoins de méthodes capables de traiter ces données, sans nécessiter le recours à des infrastructures de calcul gigantesques, est alors criant.

Mes recherches, comme maître de conférences dans l'équipe de bioinformatique Bonsai depuis 2010, ont eu pour but d'offrir la capacité aux équipes intéressées de tirer parti, à moindre coût, de ces données produites. Deux axes principaux de recherche correspondent à mes travaux sur la période afin de remplir cet objectif.

L'un de ces axes consiste à proposer des méthodes de comparaison qui évitent de recourir à de coûteuses étapes d'alignement, il s'agit des méthodes sans alignement. Avec mes collègues, j'ai proposé des méthodes d'analyse sans alignement, efficaces en temps et en espace qui fournissent des résultats pertinents pour les recherches ou les soins en oncologie. L'un de ces projets, Vidjil, est un logiciel désormais utilisé par des dizaines d'hôpitaux autour du monde dont certains financent deux ingénieurs qui continuent à maintenir et améliorer le logiciel.

L'autre axe correspond à l'indexation des données de séquençage avec l'objectif de faciliter leur exploitation, voire leur réexploitation. L'indexation de données est une étape indispensable afin d'obtenir rapidement des informations recherchées. Néanmoins, le type d'information qui peut être recherché dépend de la nature de l'indexation. Les méthodes présentées sont de différentes natures : certaines visent à offrir des recherches plus souples, tandis que d'autres ont pour objectif de favoriser la réexploitation de données existantes.

Enfin, j'aborde également succinctement mes activités de diffusion de la culture scientifique auprès du grand public ou dans le cadre de formations de journalisme.

Table des matières

Table des figures

Chapitre 1

Introduction

Les données produites par les séquenceurs d'ADN à haut débit ont suscité un intense effort de recherche, afin de concevoir des méthodes capables de faire face aux volumes disponibles croissants. Dans la communauté, il est de coutume d'illustrer cette évolution par la courbe de la figure 1.1, montrant l'augmentation du volume de données dans les banques de données publiques, comme SRA [1] qui donne accès à ces données de séquençage.

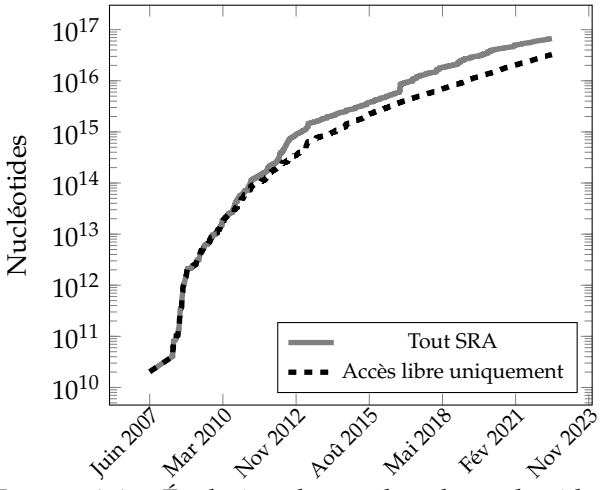

FIGURE 1.1 – Évolution du nombre de nucléotides disponibles dans SRA.

À la place, opérons un pas de côté en comparant l'arrivée des données de séquençage avec un projet pharaonique de la recherche en physique des particules. La construction du grand collisionneur de hadrons (LHC), un accélérateur de particules de 27 km de circonférence situé à la frontière franco-helvétique, a coûté près de 4,5 milliards de francs suisses [2], soit un peu moins de 4 milliards d'euros. Le projet génome humain (*Human genome project*), démarré à la fin du XX^{e} et achevé au début du XXI^{e}, qui visait à établir une séquence de référence du génome humain, a quant à lui, coûté environ 3 milliards de dollars [3]. Ces investissements ont en partie permis le développement des séquenceurs à haut débit. Les coûts initiaux des deux projets sont donc du même ordre de grandeur.

Maintenant en opération, le LHC produit environ 1 Po de données par jour. Le séquenceur NovaSeq 6000 d'Illumina, l'un des plus productifs, qui génère environ 1,8 To de données par jour a été vendu a plus de 1 800 exemplaires [4], ce qui donne un potentiel de plus de 3 Po de données par jour, qu'on peut ramener par prudence à 1 Po. Sans même compter les données produites par tous les autres séquenceurs, le volume de données produit par le LHC semble donc comparable, voire inférieur, à celui produit par les séquenceurs à haut débit dans le monde. Le but de cette comparaison, qui a évidemment ses limites, est d'illustrer l'aspect massif du séquençage à haut débit, en le comparant à un projet titanesque — et bien plus connu du grand public — comme le LHC. L'aspect hautement distribué du séquençage à haut débit, autour du monde, peut nous faire perdre de vue l'ampleur des données produites, dont SRA n'est que la tête d'épingle, et donc, l'enjeu méthodologique qui se pose à notre communauté.

Les fragments de séquences générés par les séquenceurs à haut débit, que j'appellerai *reads* dans la suite, constituent ce volume de données gigantesque auquel faire face. La production de ces séquences vise à remplir des objectifs divers : l'assemblage de nouveaux génomes pour établir les séquences de référence pour de nouvelles espèces ; l'étude d'échantillons environnementaux afin de déterminer les communautés microbiennes qui les composent ; une meilleure

1. *Sequence Read Archive* (*https://www.ncbi.nlm.nih.gov/sra*)
2. https://home.cern/resources/faqs/facts-and-figures-about-lhc
3. https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost
4. https://www.genomeweb.com/business-news/illumina-seeing-strong-novaseq-x-demand-reports-preliminary-q4-revenues-down-10

compréhension des relations phylogénétiques entre espèces, en comparant les séquences et en reconstruisant l'histoire évolutive ; le suivi d'épidémies, comme avec le Covid-19 ; la caractérisation des structures d'ARN, afin de mieux identifier leurs rôles ; la médecine personnalisée, en identifiant des mutations génétiques susceptibles d'orienter le traitement, etc.

1.1 Comparaison de séquences sans alignement

Dans de nombreux cas, l'analyse des données de séquençage débute, après une éventuelle étape de pré-filtrage, par le *mapping* des reads sur une séquence de référence, qui consiste à identifier la probable position d'origine de chaque read dans cette séquence. Or, connaître la position exacte de centaines de milliards de nucléotides séquencés n'est en fait pas un prérequis indispensable pour tirer des informations pertinentes de ces données. D'autant plus que la programmation dynamique, utilisée pour cette étape de mapping, n'est pas garantie d'exactitude. La garantie de trouver la solution optimale avec un algorithme de programmation dynamique est parfois une garantie rassurante. Pour autant, l'optimalité est celle d'une fonction de score, choisie parfois un peu arbitrairement. De plus, la programmation dynamique a ses propres limites : elle n'est adaptée que pour comparer des séquences colinéairement similaires. Dans le cas de variations structurales, elle devient moins pertinente, d'autant que la programmation dynamique sur-estime la similarité : des séquences générées pseudo-aléatoirement s'aligneront avec des taux d'identité non nuls. De plus, un algorithme de programmation dynamique ne pourra pas prendre en compte des mutations « inutiles » (par exemple, un nucléotide qui a été supprimé puis ré-inséré) (Ren *et al.*, 2018). Enfin, la programmation dynamique est un algorithme quadratique en temps, ce qui en fait une tâche très lourde lorsqu'elle est appliquée à des centaines de milliards de nucléotides, même si des optimisations ou des heuristiques peuvent améliorer cette complexité.

Afin de dépasser certaines de ces limites, des approches sans alignement sont développées depuis longtemps, avant même l'avènement du séquençage à haut débit (Vinga et Almeida, 2003). Ces méthodes, ainsi que celles développées ultérieurement pour des données de séquençage à haut débit (Bonham-Carter *et al.*, 2014), portent dans un premier temps sur des mesures globales de similarité entre deux séquences ou entre deux jeux de données. Il s'agit alors de méthodes globales et non de méthodes visant à détecter des événements dans chaque read.

Pour autant, les approches sans alignement sont également pertinentes à des échelles plus fines. En effet, l'alignement n'étant pas une fin en soi, il est souvent utilisé comme préalable afin d'identifier des mutations ponctuelles, des épissages, des variants plus complexes, de distinguer des sous-ensembles de séquences semblables, etc. Néanmoins, certaines de ces tâches ne nécessitent pas forcément d'alignement. Nous l'avons montré avec CRAC, publié en 2013 (voir section 2.1 page 6).

La quantification d'ARN a également connu un renouveau avec l'introduction d'approches sans alignement, plus rapide de plusieurs ordres de grandeur et donnant des résultats de bonne qualité (Zhang *et al.*, 2017). Ces approches sont de plus en plus sollicitées afin de traiter en un temps raisonnable des quantités de données croissantes. La figure 1.2 illustre l'intérêt pour les méthodes sans alignement. Au-delà des chiffres absolus (il est possible de publier une méthode sans alignement sans le mentionner dans le titre ou le résumé… voire sans en avoir conscience), il est intéressant de noter l'engouement pour ces approches avec le développement du séquençage à haut débit.

Nombre d'articles « alignment-free » publiés par an

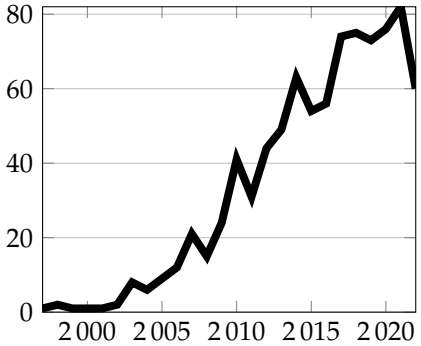

FIGURE 1.2 – Articles publiés dans la littérature et indexés par *Web of Science* ayant « alignment-free » dans le titre ou le résumé (articles autour de la physique, des communications sans fil retirés).

Éviter de recourir à une étape d'alignement, ou la limiter à un sous-ensemble de données le plus restreint possible, a été une ligne directrice que j'ai suivie lors de mes recherches, celles-ci sont présentées dans le chapitre 2 (page 5). Ces recherches ont débuté avec le logiciel CRAC, évoqué précédemment, que j'aborde rapidement dans la première partie. Ensuite, j'ai contribué avec Mathieu Giraud au logiciel Vidjil, dédié à la recherche de recombinaisons propres aux lymphocytes. Ce logiciel, réalisé en collaboration avec le CHU de Lille, s'appuie également sur une heuristique sans alignement, ce qui le rend particulièrement rapide. Enfin, j'ai récemment introduit un nouveau logiciel, FiLT3r, dans le cadre d'une autre collaboration avec le CHU de Lille, afin de détecter des duplications en tandem. Là encore, le choix d'une approche sans alignement le rend très efficace.

1.2 Indexation de séquences nucléiques

La constitution d'un index est la clé pour accéder rapidement à des informations d'intérêt dans un grand jeu de données, le plus souvent statique. Plus le volume de données à interroger est important, plus l'utilisation d'un index est pertinente. Sans être spécialiste, la question de l'indexation pourrait sembler résolue. En effet, des structures d'indexation pour le texte existent depuis plus d'un demi-siècle (Weiner, 1973) ; des moteurs de recherche internet ont la capacité d'indexer des dizaines de milliards de pages webs et de répondre en une fraction de seconde à n'importe quelle requête. Pourtant, ces moteurs de recherche sont très limités et ne peuvent répondre qu'à des requêtes très contraintes. Ils sont incapables de répondre à des requêtes comme : quelles pages webs contiennent la chaine *nformati* (qu'elle apparaisse dans *informatique* ou dans *information*) ? Quelles pages webs contiennent la chaine *anticonstitutionnel* à trois erreurs près (donc y compris si la page web contient par exemple *antzcontqtutionnel*) ? D'autre part, ces moteurs de recherche sur internet jouent sur un terrain qui présente des facilités : celui de la langue naturelle, composée de phrases, de mots, qui peuvent eux-mêmes être normalisés.

On notera que la chaine *antzcontqtutionnel* apparaît bien sur ma page web. En revanche un moteur de recherche n'est pour autant pas capable de trouver une occurrence approchée d'*anticonstitutionnel* sur ma page.

En bioinformatique, la recherche sur l'indexation reste très active tant les problématiques sont nombreuses. Les questions auxquelles doivent répondre de telles structures sont pourtant très basiques et se résument à trois types de requêtes, de la plus simple à la plus riche :

existence tel élément est-il indexé ?

comptage combien de fois tel élément a-t-il été indexé ?

localisation à quel(s) endroit(s) cet élément est-il présent dans mes données indexées ?

La difficulté du problème dépend des données qui sont indexées et, généralement du volume qu'elles constituent. Aussi, indexer un génome comme celui du génome humain (d'environ trois milliards de nucléotides) ne pose plus de problème, grâce d'une part à l'augmentation des ressources de calcul mais surtout grâce à la mise au point du FM-index (Ferragina et Manzini, 2000), reposant sur la transformée de Burrows-Wheeler (Burrows et Wheeler, 1994), qui utilise entre 1 et 2 Go pour indexer un génome humain, selon les implantations et les paramètres utilisés. En revanche, indexer des centaines de milliers de génomes d'une même espèce, présentant donc une redondance extrême, reste un sujet de recherche contemporain.

Le sujet peut encore être complexifié en ajoutant des requêtes plus sophistiquées : plutôt que de se poser la question de l'existence, du comptage ou de la localisation d'un élément, elle peut également être posée pour des éléments qui ressemblent à l'élément recherché. On cherche alors à connaître toutes les occurrences d'un élément en tolérant un nombre ou un pourcentage donné d'erreurs. Ce type de recherche approchée rend les solutions beaucoup moins rapides et donc beaucoup moins aptes à passer à l'échelle : en terme de complexité, les solutions sont souvent exponentielles dans le nombre de différences prises en compte.

À titre d'illustration, lorsqu'il s'agit d'indexer des jeux de données de séquençage bruts et qu'on désire autoriser un nombre défini d'erreurs entre la requête et ses occurrences, il n'y a, à l'heure actuelle et à ma connaissance, pas de solution exacte garantissant de trouver toutes les occurrences.

Sur cette thématique de l'indexation, j'ai co-encadré deux thèses et un post-doctorat, dont les thématiques sont plus détaillées dans le chapitre 3 page 27.

1.3 Des conséquences de nos travaux

Dans ce manuscrit, pour chacun des travaux que je présenterai, je ne rentrerai pas dans le détail des résultats que nous avons obtenus, ni ne discuterai de l'influence de tel paramètre sur la sensibilité, la spécificité ou l'efficacité de la méthode. À la place, je préfère donner une vision un peu plus globale des résultats de ces méthodes, les résultats détaillés se trouvant de toute façon dans les articles publiés. Dans ce but de prise de distance, une section *Impacts des recherches* résumera les conséquences qu'ont eues ces recherches auxquelles j'ai contribué et que j'ai co-encadrées.

Pourquoi présenter une telle section ? La recherche doit-elle nécessairement avoir des impacts, *a fortiori* à court ou moyen terme, puisque je n'ai pas le recul nécessaire pour en donner à long terme ? Évidemment que non, toutes les contributions que j'évoque ne donnent d'ailleurs pas lieu à des impacts de même nature. Certains peuvent être plutôt théoriques, d'autres plus concrets. En cohérence avec la déclaration DORA [5], notamment signée par le CNRS, je ne fournis pas d'évaluation bibliométrique quantitative, mais je mets plutôt l'accent sur des aspects qualitatifs.

Mes recherches sont, pour la quasi-totalité, menées en collaboration avec d'autres équipes en biologie, en santé ou en bioinformatique. Les résultats de nos recherches, proposer des méthodes adaptées à la question posée, sont importants pour que nos collègues puissent également avancer dans les leurs. Aussi, dans un tel contexte, des impacts à relativement court terme peuvent être attendus. D'autre part, lorsque nous publions un logiciel, c'est dans l'espoir qu'il soit utilisé au-delà de nos collègues proches. De même, lorsque nous publions un article, c'est aussi pour que des collègues le lisent et puissent s'appuyer dessus.

La raison d'une telle section tient aussi au retour d'expérience que je juge utile de faire. Quelles sont, selon moi, les raisons qui ont pu conduire au succès différencié de mes contributions ? Comme le disent Douglas *et al.* (2011), nous devons nous intéresser aux facteurs socio-culturels qui influent sur l'utilisation de nos logiciels. Il n'y a en effet pas que les qualités intrinsèques d'un algorithme ou d'une implantation qui participent à son utilisation mais également bien d'autres aspects comme son accessibilité, son utilité et sa portabilité, selon les critères définis par Douglas *et al.* (2011) (qu'ils appliquent plutôt aux bases de données en bioinformatique) :

accessibilité : l'ouverture du logiciel, la compréhension mutuelle de la problématique et de la réponse apportée, la réputation de l'équipe de recherche ;

utilité : la confiance dans les résultats obtenus et la maintenance, l'intégration de la solution dans les pratiques de recherche, l'interface d'utilisation ;

portabilité : l'utilisation de formats de donnée standards, la maintenance de long terme, l'application à d'autres contextes.

À partir de la grille d'analyse de Douglas *et al.* (2011), j'identifierai les critères remplis dans les méthodes auxquelles j'ai contribué et que je vais passer en revue dans la suite de ce document.

Enfin, dans un dernier chapitre j'évoquerai la diffusion de la culture scientifique, en décrivant quelques activités que j'ai entreprises et les raisons pour lesquelles je les ai initiées (chapitre 4 page 43).

5. https://sfdora.org/read/

Chapitre 2

Méthodes sans alignement pour l'oncogénétique

Bien que mon sujet de doctorat [1] était plutôt orienté algorithmique du texte que bioinformatique, j'ai commencé à collaborer en parallèle de ma thèse avec des équipes de Montpellier (au LIRMM (Laboratoire d'informatique, de robotique et de microélectronique de Montpellier) et à l'IRMB (*Institute for Regenerative Medicine and Biotherapy*)) sur une problématique bioinformatique d'analyse de données de séquençage d'ARN, notamment dans le cadre de cancers du sang.

À la fois par les hasards que ne manquent pas d'émailler une carrière, mais également en conséquence de collaborations pré-existantes, nombreuses de mes contributions en bioinformatique ont continué à porter sur ces thématiques : celles de la génétique des cancers (oncogénétique) et même, plus précisément, la génétique des cancers du sang.

Les cancers sont généralement définis comme une prolifération anormale et incontrôlée de cellules. Cette perte de contrôle peut notamment être due à des mutations dans des gènes clés. La nature des mutations est diverse. Il peut à la fois s'agir de mutations ponctuelles (la substitution, l'insertion ou la délétion de quelques nucléotides) ou de variations structurales (par exemple des translocations entre chromosomes, qui peuvent donner lieu à des *transcrits de fusion* si le résultat de la translocation est transcrit en ARN).

De telles mutations ont été identifiées dès le milieu du XX^{e} siècle pour certains cancers (Pane *et al.*, 2002). Ces mutations peuvent alors jouer le rôle de marqueurs moléculaires qui offrent un moyen détourné d'identifier une maladie ou un facteur de risque (comme avec la translocation BCR-ABL pour les leucémies myéloïdes chroniques (Preudhomme *et al.*, 1999) ou les leucémies aiguës lymphoblastiques (Maurer *et al.*, 1991)). Ces marqueurs sont d'autant plus importants pour des cancers du sang, qu'ils ont cette particularité d'être diffus et, donc, de ne pas bénéficier de l'imagerie médicale afin de les diagnostiquer ou d'évaluer leur évolution.

L'avènement du séquençage à haut débit a permis des analyses à plus large spectre, en ciblant des centaines de gènes chez des dizaines de patients en un seul séquençage, ainsi que des analyses plus fines en recherchant par exemple des transcrits de fusion, sans *a priori* sur le type de fusions à rechercher. La quantité de données qui en découle n'est évidemment plus la même que lorsqu'il s'agissait de « simples » PCR quantitatives [2]. Il devient indispensable pour les laboratoires médicaux de recourir à des outils bioinformatiques afin d'en extraire les informations pertinentes.

D'un point de vue bioinformatique se pose la question de méthodes adaptées à ces besoins. En terme de résultat, pour des questions de soin évidentes, il est indispensable d'obtenir des informations au moins aussi bonnes qu'avec les méthodes plus anciennes. En terme de méthodologie, des questions se posent également : en bioinformatique la boîte à outils est riche de méthodes plus ou moins précises (et plus ou moins efficaces, les deux dimensions pouvant être inversement corrélée). Nous verrons que ces questions ne sont pas si anodines et peuvent avoir des conséquences sur l'adoption (ou non) d'une méthode dans un cadre clinique. Dans la

1. Qui portait sur la mise à jour des structures d'indexation compressées pour le texte.

2. La PCR quantitative est une technique de biologie moléculaire cherchant à quantifier le ratio d'une séquence ADN donnée dans un échantillon. C'est le type de technique qui est utilisée dans le désormais fameux « test PCR » cherchant à identifier la présence (et, en réalité, à la quantifier) de fragments de génomes du SARS-CoV-2.

suite, je présente trois méthodes d'analyse sans alignement pour des données de séquençage à haut débit de patients dans le cadre de l'oncohématologie.

2.1 Analyse de données de séquençage transcriptomique

Dans le cadre d'une collaboration commencée en 2009 avec Nicolas Philippe, alors doctorant au LIRMM, à Montpellier, sous la co-direction d'Éric Rivals (LIRMM) et de Thérèse Commes (IRMB), nous avons réfléchi à la manière d'identifier des événements biologiques pertinents (mutations ponctuelles, épissages, translocations) dans des reads de RNA-seq [3]. L'équipe de Thérèse Commes est spécialisée dans la biologie des ARN, en particulier dans le contexte de cancers hématologiques où ces transcrits de fusion peuvent servir de marqueurs pour le suivi de la maladie, voire de facteurs pronostiques de l'évolution du cancer (Chwalenia *et al.*, 2017).

À l'époque où la collaboration a débuté, ni Bowtie, ni BWA (outils depuis devenus incontournables pour localiser des reads sur un génome de référence) n'étaient publiés (Langmead *et al.*, 2009 ; Li et Durbin, 2009). L'outil phare était alors SOAP qui proposait une approche que l'on considérerait aujourd'hui comme rudimentaire. Les reads étaient recherchés soit avec au maximum deux substitutions soit avec une insertion ou délétion de 1 à 3 nucléotides (Li *et al.*, 2008). Le génome était indexé dans une table de hachage, ce qui nécessitait de l'ordre de 14 Go de mémoire pour le génome humain, soit bien au-delà des capacités des machines de bureau de l'époque.

Problème

> *Proposer une méthode rapide et plus économe en mémoire pour identifier des événements biologiques dans des données de séquençage transcriptomique (RNA-seq).*

L'étude de données transcriptomiques ajoute la difficulté d'avoir à traiter des données épissées, c'est-à-dire dans lesquelles des parties des gènes ont disparu (les introns). Dans un read, il faut donc pouvoir identifier des régions qui ne sont pas nécessairement contiguës sur le génome de référence.

La méthode que nous avons proposée reposait sur plusieurs idées :

1. indexer la référence en utilisant une transformée de Burrows-Wheeler afin d'économiser de l'espace mémoire, bien moins abondant à cette époque ;
2. découper chaque read en séquences chevauchantes de longueur k, des k-mers, afin d'être robuste à tout type de mutations (substitution, insertion ou délétion) ainsi qu'aux épissages ou fusions ;
3. prendre en compte le nombre d'occurrences de chaque k-mer provenant des reads pour distinguer les erreurs de séquençage ;
4. réaliser un micro-assemblage des reads afin de pouvoir identifier des événements biologiques qui surviennent aux extrémités d'un read.

L'intérêt de réaliser un micro-assemblage s'explique par l'utilisation de k-mers. En effet, pour identifier correctement un événement biologique (ou une erreur de séquençage), il est nécessaire d'avoir un k-mer avant et après l'événement correctement localisé sur le génome (voir figure 2.1 page ci-contre). Dans le cas d'un événement séquencé à l'extrémité d'un read, il n'est pas possible que des k-mers soient identifiés avant et après celui-ci, rendant impossible sa détection. Ce problème était d'autant plus saillant que les reads étaient particulièrement courts en 2009 (et pouvaient souvent faire moins de 100 nucléotides (nt)).

Afin de pouvoir à la fois connaître le nombre d'occurrences de k-mers et naviguer dans les reads pour trouver ceux qui chevauchent celui qu'on aimerait étendre, nous avions besoin d'une structure d'indexation de reads idoine. Nous avons conçu une structure d'indexation pour des reads qui permette à la fois de connaître les occurrences de k-mers de reads, mais également de trouver tous les reads qui comportent un k-mer donné. Cette structure d'indexation, appelée Gk-arrays (Philippe *et al.*, 2011), s'appuie sur une table des suffixes modifiée, une table inverse et une table de comptage.

3. Le RNA-seq est le séquençage à haut débit des ARN présents dans un échantillon, c'est-à-dire des séquences transcrites d'un ensemble de cellules. L'analyse de ces données permet de connaître les gènes qui sont exprimés et le niveau de ces expressions.

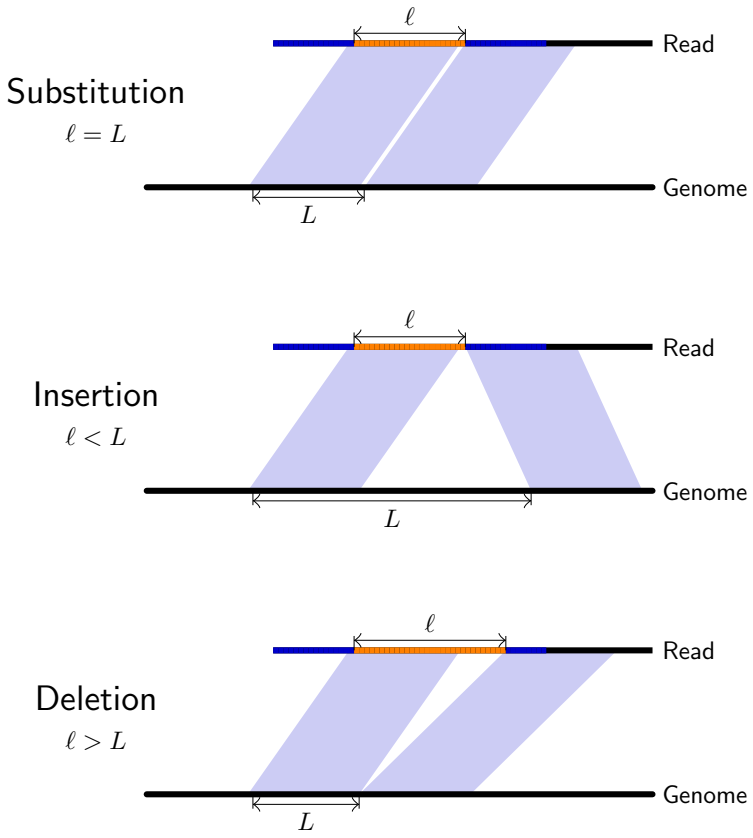

FIGURE 2.1 – En bleu foncé, sont représentées les positions de début de k-mers dans les reads qui sont localisés de manière unique sur le génome, en orange sont représentées les positions de début de k-mers qui ne sont pas localisées sur le génome, en bleu clair sont schématisées les localisations sur le génome du k-mer qui précède et qui suit la rupture de localisation. La différence entre l'écart de positions de ces k-mers sur le read et sur le génome permet de déduire le type d'événement biologique en cause, sans réaliser d'alignement (figure issue de Philippe *et al.* (2013)).

La méthode de localisation des reads et de détection d'événements biologiques a été mise en œuvre dans un logiciel libre appelé CRAC, et publiée ensuite (Philippe *et al.*, 2013). Néanmoins, l'idée de micro-assemblage des reads a été abandonnée faute de temps pour la développer.

2.1.1 Impacts des recherches

Ce projet est d'une certaine manière assez emblématique de la suite de mon parcours en bioinformatique. C'est d'une part un projet assez appliqué puisqu'il consiste à produire un logiciel pouvant être utilisé par toute équipe de bioinformatique intéressée. D'autre part, CRAC s'appuie sur des fondements théoriques qui permettent de proposer un logiciel efficace en temps et en mémoire. Enfin, la concrétisation même des idées sous-jacentes à CRAC a nécessité de revenir sur des considérations théoriques et de proposer une nouvelle structure d'indexation. Cela illustre — si besoin était — la manière dont recherche théorique et appliquée peuvent être particulièrement intriquées, si tant est qu'on arrive à les distinguer, y compris au sein d'un même projet. Cela montre également que le découpage choisi au sein de ce manuscrit est forcément en partie artificiel, même s'il a aussi sa cohérence.

Plus fondamentalement, la réalisation du logiciel CRAC a été très riche d'enseignements. La collaboration avec un laboratoire de biologie est primordiale afin de proposer un logiciel qui puisse répondre à des questions pertinentes, plutôt qu'à des mirages. Revenir aux données, comprendre les résultats produits par le logiciel sur des séquences caractéristiques a été primordial afin d'améliorer le logiciel et pour combler le fossé qui sépare les belles idées

théoriques et leur confrontation à la dure réalité des suites de nucléotides séquencées. Vouloir produire un logiciel utilisable par toute équipe de bioinformatique [4] requiert des efforts qui sont probablement incompatibles avec une éventuelle volonté de publier à un rythme effréné (les premiers développements ont eu lieu en 2009 et le papier a été publié en 2013) mais ils favorisent sans doute une adoption plus large du logiciel. Pour autant, afin qu'il soit utilisé dans le long terme, un logiciel doit non seulement être maintenu mais il doit également évoluer avec les avancées méthodologiques, ce qui demande un investissement conséquent qui n'a pas été possible sur ce logiciel.

Plus concrètement, le logiciel a été bien accueilli (Melton, 2013) et a été utilisé dans divers laboratoires à travers le monde avec lesquels nous n'avions aucun lien. Le développement autour de CRAC a continué, surtout à Montpellier, avec des étapes de post-traitement afin de raffiner la recherche de gènes ou transcrits de fusion et d'intégrer des informations obtenues à partir des annotations. Il a aussi donné lieu à la création d'une startup (SeqOne) par Nicolas Philippe, depuis devenue une PME, même si au final l'entreprise n'utilise pas CRAC en interne (leurs applications portant peu sur du RNA-seq).

2.2 Étude des recombinaisons V(D)J

Fin 2010, Martin Figeac, qui était ingénieur de recherche à l'institut de recherche en cancérologie de Lille (IRCL), a contacté notre équipe de bioinformatique Bonsai (que je venais de rejoindre en tant que maître de conférences) afin d'initier une collaboration pour étudier des données de séquençage à haut débit portant sur les recombinaisons V(D)J. Nous avons répondu à cette demande, avec Mathieu Giraud, qui était CR CNRS dans l'équipe à cette époque.

Les recombinaisons V(D)J sont un mécanisme de recombinaison de l'ADN survenant dans les lymphoblastes, les lymphocytes immatures. Ce mécanisme, mis en évidence par Tonegawa (1983), est à l'origine de la production d'anticorps ou de récepteurs des lymphocytes T, impliqués dans la réponse immunitaire adaptative (Murphy et Weaver, 2016).

Au-delà de l'intérêt évident pour l'immunologie, l'étude de ces recombinaisons a aussi un grand intérêt pour le suivi, voire le pronostic, de cancers du sang. En effet, une recombinaison V(D)J peut être vue comme un processus aléatoire qui va engendrer un segment d'ADN hautement spécifique de la cellule dans laquelle elle s'est produite. Ainsi, la recombinaison V(D)J peut servir de marqueur de cette cellule, et de toutes ses descendantes, ce qui est très utile quand il s'agit de suivre l'évolution d'un cancer au cours du temps. En effet, si un lymphoblaste ou un lymphocyte devient cancéreux, toutes ses cellules descendantes partageront la même recombinaison V(D)J [5], on parle alors de *clone*. Une fois la recombinaison V(D)J de la population de cellules cancéreuses identifiée, il suffit ensuite de suivre, avec une PCR quantitative, sa concentration par rapport aux autres cellules, pour connaître l'évolution de la maladie. Il s'agit de la quantification de la maladie résiduelle.

Pour initier un tel projet, comprendre plus précisément la biologie des recombinaisons V(D)J est indispensable.

2.2.1 Biologie des recombinaisons V(D)J

Tous les vertébrés à mâchoire disposent d'un ensemble de gènes V (*Variable*), D (*Diversity*) et J (*Joining*) (Hsu, 2009). Lors de la maturation des lymphocytes, seul un exemplaire de chaque catégorie est aléatoirement sélectionné et ils sont juxtaposés sur le génome, à travers un mécanisme de recombinaison, de telle manière qu'une seule protéine sera produite à partir de la recombinaison de trois gènes (Murphy et Weaver, 2016), voir figure 2.2 page suivante. Ce mécanisme de recombinaison VDJ est à l'origine, chez l'être humain, des chaines lourdes des immunogobulines (IGH, pour *ImmunoGlobulin Heavy*) et des chaines β et δ des récepteurs des lymphocytes T (TRB et TRD, pour *T-cell Receptor Beta/Delta*). Un mécanisme similaire de recombinaison (la recombinaison VJ) existe pour les chaines légères des immunoglobulines

4. Ce que certains qualifieraient de TRL (*Technology Readiness Level*) 7 ou plus.

5. Par souci de simplicité, on considèrera que la recombinaison V(D)J d'une population de cellules cancéreuses reste la même. Ce n'est que partiellement vrai puisque des mutations, voire des changements de gènes V peuvent survenir. Sans même parler des cas où les tumeurs sont hétéroclites et où des populations initialement minoritaires se révèlent être résistantes au traitement, leur conférant un avantage sélectif évident.

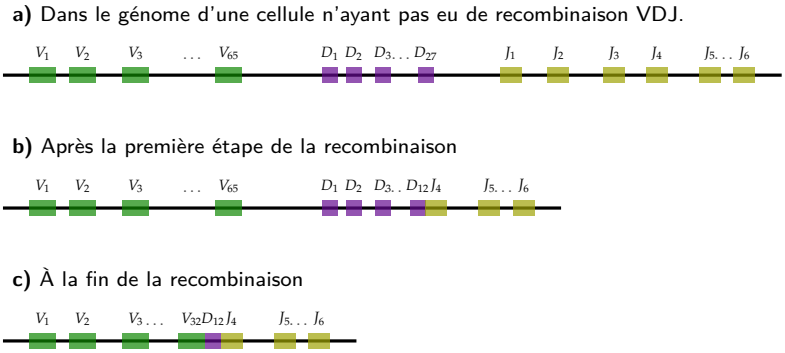

FIGURE 2.2 – Les étapes d'une recombinaison VDJ, qui recombinent un ensemble de gènes V (*Variable*), D (*Diversity*) et J (*Joining*). a) Dans un génome n'ayant pas eu de recombinaison VDJ (ce qui est le cas de toutes les cellules sauf les lymphoblastes ayant entamé leur maturation), les gènes V, D et J ne sont pas recombinés. b) La recombinaison VDJ débute par la recombinaison d'un gène D avec un gène J. c) Ensuite le même processus a lieu entre un gène V et les gènes D-J déjà recombinés. C'est la fin de la recombinaison.

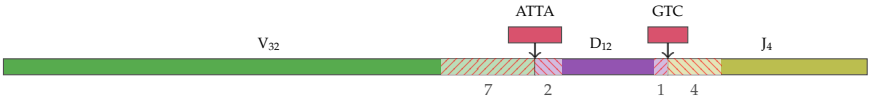

FIGURE 2.3 – Détail d'une recombinaison VDJ. De la diversité est ajoutée lors de la recombinaison : des nucléotides sont supprimés au niveau de la jonction entre les gènes et d'autres nucléotides peuvent être aléatoirement insérés. Ici 7 nucléotides sont supprimés à la fin du gène V recombiné, 2 nucléotides sont supprimés au début et 1 à la fin du gène D recombiné et 4 nucléotides sont supprimés au début du gène J recombiné. De plus, les nucléotides ATTA sont ajoutés entre le gène V et D, les nucléotides GTC sont ajoutés entre les gènes D et J.

(IGL ou IGK) et les chaines α et γ des récepteurs des lymphocytes T (TRA et TRG), dans lesquelles il n'y a pas de gène D. La recombinaison VJ s'effectue donc uniquement entre un gène V et un gène J.

Chez l'être humain, pour la chaine lourde des immunoglobulines, 194 gènes V, 37 gènes D et 9 gènes J sont recensés par la base de données de références d'IMGT®, IMGT/GENE-DB (Giudicelli *et al.*, 2005). D'un point de vue combinatoire, près de 65 000 recombinaisons différentes sont possibles. Ce nombre peut sembler élevé, il est en fait ridiculement faible au regard de la diversité des épitopes [6] à reconnaître.

En réalité, la recombinaison VDJ ne consiste pas uniquement en une recombinaison de trois ensembles de gènes, mais également en un enrichissement en diversité. Au moment de la recombinaison d'un gène D et d'un gène J, ainsi que lors de celle d'un gène V avec les gènes D-J déjà recombinés, des nucléotides sont retirés au niveau de la jonction entre les gènes recombinés. De plus, des nucléotides sont aléatoirement ajoutés au niveau de ces jonctions (voir figure 2.3). Au final, la recombinaison V(D)J est longue de 250 à 550 nucléotides, avec une région hautement spécifique (de la fin du V au début du J) qui fait, au maximum, quelques dizaines de nucléotides. Cette région, délimitée par une cystéine à la fin du V et une tryptophane au début du J, s'appelle le CDR3 (*Complementary-Determining Region* 3). C'est la région la plus importante d'un point de vue immunologique car c'est elle qui sera le plus en contact avec l'épitope.

D'un point de vue combinatoire, cette diversité de recombinaisons VDJ ou VJ (ensuite notées V(D)J), constitue pour les lymphocytes B un répertoire d'anticorps potentiels estimé à 10^{12} et de 10^{16} à 10^{18} en tenant compte des mutations somatiques (Briney *et al.*, 2019). Bien entendu, il s'agit d'un répertoire **théorique** et ne représente pas le nombre d'immunoglobulines différentes qu'un individu peut avoir à un instant donné (Rees, 2020). Néanmoins, cela donne une idée de l'immensité de la diversité des répertoires des recombinaisons V(D)J et cela aide à comprendre en quoi une recombinaison V(D)J peut servir à identifier de manière probable-

6. Les épitopes sont des fragments de protéines auxquels se lient les immunoglobulines (ou anticorps) et les récepteurs des lymphocytes T.

ment unique une population de cellules, par exemple cancéreuse. Pour autant, il faut prendre garde au fait que certaines recombinaisons sont retrouvées chez de nombreux individus différents, possiblement en raison de réponses convergentes à de mêmes épitopes (Briney *et al.*, 2019).

2.2.2 Historique des méthodes d'identification de recombinaisons V(D)J

L'obtention de séquences nucléiques de recombinaisons V(D)J n'a pas attendu le séquençage à haut débit mais a débuté avec le séquençage Sanger. Des outils bioinformatiques adaptés à l'analyse de quelques dizaines ou centaines de recombinaisons ont été développés à cette époque. Le premier, dès le début des années 1990, semble être DNAPLOT (Müller et Althaus), ensuite rendu utilisable plus simplement via une interface par IMGT (Giudicelli *et al.*, 1997), puis renommé en IMGT/V-QUEST. Ensuite, le logiciel Blast a été adapté à la question des recombinaisons V(D)J, dans une déclinaison appelée IgBlast à partir de l'an 2000[7], mais formellement publié bien après (Ye *et al.*, 2013). Pour le premier, DNAPLOT, la méthodologie n'est que peu décrite mais il s'agit *a priori* d'alignement d'une recombinaison vis-à-vis des bases de données de gènes V, D et J de référence.

Vont ensuite suivre JOINSOLVER (Souto-Carneiro *et al.*, 2004), SoDA (Volpe *et al.*, 2006), VDJSolver (Ohm-Laursen *et al.*, 2006), iHMMune-align (Gaëta *et al.*, 2007) et AB-origin (Wang *et al.*, 2008). Dans ces logiciels, les approches sont diverses (Blast, maximum de vraisemblance, HMM, simulations de Monte-Carlo, ...) mais le but reste le même : caractériser le mieux possible quelques recombinaisons V(D)J.

La logique va changer avec le développement du séquençage à haut débit. Alors qu'auparavant une seule séquence était obtenue par séquençage Sanger, à partir d'un ensemble de molécules *a priori* semblables et issues d'un même clone, désormais chaque molécule est séquencée et un même clone biologique peut être à l'origine de milliers, voire millions, de séquences produites par le séquenceur.

D'une part il devient important de rassembler les séquences qui proviennent probablement d'une même population clonale, on parlera de *clonotypes*. D'autre part, il est indispensable de proposer des méthodes capables de passer à l'échelle.

C'est dans ce contexte que nous avons commencé à travailler (Mathieu Giraud et moi-même) sur la question de l'identification de recombinaisons V(D)J dans des données de séquençage à haut débit, en collaboration avec le CHU de Lille qui avait sollicité notre expertise à ce sujet fin 2010.

Problème

> *À partir d'un ensemble de reads issus d'une expérience de séquençage à haut débit, proposer une méthode efficace en temps rassemblant ces reads par clonotype et caractérisant ces clonotypes par leur recombinaison V(D)J.*

2.2.3 Conception de notre méthode d'identification – Vidjil

Avant d'aborder la méthode que nous avons conçue pour détecter les clonotypes et identifier les recombinaisons V(D)J, je vais brièvement commencer par décrire la méthode que nous n'avons pas retenue. Il existera ainsi au moins un endroit où sera présent ce résultat négatif !

Inspirés par CRAC, nous avons imaginé une approche sans *a priori* dans laquelle nous pourrions découper les parties V, D et J d'une recombinaison ADN juste en comptant les occurrences des *k*-mers dans tous les reads. L'évolution de ce comptage le long du read devrait nous permettre d'identifier la zone hautement spécifique de la recombinaison V(D)J, qui contient les insertions et le gène D le cas échéant. En effet, les *k*-mers faisant partie de cette région devraient être de bien plus faible occurrence que les *k*-mers avoisinants qui ne se trouveraient que dans le gène V ou que dans le gène J. De même, comme il existe plus de gènes V que de gènes J, un *k*-mer provenant d'un gène V devrait avoir une plus faible occurrence dans le jeu de reads qu'un *k*-mer provenant d'un gène J. L'étude du nombre d'occurrences des *k*-mers dans un read comportant une recombinaison V(D)J devrait donc avoir le profil suivant : d'abord le gène V avec un nombre d'occurrences relativement élevé, puis une région assez courte, entre le gène V et le gène J, où le nombre d'occurrences s'effondre (la région contenant

7. On ne retrouve pas de trace du logiciel avant.

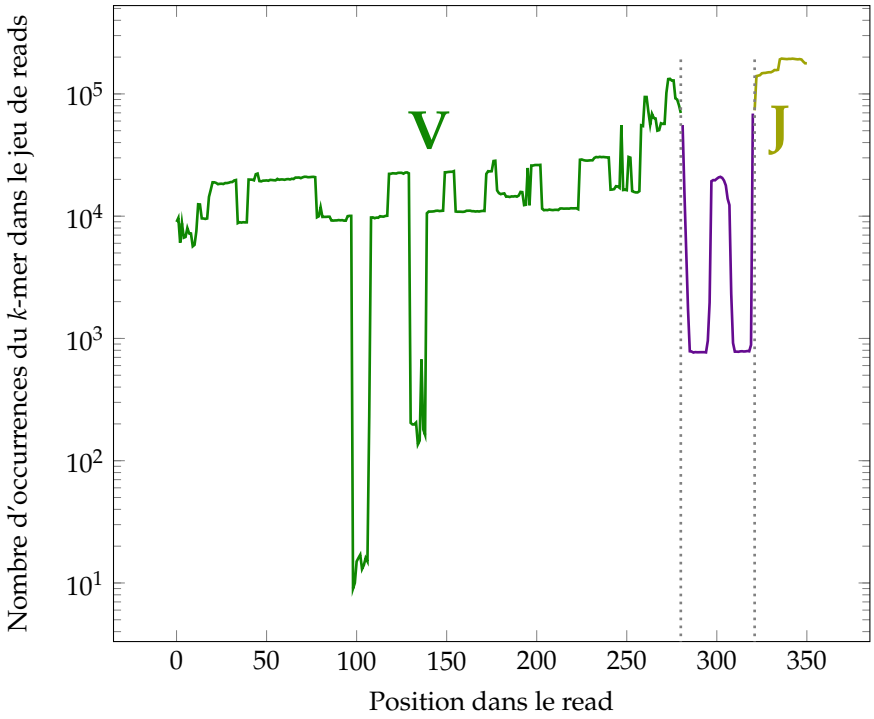

FIGURE 2.4 – Nombre d'occurrences des k-mers d'un read contenant une recombinaison V(D)J parmi la collection complète de k-mers des reads. L'évolution de la courbe permet de constater une chute soudaine du nombre d'occurrences de k-mers vers la fin de la séquence pendant une quarantaine de nucléotides, légèrement interrompue par un bref sursaut. Cette évolution permet d'identifier relativement précisément la fin du gène V, et le début du gène J, dont le nombre d'occurrences est très élevé car il existe peu de gènes J, il y a donc une faible diversité. Le petit sursaut entre le gène V et le gène J peut être dû à la présence d'un gène D. Les k-mers présents à la jonction V-D et D-J sont très spécifiques de ce clonotype et donc relativement rares.

les jonctions, avec leurs insertions et délétions) puis une région correspondant au gène J où le nombre d'occurrences devient élevé. Ce travail a été mis en œuvre par un stagiaire de M2, David Chatel, encadré par Mathieu Giraud et moi-même (voir figure 2.4). Malheureusement, les bornes de la fin du V et du début du J n'étaient pas détectées suffisamment finement avec une telle méthode, en particulier en présence d'un clone ultra-majoritaire, où tous les k-mers sont très fortement retrouvés.

2.2.3.1 Une approche qui constitue d'abord les clonotypes

Néanmoins, nous avons gardé un principe de cette approche : nous considérons qu'il n'est pas indispensable d'aligner tous les reads contre les gènes V, D et J de référence. En effet, en partant du postulat que nous pouvons arriver à reconstituer les clonotypes correctement, alors chaque clonotype correspond à un ensemble de reads qui contiennent une même recombinaison V(D)J (éventuellement à quelques menues différences près). Il suffit alors de désigner un read comme bon représentant de cet ensemble et d'identifier sa recombinaison V(D)J. Une telle approche a le gros avantage de limiter drastiquement les calculs, puisque dans les cas pathologiques que nous avons à traiter un même clonotype peut être constitué de dizaines de milliers voire de centaines de milliers de reads.

Il reste alors à identifier les clonotypes sans avoir à aligner tous les reads contre les séquences de référence. Pour cela nous allons nous contenter de détecter une « fenêtre » de taille fixe centrée entre la fin du V et le début du J. Tout read possédant la même séquence nucléique dans la fenêtre sera considéré comme faisant partie du même clonotype. L'approche requiert d'identifier la fin du V et le début du J dans le read. En découpant chaque read en graines chevauchantes (en k-mers, par exemple) et en recherchant ces graines parmi les gènes V ou J de référence, nous allons pouvoir identifier le début ou la fin (approximative) des gènes V et J.

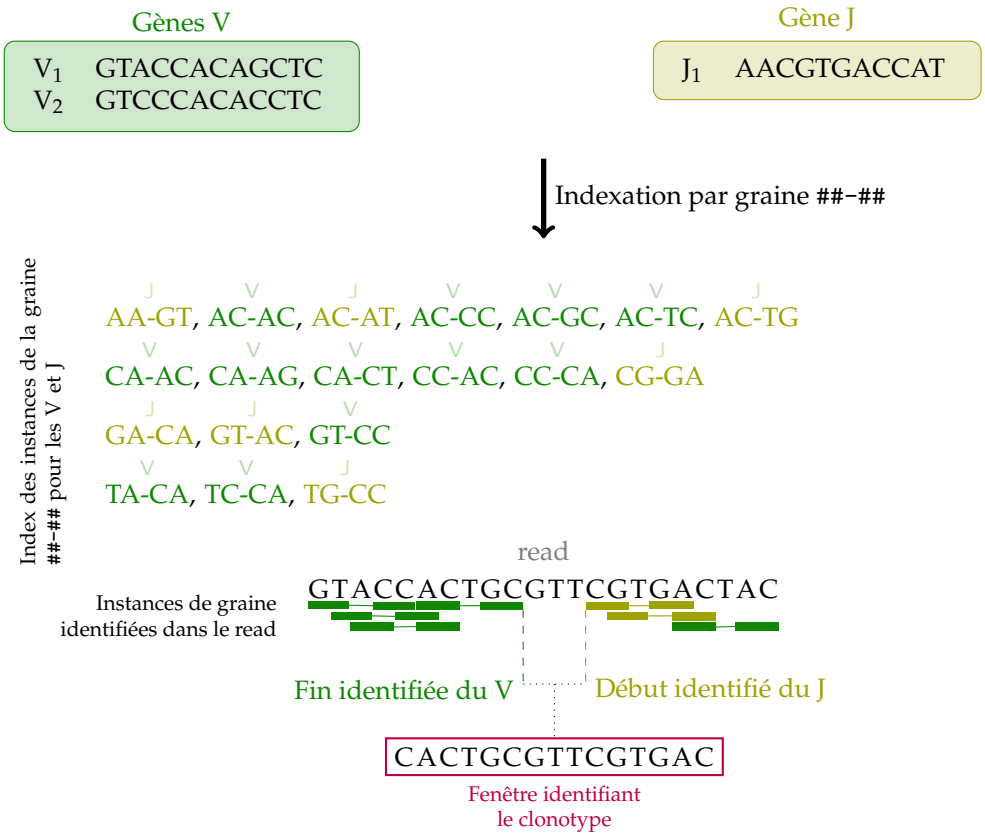

FIGURE 2.5 – Illustration de l'heuristique employée pour définir les clonotypes. Une ou plusieurs graines sont utilisées pour indexer le répertoire des gènes V et J connus. Chacune des instances de graine dans le read est recherchée dans l'index, afin de déterminer s'il s'agit d'une instance V ou J ou si elle n'appartient à aucun des deux. Par exemple pour GTACC, au début du read, l'instance de la graine ##-## est GT-CC, qui apparaît parmi les instances de graines V. La position du milieu de la fenêtre, de taille fixe, identifiant le clonotype, est définie comme le milieu entre la fin identifiée du V et le début identifié du J.

Une telle approximation suffira à définir nos clonotypes, qui seront identifiés par une fenêtre de w nucléotides centrée sur la position centrale entre la fin du V et le début du J (voir figure 2.5). Les graines employées peuvent aussi bien être des k-mers que des graines espacées. En pratique, nous utilisons des graines espacées, que l'on peut voir comme des k-mers dans lesquels certaines positions ne sont pas considérées (des jokers). Dans notre cas, nos graines ont un seul joker, il est au milieu. Ainsi, en un temps linéaire dans la taille du jeu de reads en entrée nous identifions les clonotypes de ce jeu de reads.

Cette approche, où on commence par constituer les clonotypes avant de procéder à l'alignement, est à rebours des autres méthodes, notamment IMGT/HighV-QUEST (Alamyar *et al.*, 2012), MiXCR (Bolotin *et al.*, 2015), IMSEQ (Kuchenbecker *et al.*, 2015), partis (Ralph et Matsen IV, 2016), IgRec (Shlemov *et al.*, 2016) pour citer les plus connues. En effet, l'approche classique que suivent ces autres méthodes est, dans un premier temps, d'aligner chaque read contre les séquences de référence, puis, une fois la recombinaison V(D)J bien caractérisée, de constituer les clonotypes selon différents critères (identité du CDR3 en acides aminés, prise en compte de mutations, etc.).

Notre constitution de clonotypes est une méthode sans alignement. Pour la mener à bien, une table de hachage suffit à identifier les instances de graines espacées qui sont présentes dans un gène V ou J. L'approche est donc linéaire[8], mais pour une seule chaine (de récepteurs) donnée. En effet, le répertoire des gènes V et J change pour chaque chaine et il faut donc renouveler l'opération pour chacune, notamment car des graines différentes peuvent être utilisées selon les chaines (des graines plus courtes sont utilisées pour des chaines ayant peu de

8. Si on suppose le temps d'accès aux tables de hachage constant.

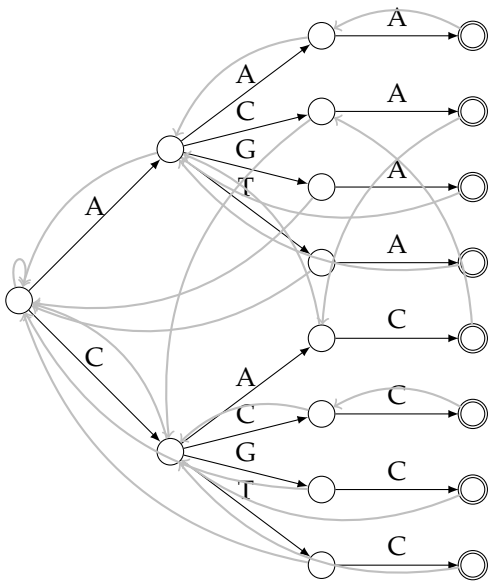

FIGURE 2.6 – Automate d'Aho-Corasick pour la graine espacée `#-#` sur la séquence `ACAC`. Les instances de la graine sont donc `A-A` et `C-C`, qui seront indexées comme {AAA, ACA, AGA, ATA, CAC, CCC, CGC, CTC}. Chaque flèche grisée correspond au résultat de la fonction d'échec, telle que définie par Aho et Corasick, pour l'état de départ.

gènes). Il s'agit de la méthode que nous avons initialement développée et publiée dans le logiciel Vidjil (Giraud *et al.*, 2014). Au final, notre complexité ne dépend pas que de la longueur de la séquence donnée en entrée mais également du nombre de chaines à analyser, ce qui est peu élégant en théorie, et plus coûteux en pratique.

Indexation multi-graines

Pour remédier à cet écueil, il est indispensable d'indexer les instances de graines des différentes chaines dans un même index. Une table de hachage n'est donc plus adaptée puisqu'il faudrait tester explicitement les différentes longueurs de graines possibles. À la place, nous avons besoin d'une solution pour trouver toutes les occurrences de graines (peu importe leurs tailles) dans un read. Cela correspond donc au problème de rechercher plusieurs motifs dans un texte, pour lequel il existe une solution dont le temps ne dépend pas du nombre ou de la longueur des motifs à rechercher : l'automate d'Alfred Aho et Margaret Corasick (1975).

En effet, chaque instance de graine peut être vue comme un motif à ajouter dans l'automate d'Aho-Corasick et le read est le texte dans lequel rechercher ces instances de graines. L'automate est donc construit une seule fois sur l'ensemble des gènes V et J de toutes les chaines. Les instances des graines espacées sont gérées de manière naïve : chaque trou de la graine est remplacé par chacun des 4 nucléotides possibles, le nombre d'instances de graines est donc exponentiel dans le nombre de trous de la graine espacée d'origine, mais en pratique les graines que nous utilisons n'ont qu'un seul trou (voir figure 2.6).

Pour chaque read, l'automate d'Aho-Corasick permet d'avoir l'ensemble des instances de graines qui ont été retrouvées, parmi l'ensemble des chaines existantes. Parmi les instances retrouvées, certaines probablement juste par chance, il reste à identifier de quelle chaine est le plus probablement issu notre read. Le nombre de hits ne suffit pas à identifier cela, puisque les graines peuvent être de tailles variables. À la place, nous utilisons un calcul de probabilités qui a pour but d'évaluer si le nombre de graines retrouvées diffère de ce qu'on aurait pu attendre par hasard. Nous considérons qu'une seule graine d'une chaine donnée a une probabilité α d'apparaître dans une séquence aléatoire à une position donnée, où α est le taux de chargement de l'index (c'est-à-dire le nombre d'instances de graines de cette taille indexées pour cette chaine par rapport au nombre total d'instances possibles de cette taille). Ainsi, nous calculons la probabilité d'avoir, par chance, au moins x instances de graines avec une probabilité α, dans une séquence de n k-mers, avec $\sum_{i=x}^{n} \binom{n}{i} \alpha^i (1-\alpha)^{n-i}$. Une fois la chaine identifiée, celle

ayant la probabilité minimale, il reste à déterminer la dernière instance pertinente de graine appartenant au gène V et la première pertinente appartenant au gène J afin d'identifier la « fenêtre » qui nous sert à définir un clonotype.

Atténuer l'influence des faux positifs

Loin de l'idéal théorique qu'on pourrait imaginer, les gènes V et J peuvent présenter des similitudes, plus que ce qui pourrait être attendu par hasard. Aussi, il est possible que, par malchance, une instance de graine J soit identifiée au milieu d'un V, ou inversement. Ce cas est illustré dans la figure 2.5 page 12 : la dernière instance de graine identifiée dans le J vient en fait d'un gène V. Afin de ne pas méprendre cette occurrence pour la fin du V, la fin identifiée du V est définie comme la position de fin de la dernière instance de graine V qui maximise le nombre d'instances de graines V à sa gauche et d'instances de graines J à sa droite. On définit de manière similaire la position de début du J. Dans la figure 2.5 page 12, à la position identifiée de fin du V, il y a quatre instances de graines V à gauche et deux instances de graine J à droite (soit un total de six). Cela maximise bien le nombre d'instances de graines, car si nous avions choisi la dernière instance de graine V, nous aurions eu cinq instances V à gauche et zéro instance J à droite.

Conséquence des faux négatifs

Les erreurs de séquençage ou des mutations non documentées dans les bases de référence peuvent induire des faux négatifs, c'est-à-dire des instances de graines non identifiées. La pire des situations correspond à une différence qui apparaîtrait dans ce qui aurait dû être la dernière instance de graine V ou la première instance de graine J. Dans cette situation, un faux négatif aura pour effet de légèrement décaler la fenêtre par rapport à la position qu'elle aurait dû avoir.

Nous atténuons cet effet en utilisant des graines espacées, dont le joker est au milieu. Ainsi, le décalage maximal induit par une différence mal placée sera au maximum de la moitié de la taille de la graine.

Dans le cas d'une erreur de séquençage à proximité de la fenêtre, le read ne fera pas partie du clonotype auquel il aurait dû appartenir et cela aura pour effet de minorer la taille absolue des clonotypes. La taille relative ne sera que peu affectée, à moins qu'un clonotype soit particulièrement touché par des erreurs de séquençage. Nos usagers hospitaliers étant intéressés par les clonotypes les plus abondants, ce point n'a pas été identifié comme critique.

Dans le cas d'une différence due à une mutation, le décalage de la fenêtre sera reproductible pour tous les clonotypes présentant la mutation. Le clonotype sera donc bien identifié. Le risque est que la fenêtre se retrouve intégralement dans le V ou dans le J (et non à cheval entre les deux), ce qui pourrait la rendre non spécifique. Néanmoins, puisque le décalage maximal induit par une telle différence est de la moitié de la taille de la graine, ce risque est improbable avec une taille de fenêtre suffisante.

2.2.3.2 Caractérisation de la recombinaison V(D)J d'un clonotype

Une fois les clonotypes constitués, il reste indispensable d'identifier la recombinaison V(D)J qui correspond au clonotype. En effet, la recombinaison V(D)J n'est pas seulement un moyen de distinguer les populations de lymphocytes, mais les gènes utilisés, leurs bornes de début ou de fin, le taux de mutations, sont des informations utiles pour le pronostic ou le suivi de la maladie.

Heuristique à base de k-mers pour l'identification de la séquence représentative d'un clonotype

À partir d'un clonotype, correspondant à m reads pour lesquels une fenêtre identique a été identifiée, nous calculons une séquence représentative de ces reads, ce qui est une heuristique pour éviter le calcul d'une séquence consensus qui serait trop coûteux. À la place, si le nombre de reads est élevé dans le clonotype, seul un échantillon des reads aléatoire, donc représentatif, est considéré. Ensuite, par définition, tous les reads partagent une même séquence

Fonction *representativeClonotype(R, w, k)*:
Entrées : R : reads (éventuellement échantillonnés) constituant le clonotype
Entrées : w : séquence de la fenêtre identifiant le clonotype
Entrées : k : taille des k-mers

$representative \leftarrow \varepsilon$;
$index \leftarrow countKmers(R, k)$;
pour chaque *read dans R* **faire**
  $seq \leftarrow plusLongueSequenceRepresentative(read, index, |R|, w, k)$;
  si $|seq| > |representative|$ **alors**
    $representative \leftarrow seq$;
  fin
fin
retourner *representative*;

Fonction *plusLongueSequenceRepresentative(read, index, w, nb, k)*:
Entrées : *read* : séquence d'un read
Entrées : *index* : comptage des k-mers sur l'ensemble des reads
Entrées : *nb* : nombre de reads pris en compte
Entrées : w : séquence de la fenêtre, considérée unique
Entrées : k : taille des k-mers

$pos \leftarrow read.\text{find}(w)$;
$debut \leftarrow pos - 1$;
tant que $debut \geq 0$ **et** $index[read[debut\,..\,debut + k - 1]] \geq \frac{nb}{2}$ **faire**
  $debut - -$;
fin
$debut + +$;
$fin \leftarrow pos + |w|$;
tant que $fin < |read|$ **et** $index[read[fin\,..\,fin + k - 1]] \geq \frac{nb}{2}$ **faire**
  $fin + +$;
fin
$fin - -$;
retourner $read[debut\,..\,fin]$;

Algorithme 1 : Algorithme de calcul d'une séquence représentative pour un ensemble de reads partageant une séquence w commune. On considère que la fonction *countKmers* existe par ailleurs et construit un index des k-mers de la collection de reads fournie en paramètre.

correspondant à la fenêtre. Il s'agit donc du point de départ au calcul de la séquence représentative et celle-ci est étendue autant que possible vers la droite ou la gauche, tant que le résultat est suffisamment consensuel (tant qu'on ajoute des k-mers qui sont vus dans au moins 50 % des reads [9]). La séquence représentative la plus longue est conservée (voir l'algorithme 1).

Par simplicité, nous considérons que la séquence représentative doit apparaître dans un read. L'hypothèse est moins forte qu'il n'y paraît. À 0,5 % d'erreur (le taux habituel pour un séquenceur MiSeq (Stoler et Nekrutenko, 2021)), un read de 500 nt a 8 % de chances d'être sans erreur. Même pour un clonotype de seulement 10 reads, la probabilité d'avoir un read sans erreur est de 57 % [10]. En pratique, les clonotypes les plus abondants, qui sont ceux investigués par les biologistes sur un échantillon diagnostic, sont généralement constitués de milliers de reads.

9. Ce n'est pas exactement ce qui est indiqué dans l'algorithme : le nombre d'occurrences du k-mer ajouté doit représenter au moins la moitié du nombre de reads du clonotype. S'il existe des k-mers fortement répétés, la condition pourrait être atteinte sans que la moitié des reads du clonotype présente le k-mer. Néanmoins les recombinaisons V(D)J sont peu répétées (elles sont codantes). L'approximation est donc raisonnable, à partir du moment où k est suffisamment élevé.

10. En pratique les reads sont plus courts et chevauchants, ce qui permet de corriger une partie des erreurs. Les hypothèses présentées ici sont donc plutôt défavorables, même si l'hypothèse sous-jacente d'indépendance des erreurs n'est pas totalement acquise.

Comparaison de la séquence représentative aux gènes de référence

Une fois la séquence représentative obtenue, ce qui nous intéresse est de connaître les caractéristiques de la recombinaison V(D)J qu'elle contient. Plutôt que d'aligner la séquence représentative obtenue directement contre tous les gènes V et J de la chaine considérée, nous utilisons à nouveau l'heuristique qui identifiait le clonotype, consistant à trouver des graines provenant des gènes V ou J. Néanmoins, cette fois nous ne nous intéressons pas uniquement à savoir si une graine du read provient d'un gène V ou d'un gène J, mais également de quel gène il s'agit. Cette opération permet d'identifier en temps linéaire les gènes V les plus probables (cette opération n'est pas réalisée pour les gènes J car ils sont beaucoup plus courts et moins nombreux, l'alignement sur ceux-ci est donc peu coûteux). Cela limite l'alignement par programmation dynamique à ces gènes (et à leurs différents allèles).

À l'heure actuelle, près de 200 gènes V différents sont recensés par IMGT, la base de référence, pour la chaine lourde des immunoglobulines (IGH). Au lieu d'aligner chaque séquence contre ces 200 gènes (et leurs allèles !), il suffit, dans le meilleur des cas, de n'aligner que contre un seul gène. En pratique, cette heuristique permet d'accélérer l'alignement sur les IGH (la chaine la plus diverse) par un facteur 20.

2.2.4 Vers une application utilisable en routine clinique

L'importance clinique des résultats produits par Vidjil (l'identification de séquences, servant comme marqueurs utilisés pour le suivi de leucémies) rend indispensable la mise en place de certaines pratiques pour assurer une qualité adéquate du logiciel.

Il est indispensable de s'assurer que les recombinaisons V(D)J identifiées par Vidjil sont correctement annotées, c'est-à-dire que les gènes V, D et J sont correctement positionnés sur la séquence. En effet, dans cette recombinaison de plusieurs centaines de nucléotides, seule une quinzaine de nucléotides consécutifs seront utilisés afin de concevoir la sonde qui servira à la quantification par PCR lors des suivis du patient. Cette séquence se doit d'être spécifique du patient et, plus précisément, du clonotype identifié afin de bien suivre l'évolution d'une population de cellules cancéreuses. Or, mal positionner les gènes V, D et J, c'est risquer de laisser penser à un·e biologiste qu'une séquence n'est pas présente dans un gène V, D ou J, alors qu'elle l'est. Si la sonde utilisée par le laboratoire correspond à une séquence d'un gène V, D ou J, il est évident que la quantification ne reflètera pas celle de la population cancéreuse. Nous avons mis à contribution les biologistes utilisant Vidjil : nous leur avons demandé de recenser des recombinaisons V(D)J et d'indiquer les annotations qui leur semblent pertinentes. Une demi-douzaine de collègues ont contribué, ce qui nous a initialement permis d'intégrer plus de 200 séquences réelles à un jeu de tests (Salson *et al.*, 2016) sur lequel les résultats de Vidjil sont systématiquement vérifiés, en plus de tests unitaires ou fonctionnels plus classiques. Désormais, ce sont plus de 500 séquences qui composent ce jeu de tests.

2.2.5 Mise en œuvre et résultats

La partie analyse de séquences a été codée en C++ dans un logiciel libre, désormais appelé Vidjil-algo. Dans notre article initial (Giraud *et al.*, 2014), nous montrons sur des données simulées que notre heuristique est autant en concordance avec des logiciels commençant par aligner chaque read sur les gènes de référence, que ces logiciels entre eux (IMGT/V-QUEST et IgBlast), y compris avec des taux de mutations élevés atteignant 9 %. Dans un article en préparation, nous montrons que notre nouvelle heuristique, utilisant l'automate d'Aho-Corasick, est jusqu'à cinq fois plus rapide que notre heuristique précédente. Elle peut traiter 2 millions de séquences en moins de 3 minutes, là où MiXCR, un logiciel très utilisé, y passe un peu moins de 3 heures, mais fournit des informations plus complètes.

D'autre part, afin d'assurer une adoption plus aisée du logiciel par les biologistes ou techniciens hospitaliers, et en lien avec la notion d'utilité développée par Douglas *et al.* (2011), nous avons entrepris de proposer une application web pour gérer les aspects serveur (upload de fichiers, lancement de jobs avec file d'attente, enregistrement des métadonnées, etc.) ainsi que client (visualisation interactive des résultats d'analyse de Vidjil). Cette application libre a été développée en Python et Javascript (voir figure 2.7 page suivante). Marc Duez, stagiaire de M2 ensuite recruté comme ingénieur en CDD, a travaillé sur ces aspects (Duez *et al.*, 2016), puis différents contrats ingénieurs ont permis de continuer le développement. Cette applica-

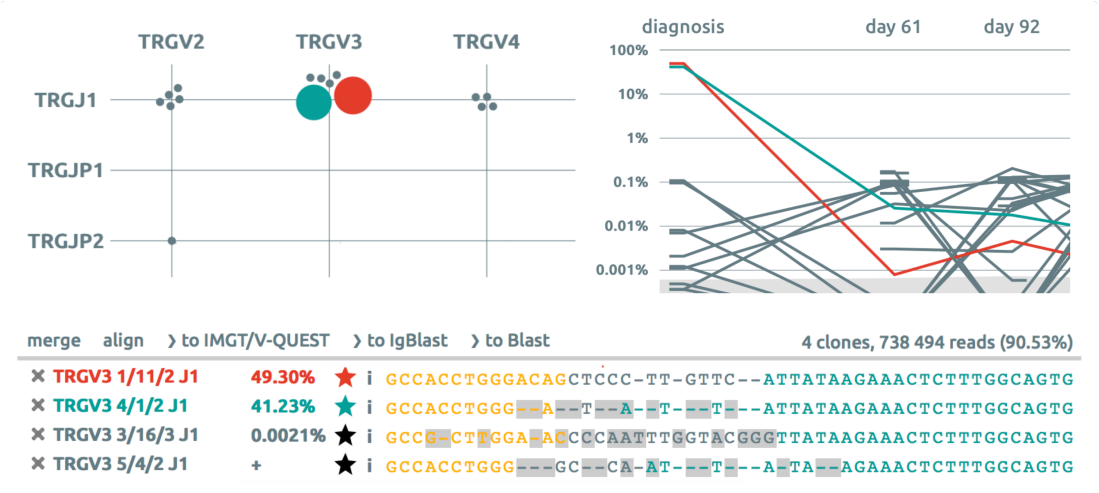

FIGURE 2.7 – Trois vues de l'application web Vidjil : en haut à gauche, la vue en *scatterplot* où chaque cercle représente une population de séquences ADN partageant la même fenêtre (un même clonotype), arrangé selon les gènes V et J du clonotype, avec la taille des cercles qui correspond à la quantification associée à chaque clonotype ; en haut à droite, le graphique de l'évolution au cours du temps des différents clonotypes ; en bas, la sélection de quelques-unes des courbes/cercles permet de visualiser les séquences ADN correspondantes et de les aligner, voire de les envoyer à des outils externes pour confirmer les analyses ou pour obtenir de plus amples informations.

tion web est également testée fonctionnellement par interaction automatique avec le navigateur, y compris sur des navigateurs anciens reflétant l'usage dans les hôpitaux, afin d'éviter le plus possible des problèmes bloquants qui impacteraient les hôpitaux dans leur utilisation quotidienne.

Le code de l'application globale est accessible sur un dépôt Git[11] et des images Docker pour l'application web sont fournies sur Docker Hub.

2.2.6 Impacts des recherches

Le projet a été initié par une collaboration avec l'IRCL et le département d'hématologie du CHU de Lille. De fait, l'outil que nous avons développé avec Mathieu Giraud était donc destiné à une utilisation probable en hôpital. Des études rétrospectives, en lien avec nos collègues du CHU de Lille, ont permis de s'assurer de la fiabilité des résultats obtenus, y compris pour le suivi de maladie résiduelle (Salson *et al.*, 2017).

Rapidement, le CHU de Lille a pu prendre possession du logiciel, en toute indépendance, et l'utiliser en parallèle de la méthode conventionnelle dès janvier 2015 afin de vérifier la robustesse des résultats obtenus avec Vidjil sur les données de séquençage à haut débit (Ferret *et al.*, 2016).

En parallèle, d'autres hôpitaux ont eu connaissance de l'application. En particulier, des hôpitaux parisiens (Necker, Pitié-Salpêtrière) ont rapidement évalué l'application et l'ont adoptée. L'un de ces hôpitaux a d'ailleurs recruté un ingénieur, Florian Thonier, afin de travailler sur l'application web Vidjil. L'application web intuitive n'est pas étrangère à l'adoption précoce et rapide de Vidjil.

Néanmoins, je pense que l'algorithme y a joué un rôle également. Loin de l'image d'utilisateurs qui voient l'application comme une boîte noire, nos collègues biologistes ou hématologues cherchent à comprendre le fonctionnement interne de notre algorithme, afin d'en cerner les limites. Pour autant, je ne prétendrais pas qu'il s'agit d'algorithmicien·ne·s sensibles à la beauté d'une heuristique et que c'est cet aspect qui aurait suscité leur adhésion mais, à l'inverse, l'aspect « boîte noire » de certaines solutions commerciales peut les rebuter. De plus, si nous avons pu proposer une application web ouverte publiquement, sans financement dédié, pour lancer des analyses sur des données de séquençage à haut débit, c'est que l'heuristique

11. https://gitlab.inria.fr/vidjil/vidjil

de notre logiciel Vidjil-algo était frugale. Aussi, un simple serveur, loué quelques euros par mois, moins puissant que mon ordinateur portable de l'époque suffisait à répondre aux besoins. Il n'en aurait pas été de même avec les logiciels qui utilisent l'autre approche, celle de commencer par aligner tous les reads.

Avec l'aide des responsables des départements d'hématologie de ces hôpitaux parisiens (Elizabeth Macintyre et Frédéric Davi), nous avons intégré en 2014 le consortium européen EuroClonality-NGS, chargé de développer, standardiser et valider des protocoles d'analyse de recombinaisons V(D)J dans le cadre du suivi des cancers hématologiques. Ce consortium nous a ouvert la porte de laboratoires européens qui ont commencé à utiliser Vidjil (Allemagne, Italie, République Tchèque, Royaume-Uni), même si ce consortium développait également un logiciel en interne. Grâce à cette implication dans le consortium, l'application Vidjil figure dans les recommandations du consortium (même si, évidemment, de manière moins percutante que pour le logiciel interne) (Brüggemann *et al.*, 2019). D'autre part, il s'agit du seul logiciel cité dans deux chapitres, portant sur l'identification de marqueurs dans les leucémies aiguës lymphoblastiques et sur l'évaluation du statut mutationnel dans les leucémies lymphoïdes chroniques, dans un livre édité par le consortium EuroClonality-NGS (de Septenville *et al.*, 2022 ; Villarese *et al.*, 2022).

Une structure pour pérenniser l'application

Avec l'adoption croissante de notre application Vidjil, en particulier par des structures hospitalières ayant des contraintes réglementaires d'accréditation, son support par une équipe de recherche n'était ni raisonnable ni envisageable. En effet, en 2019, ce sont chaque jour pas moins d'une cinquantaine de lancements de notre algorithme (Vidjil-algo) que nous avons recensés sur notre serveur principal. Ce sont également plusieurs hôpitaux qui sont passés au séquençage à haut débit et à l'utilisation de Vidjil pour l'analyse des recombinaisons V(D)J en routine hospitalière. En 2023, ce sont au moins les hôpitaux de Lille, Paris (Necker, Pitié-Salpêtrière, Robert Debré, Saint-Louis), Caen, Nantes, Limoges, Lyon, Nice, Rennes, Toulouse, Bruxelles (Belgique), Monza (Italie), Padoue (Italie), Vilnius (Lituanie), Londres (Royaume-Uni) et Boldrini (Brésil) qui ont recours à Vidjil en routine hospitalière. D'autres, comme Strasbourg ou Lausanne, sont en phase d'expérimentation.

Nos tutelles nous ont fortement incités à monter une start-up afin de pérenniser l'activité autour de Vidjil. Néanmoins, une telle demande me semble éthiquement questionnable, pour des raisons bien mises en avant dans les recommandations de justice sociale du comité éthique de HUGO (Human Genome Organisation) :

> *Therefore, genomic research should be a reciprocal exchange between individuals and communities, with researchers, funders, and sponsors, so that all participants (human beings as originators of sequences) share in the benefits of the research through knowledge dissemination and progress, and not just as end-product users, for the reason that may create inequity because of commercial interests and differential access.*
>
> Capps *et al.* (2019)

Pour ces raisons éthiques d'une part, pour des raisons pratiques d'autre part, ce n'est pas la solution que nous avons privilégiée. Nous avons cherché à monter une structure la plus ouverte possible, afin de rester dans l'esprit du logiciel libre que nous avions mis au point, et afin de poursuivre l'aspect collaboratif fort, que nous avons eu dès le démarrage de ce projet.

À partir de 2018, nous avons construit un consortium sans but lucratif, VidjilNet, au sein de l'action InriaSoft, conduite par l'Inria, destinée à pérenniser les logiciels libres développés au sein d'équipes Inria. Nous avons ainsi pu monter une structure répondant à nos aspirations, à laquelle des hôpitaux adhèrent (en fonction de leur usage du logiciel) et qui leur offre une écoute, une co-construction du logiciel, ainsi qu'une garantie sur des aspects réglementaires. Les adhésions des hôpitaux servent ensuite à financer les salaires des ingénieurs qui travaillent sur le projet (1,5 à 2 ETP), ainsi que les frais annexes. Florian Thonier a été le premier ingénieur ainsi financé dès 2019, ensuite rejoint par Marc Duez en 2020. Le contrat de Marc s'est terminé et il n'a pas souhaité continuer. Un nouvel ingénieur rejoint l'équipe fin 2023, Clément Chesnin. À terme, notre but est que le consortium VidjilNet puisse à la fois s'auto-financer et s'auto-gérer sans que Mathieu Giraud ou moi n'y ayons de rôle d'encadrement.

Le lien que nous avons tissé avec notre communauté d'utilisateurs hospitaliers a été reconnu par la remise de l'accessit du prix du logiciel libre de la recherche dans la catégo-

rie « communauté » remis en 2022 par le ministère de l'enseignement supérieur et de la recherche [12].

2.2.7 Pistes de recherches

L'utilisation importante de Vidjil en milieu hospitalier a inévitablement amené un certain nombre de questions, certaines soulevant des aspects algorithmiques intéressants.

2.2.7.1 Indexation

La quantité importante de données que nous avons accumulée au cours du temps (début 2023, ce sont plusieurs dizaines de milliers d'échantillons qui ont été analysés) a rapidement posé la question de l'indexation de ces données.

En effet, afin d'assurer le suivi des patients, les biologistes médicaux utilisent Vidjil pour identifier les clonotypes majoritaires et utilisent les séquences ADN de ces clonotypes afin de définir des sondes pouvant être utilisées en PCR quantitative pour mesurer la maladie résiduelle. La conception de sondes pose la question de la spécificité d'une telle sonde. Or, on sait que certains clonotypes peuvent être relativement courants (on parle de clonotypes publics ou partagés (Briney *et al.*, 2019 ; James et King, 2020)), il s'agit par exemple des clonotypes avec peu ou pas d'insertion/délétion. Pour le suivi d'un cancer, il serait ennuyeux d'utiliser une séquence ADN qui corresponde à un clonotype public. Dans un tel cas, la maladie résiduelle quantifiée correspondrait-elle vraiment à de la maladie résiduelle ou à la présence habituelle de clonotypes publics ?

Pouvoir rechercher dans les données déjà séquencées permet de s'assurer qu'un clonotype particulier n'a jamais été rencontré auparavant parmi les séquences des patients déjà analysés. Cette solution d'indexation, qui associe à la fois des données de séquençage et certaines métadonnées, est détaillée en section 3.3 page 36.

2.2.7.2 Exploitation des données existantes

Certains facteurs pronostiques ont été établis en fonction des décennies de recul à partir de l'analyse de données de recombinaisons V(D)J obtenues par électrophorèse sur gel ou séquençage Sanger. Ainsi, pour les leucémies aiguës lymphoblastiques (LAL), la quantification des clonotypes principaux, un mois après le diagnostic, définit le niveau de risque d'évolution défavorable de la maladie. Concernant les leucémies lymphoïdes chroniques, ce risque est évalué par le taux de mutation du gène V du clonotype principal.

Pour autant, l'avènement du séquençage à haut débit donne accès à des informations d'une richesse bien plus importante. L'exploitation d'une partie des données analysées par Vidjil, en les mettant en lien avec des informations de survie ou de rechute des patients, pourraient permettre d'identifier de nouveaux facteurs pronostiques s'appuyant sur la totalité du répertoire plutôt que sur le ou les clonotypes principaux. Idéalement, ces nouveaux facteurs seraient plus efficaces à classer les patients à bas risque ou à haut risque, mais ils pourraient aussi servir à avancer cette stratification dès le diagnostic pour les LAL et non plus un mois après.

Dans un tel projet, le défi n'est pas tant du côté bioinformatique que du côté clinique, avec la nécessité de connaître l'évolution du patient, alors que les biologistes médicaux n'ont pas nécessairement cette information. La tenue d'un tel projet nécessiterait des financements, à la fois pour la partie bioinformatique, mais également pour la partie clinique. En effet, il serait nécessaire de recontacter les hôpitaux d'où proviennent les patients afin d'avoir des informations à jour sur leur devenir.

2.2.7.3 Décomposition en blocs

Le mécanisme de recombinaisons V(D)J apparaît simple et propre, tel que je l'ai présenté. Néanmoins, au fur et à mesure de l'évolution du projet, nous avons pu constater que ce mécanisme n'était pas toujours aussi bien réglé que la théorie pourrait le laisser penser. Le cas le plus simple de dérogation à la règle correspond aux recombinaisons incomplètes, pour lesquelles le mécanisme de recombinaison n'est pas allé à son terme. Des recombinaisons moins

12. https://www.enseignementsup-recherche.gouv.fr/fr/remise-des-prix-science-ouverte-du-logiciel-libre-de-la-recherche-83576

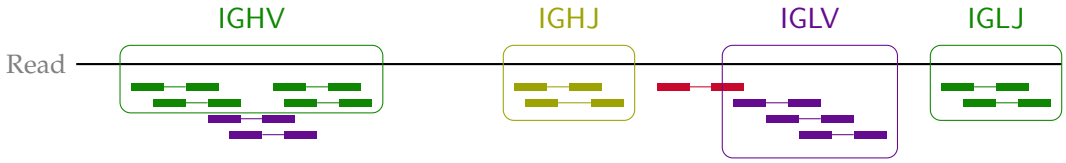

FIGURE 2.8 – Illustration du résultat d'une heuristique plus souple de détection de séquences appartenant à différents groupes. Dans un premier temps, les occurrences des instances des graines sont identifiées, pour chaque groupe de séquences d'intérêt (ici les groupes de gènes V et J des différentes chaines de récepteurs, dénotés par différentes couleurs). Ensuite, à partir des occurrences ces instances, des zones cohérentes correspondant à un même groupe de séquences sont déterminées. Notons que certaines instances peuvent être ignorées car elles ne contribuent pas suffisamment pour constituer un groupe en tant que tel. Par exemple, les deux instances violettes (correspondant à IGLV) dans le premier groupe IGHV sont ignorées (car il est plus probable qu'il s'agisse d'un groupe IGHV) et l'instance rouge unique, entre le groupe IGHJ et IGLV, est également ignorée.

attendues peuvent survenir comme des recombinaisons ayant de multiples gènes D, des recombinaisons entre deux gènes V (éventuellement dans des sens différents), entre deux gènes D, etc.

Notre heuristique initiale cherchant des instances de graines correspondant à un gène V d'une part et correspondant à un gène J d'autre part était bien adaptée à une recombinaison idéale mais elle l'est moins pour des recombinaisons atypiques. Nous pouvons adapter l'heuristique pour des recombinaisons incomplètes (en fournissant les gènes D à la place du répertoire des gènes V, ce sera une recombinaison DJ qui sera cherchée), voire des recombinaisons VV inversées. Néanmoins, ce bidouillage est intellectuellement peu satisfaisant et montre ses limites sur des situations plus complexes.

À la place, il serait plus pertinent de définir une heuristique qui délimite des blocs dans une séquence à partir d'un nombre r de répertoires différents (comme dans la figure 2.8). Chaque bloc, non chevauchant, correspondrait à un facteur de la séquence présentant des similarités statistiquement significatives avec l'un des répertoires. Certaines parties de la séquence pourraient ne correspondre à aucun bloc. Dans l'état actuel de notre heuristique, nous sommes limités à $r = 2$. L'idée serait donc de généraliser l'heuristique. Avec Mathieu Giraud, nous avons co-encadré un stage court de M1 mais le sujet nécessiterait un travail plus approfondi. Un tel travail pourrait trouver des applications au-delà des recombinaisons V(D)J, dès qu'il est nécessaire de détecter rapidement des séquences composées de différents blocs.

2.2.7.4 Détection de sous-clonotypes

L'heuristique qui est au cœur de Vidjil regroupe les reads sur la base d'une identité exacte avec une fenêtre centrée sur la région de haute diversité d'une recombinaison V(D)J. Néanmoins, dans le cadre de l'évolution clonale, et *a fortiori* pour des cancers, différentes sous-populations peuvent se développer à partir d'un même clone, il serait donc opportun de les identifier. Notre heuristique ne le permet pas.

Plutôt que de songer à une autre heuristique, qui serait probablement plus complexe, nous avons identifié qu'un spectre de k-mers pourrait suffire à déterminer que différentes sous-populations ont été mélangées sous un même clonotype. En effet, un spectre de k-mers est un histogramme du nombre de k-mers ayant un nombre d'occurrences donné. Les k-mers compris dans la fenêtre sont, par définition, tous présents parmi les reads formant le clonotype. Ces k-mers constituent un extrême du spectre. À l'autre extrême figurent les erreurs de séquençage. Au fur et à mesure qu'on s'éloigne de ces extrêmes, il devrait y avoir une décroissance rapide du nombre de k-mers (voir figure 2.9 page ci-contre). Ainsi on ne s'attend pas à ce qu'un k-mer soit vu dans 60 % des reads… sauf si deux sous-populations ont été mélangées. Dans cette situation, on devrait également voir un k-mer 40 % du temps. Cette observation est une très bonne approximation pour déceler les regroupements trop larges.

Pierre Doignies, Marie-Joe Karam et Guillaume Poslednik, pendant leur projet de fin d'année de M2 bioinformatique de Lille, ont mis en place un prototype afin de pouvoir tester cette hypothèse. Agathe Bancquart, stagiaire du M1 bioinformatique, a repris ce prototype l'année

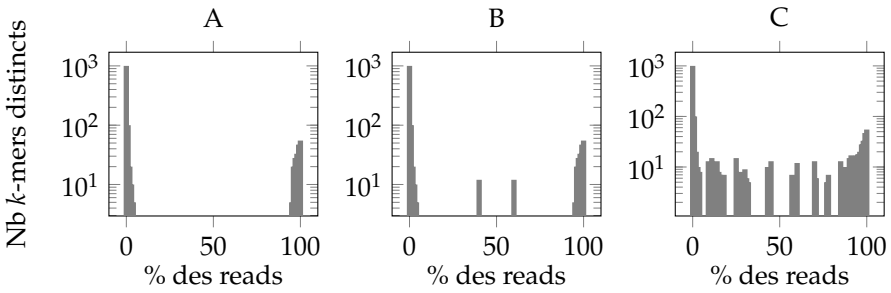

FIGURE 2.9 – Différents spectres de *k*-mers pour illustrer la détection de sous-clonotypes. Chaque spectre montre le nombre de *k*-mers distincts présents dans un pourcentage donné des reads du clonotype. Par exemple, il y a une cinquantaine de *k*-mers qui sont présents dans 100 % des reads du clonotype (ce sont les *k*-mers de la fenêtre). A. Situation où il n'y a pas de sous-clonotype : les pics à droite correspondent aux *k*-mers biologiques, les pics à gauche aux erreurs de séquençage. B. Deux pics symétriques sont présents autour de 40 % et 60 %. Cela suggère qu'il existe deux sous-clonotypes distincts : celui qui possède les *k*-mers de gauche (présents dans 40 % des reads) et celui qui possède les *k*-mer de droite (présents dans 60 % des reads). C. Le dernier cas est plus difficilement distinguable. De nombreux sous-clonotypes sont probablement mélangés.

suivante. Elle a analysé la pertinence des résultats et a raffiné les critères pour distinguer les sous-clones, afin de chercher à avoir des résultats satisfaisants.

Par la suite, il faudra intégrer une telle approche dans le logiciel Vidjil afin qu'elle puisse être proposée à nos utilisateurs et utilisatrices.

2.3 Identification des duplications en tandem

Les marqueurs génétiques dans les leucémies ne se restreignent pas aux recombinaisons V(D)J mais sont nombreux. Dans les leucémies aiguës myéloïdes (LAM), le marqueur le plus fréquemment identifié est une duplication en tandem dans le gène FLT3, entre les exons 14 à 15. Cette altération est présente dans 12 % à 38 % des cas et est prédictive d'un plus fort risque de rechute et d'une survie globale réduite (Beitinjaneh *et al.*, 2010 ; Wu *et al.*, 2016). La longueur de ces duplications est très variable : de quelques nucléotides à quelques centaines de nucléotides mais il s'agit quasi-systématiquement d'un multiple de 3, ce qui conserve le cadre de lecture. La longueur de la duplication semble également être un facteur pronostique de la maladie (Polak *et al.*, 2022). Enfin, le ratio allélique de la duplication (le rapport entre la quantification de la version contenant la duplication et la quantification de la version germinale) est un critère important. Les recommandations européennes mentionnent un ratio allélique de plus de 0,5 comme facteur de risque (Döhner *et al.*, 2017).

De manière similaire à la genèse de Vidjil, le département d'hématologie du CHU de Lille nous a sollicités en 2020 afin de mettre au point une méthode fiable. Le but du CHU de Lille est d'effectuer l'analyse de la duplication en tandem de FLT3 (FLT3-ITD) par séquençage à haut débit, dans le cadre d'un protocole de capture ciblant d'autres gènes pertinents dans les LAM. Un logiciel doit pouvoir identifier les duplications en tandem dont le ratio allélique est au moins de 0,5, déterminer leur séquence et estimer correctement le ratio allélique. Des méthodes existantes et testées par Augustin Boudry, interne en pharmacie à la plateforme de bioinformatique du CHU, co-encadré avec Martin Figeac, présentent des limites en terme de sensibilité ou de quantification.

Problème

> *À partir d'un ensemble de reads d'une expérience de séquençage à haut débit, proposer une méthode efficace pour identifier les duplications en tandem issues d'un ou plusieurs courts gènes de référence, caractériser la longueur de ces duplications et quantifier cette duplication par rapport à la version germinale.*

Un article de synthèse de 2021 résume les différentes approches alors existantes (Yuan *et al.*, 2021). L'ensemble des approches recensées s'appuient sur l'information venant de l'ali-

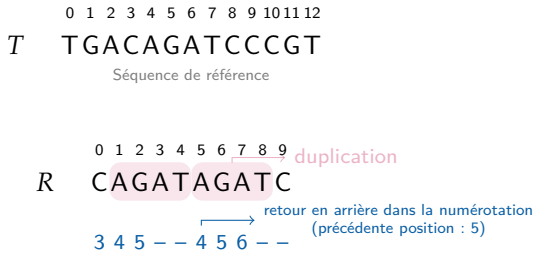

FIGURE 2.10 – Identification de duplication d'une séquence de référence T dans un read R avec des k-mers ($k = 3$). La ligne du bas représente les positions des k-mers de R dans T. Par exemple, le premier élément (3) correspond au k-mer CAG qui apparaît en position 3 dans T. Les tirets (–) correspondent aux k-mers de R qui n'apparaissent pas dans T (par exemple ATA ou TAG). Figure traduite de Boudry *et al.* (2022).

gnement des reads sur un génome de référence (hormis GetITD qui aligne uniquement sur la séquence d'intérêt). Ensuite, les auteurs distinguent deux catégories : d'une part les approches qui utilisent les résultats de l'alignement pour détecter la duplication (éventuellement en procédant à des réalignements), d'autre part les approches qui vont assembler tout ou partie des reads discordants en contigs correspondant à des duplications potentielles sur lesquelles les reads pourront être réalignés. À l'inverse des méthodes présentées dans cette synthèse, nous avons fait le choix d'une approche sans alignement. Ce n'est néanmoins pas la seule, km est également une approche sans alignement (qui n'était pas intégrée à la synthèse) mais qui procède par comptage des k-mers et utilisation d'un graphe de de Bruijn pour identifier les duplications ou d'autres événements biologiques (Audemard *et al.*, 2019).

Ici, l'analyse porte sur une séquence de référence très courte de moins de 1 000 nt. L'indexation d'une telle séquence n'est donc pas le défi. Au-delà de la précision de l'analyse, mon expérience de Vidjil m'a laissé penser qu'une analyse rapide serait un atout précieux pour deux raisons :

1. Plusieurs échantillons de patients sont séquencés en une seule fois, les analyses n'arrivent donc pas au fil de l'eau mais en bloc. Une heure de traitement pourrait sembler dérisoire, mais s'il faut la multiplier par plusieurs dizaines d'échantillons, ça ne l'est plus tant que ça.
2. Plusieurs gènes étant ciblés par le panel de capture, différentes analyses doivent être réalisées. L'identification des FLT3-ITD n'est qu'une des diverses analyses à mener.

À ces questions pratiques s'ajoute une question éthique : est-il raisonnable d'utiliser des ressources de calcul (pour calculer des alignements, par exemple) dont on aurait pu se dispenser en utilisant une autre approche ? À la place, ces ressources auraient pu être allouées à d'autres projets ou les clusters de calcul sollicités pourraient être plus frugaux.

L'approche utilisée dans CRAC (voir section 2.1 page 6), bien que très simple, avait le mérite de permettre d'identifier des événements variés. En adaptant la stratégie et en la spécialisant pour la recherche de duplications, nous avons un levier pour une approche sans alignement.

L'idée, comme dans CRAC, ou d'une certaine manière dans Vidjil, est de découper chaque read en k-mers et d'utiliser les informations fournies par ces k-mers pour identifier une duplication. Les k-mers de la séquence de référence, une partie du gène FLT3, sont connus. Détecter une duplication consiste donc simplement à détecter un retour en arrière dans la localisation des k-mers d'un read sur la référence. En effet, au fur et à mesure de la lecture des k-mers dans le read, on s'attend à ce que ces k-mers se suivent dans la séquence de référence. Dès qu'un k-mer nous fait revenir en arrière dans la séquence de référence, nous pouvons soupçonner une duplication (voir figure 2.10).

Bien que simple, cette méthode est en réalité riche d'informations : en effet les positions des k-mers suffisent à déterminer la longueur de la duplication et même à identifier si des insertions ont eu lieu en plus d'une simple duplication. Supposons que nous ayons une duplication d'une séquence d. Nous avons donc une séquence $d \cdot d$, que nous appellerons $d_1 d_2$ afin de différencier les deux parties de la duplication. Les $k-1$ k-mers chevauchant le point de cassure entre d_1 et d_2 débutent dans d_1 en positions $|d| - (k-1)$ à $|d| - 1$ et, *a priori*, n'existent pas

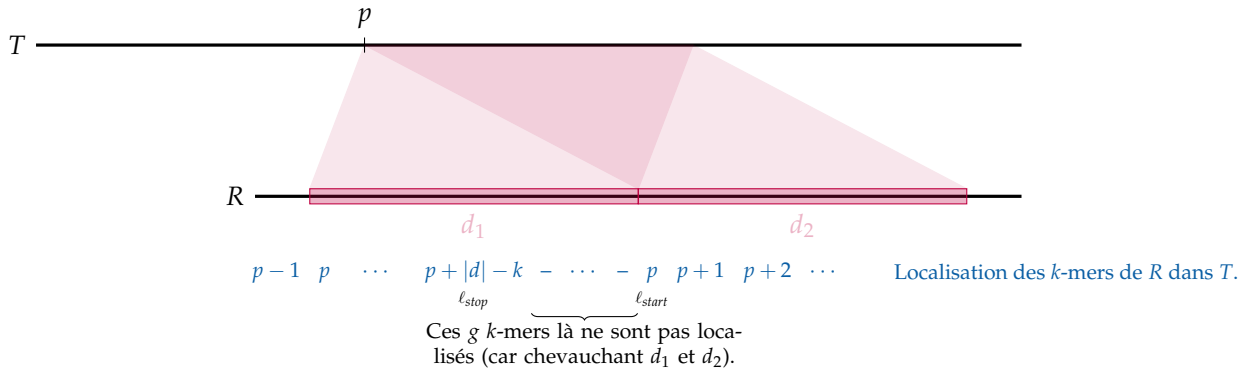

FIGURE 2.11 – Localisation des k-mers d'un read lorsque celui-ci contient une duplication. Le read contient une duplication $d_1 = d_2$, localisée en position p dans T. La localisation des k-mers de R est indiquée, comme dans la figure 2.10. Le k-mer qui démarre au début de d_1 est, par définition, localisé en position p dans T, de même pour le premier k-mer de d_2. Les g derniers k-mers de d_1 ne sont pas localisés (ici $g = k - 1$) car ils chevauchent d_1 et d_2.

dans la référence. Notons g le nombre de ces k-mers non localisés, qui est ici $k - 1$. Appelons k_{stop} le dernier k-mer à être bien localisé dans d_1, il s'agit du k-mer commençant en position $|d| - k$ dans d_1 (et finissant donc en position $|d| - 1$). Le premier k-mer à être de nouveau localisé (appelons-le k_{start}) est le premier k-mer de d_2. Supposons que d apparaisse en position p dans la séquence de référence. Alors, k_{stop} apparaît en position $\ell_{stop} = p + |d| - k$ dans la séquence de référence et k_{start} apparaît en position $\ell_{start} = p$ (voir figure 2.11). La longueur de la duplication est déduite de $\ell_{stop} + g + 1 - \ell_{start} = p + |d| - k + (k - 1) + 1 - p = |d|$. Le cas théorique présenté ici est idéal : le dernier k-mer à être localisé n'est pas nécessairement celui en position $|d| - k$, celui en position $|d| - (k - 1)$ pourrait aussi être localisé du moment où $d[0]$ est identique au nucléotide qui suit d dans la séquence de référence. Dans ce cas, ℓ_{stop} est incrémentée de 1, mais g (le nombre de k-mers non localisés) est diminué de 1. L'égalité reste donc vraie malgré des substitutions, insertions ou délétions au niveau de la duplication (voir figure 2.12 page suivante). Néanmoins, une approche par k-mers présente nécessairement des limites auxquelles il convient de prêter attention.

2.3.1 Atténuer l'impact des faux positifs

Grâce à l'expérience acquise avec CRAC, j'ai rapidement eu conscience du risque de faux positifs à cause de k-mers erronés qui pourraient par hasard correspondre à un k-mer de la référence. Dans un tel cas on risquerait de détecter, à tort, une duplication en pensant qu'il y a eu un retour en arrière dans la séquence de référence.

Pour limiter l'impact de tels faux positifs, nous avons deux stratégies. La première consiste à corriger toutes les substitutions qui peuvent l'être : dès qu'un k-mer n'est pas localisé, si une substitution permet de le localiser dans la continuité des précédents, la correction est appliquée. La seconde consiste à vérifier la cohérence des positions sur plusieurs k-mers consécutifs juste avant et juste après le point de cassure de la duplication. Cela revient à simuler des k-mers plus grands.

2.3.2 Rectifier la quantification

La quantification est importante pour la stratification des patients, en particulier avec le critère d'un ratio allélique de 0,5 indiqué précédemment. Aussi, celle-ci doit être la plus fiable possible. Or, notre approche fondée sur les k-mers souffre d'un biais : certains événements présents dans les reads ne pourront être détectés. En effet, pour qu'une duplication soit détectée, il faut que des k-mers avant et après la duplication soient localisés sur la référence. Ainsi, toute duplication à moins de $K + 1$ nucléotides [13] d'une extrémité du read ne sera pas détectée (en tout cas dans ce read-là). La couverture étant généralement importante (pour les données de capture cela peut être de l'ordre de 1 000x), il n'y a pas vraiment de risque de faux négatif,

13. Nous notons K la longueur des k-mers allongés simulés afin d'éviter les faux positifs.

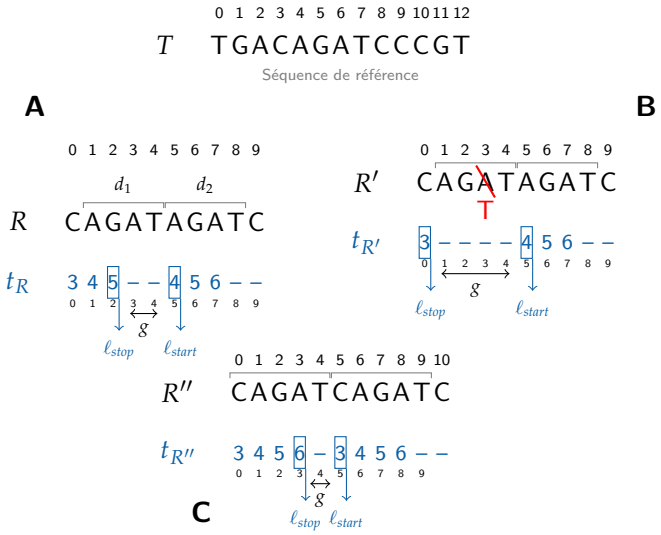

FIGURE 2.12 – Déduction de la longueur de la duplication à partir des occurrences des k-mers. Figure adaptée de Boudry *et al.* (2022). Dans cet exemple, $k = 3$. **A.** Le k-mer k_{stop}=GAT présent, une première fois, en position 2 dans R est localisé en position 5 dans T (ainsi qu'indiqué dans t_R). La longueur de la duplication est déduite de $\ell_{stop} + g + 1 - \ell_{start} = 4$, soit la longueur de d_2. **B.** Cas où une mutation survient dans une région dupliquée. Par rapport à la version A, k_{stop} apparaît deux positions plus tôt mais g fait alors deux positions de plus. Les deux se compensent et on retrouve bien que la duplication est de longueur 4. **C.** Cas où la première lettre de la duplication (C) est identique à la lettre qui suit la duplication, en position 10. Dans ce cas, aussi bien CAGAT que AGATC peuvent être considérés comme dupliqués. Dans ce cas, k_{stop} est localisé une position plus à droite que dans le cas A, ce qui est compensé par le fait que g fait une position de moins.

mais de biaiser la quantification puisque certaines duplications ne seront pas détectées alors que les reads provenant de la version sans duplication seront, eux, bien comptabilisés. Ce biais de quantification a donc pour effet de sous-estimer le ratio allélique.

Néanmoins, il existe une correction simple à apporter : elle consiste à estimer le nombre de duplications qui n'ont pu être détectées (ce qui dépend directement de la valeur de K choisie et de la longueur des reads) et de rajouter cette estimation au nombre de reads effectivement détectés comme présentant la duplication. En supposant que les reads sont uniformément distribués [14] le long de la séquence de référence, il y a une proportion de $\frac{2(K+1)}{m}$ duplications qui ne seront pas détectées, avec m la longueur du read. La quantification est donc corrigée pour prendre en compte ces duplications manquantes. Nos expériences montrent que cette correction améliore sensiblement la quantification.

2.3.3 Améliorer les performances de l'algorithme

2.3.3.1 Performances en temps

Concevoir une méthode sans alignement ne garantit pas d'obtenir les meilleures performances possibles, même si les performances seront probablement supérieures aux méthodes avec alignement.

Dans la situation qui nous concerne, à moins d'avoir des données de séquençage ayant uniquement ciblé les exons 14 à 15 du gène FLT3, il est très probable que la grande majorité des reads ne couvrent pas cette région. Aussi, interroger une table de hachage avec chaque k-mer du read pour que celle-ci réponde systématiquement que la clé n'existe pas semble relativement peu efficace, bien que la complexité asymptotique d'interrogation d'une table de hachage à adressage ouvert soit de $\mathcal{O}(1 + \tau)$, où τ est le taux de remplissage de la table (Cor-

14. À une échelle de quelques dizaines de nucléotides, l'approximation semble acceptable mais sera vérifiée expérimentalement.

men *et al.*, 2022). Une alternative consiste en l'utilisation préalable d'un filtre de Bloom (1970), une structure de données renseignant de manière approximative (il peut y avoir des faux positifs) sur la présence ou l'absence de données indexées. Dans le contexte des FLT3-ITD, un filtre de Bloom de 100 000 bits suffit amplement, ce qui en fait une structure pouvant tenir dans le cache L1 du processeur, le plus rapide. Avec trois fonctions de hachage, la probabilité d'un faux positif est inférieure à 10^{-4}, ce qui est largement suffisant. Le filtre sert d'oracle pour déterminer s'il est opportun d'interroger la table de hachage. Le filtre de Bloom sera donc interrogé avec tous les k-mers du read et si, pour une proportion suffisante d'entre eux, le filtre de Bloom indique une présence dans la séquence de référence, alors la table de hachage pourra être interrogée.

Ce simple ajout d'un filtre de Bloom a permis un gain en temps d'un facteur 12 dans nos expériences (Boudry *et al.*, 2022). Le multi-threading n'apporte qu'un faible bénéfice car le facteur bloquant devient rapidement la décompression des fichiers en entrée (des FASTQ gzippés). En effet, notre approche ne prend « que » trois fois plus de temps que la seule décompression, avec GUNZIP, sur les mêmes fichiers.

2.3.3.2 Performances en mémoire

L'algorithme passe dans un premier temps sur tous les reads et retient ceux qui présentent suffisamment de k-mers communs avec la séquence de référence. Dans un second temps, ces reads filtrés doivent être analysés afin d'identifier s'ils présentent une duplication. Ensuite, seules les duplications détectées un nombre suffisant de fois sont conservées. Enfin, pour chaque duplication, leur quantification est estimée et les reads correspondants à cette duplication sont utilisés afin de calculer la séquence de chaque duplication.

Afin d'éviter de reparcourir la totalité du fichier originel de reads, les reads filtrés doivent être conservés en mémoire. L'espace mémoire nécessaire est néanmoins limité puisque le nombre de reads présents dans la région d'intérêt est (pour l'instant) restreint. Pour autant, sur certains jeux de données, cet espace mémoire peut malgré tout atteindre quelques centaines de mégaoctets. Un tel volume peut sembler négligeable en regard de la place utilisée par d'autres logiciels du même type (plusieurs gigaoctets). Néanmoins, il n'y a pas de réelle utilité à conserver ces données dans la mémoire centrale de l'ordinateur alors que nous avons uniquement besoin d'itérer sur les reads, donc sans nécessiter d'accès aléatoire. Par conséquent, nous avons également développé une version où les reads filtrés sont stockés sur disque. Dans cette version, nous n'observons pas de pénalité sur le temps d'exécution. En revanche, l'espace mémoire utilisé ne dépasse pas la vingtaine de mégaoctets.

2.3.4 Mise en œuvre

L'approche a fait l'objet d'un prototype en Python par Sasha Darmon, stagiaire de L3 et de nombreuses expériences ont été conduites par Augustin Boudry. J'ai réécrit le code du logiciel (FiLT3r) en C++, que nous avons publié en nous appuyant sur des données de patients obtenues par le CHU de Lille, avec une comparaison à la méthode conventionnelle (Boudry *et al.*, 2022). Le logiciel est disponible sous licence libre sur un dépôt Git [15], et peut être utilisé via Docker.

Dans le but de favoriser l'utilisabilité à moindre frais, nous avons initialement envisagé de proposer une page web sur laquelle pourrait être faite l'analyse de FiLT3r uniquement côté client. Une telle possibilité est envisageable en utilisant `emscripten` (Zakai, 2011), qui permet de compiler du code C/C++ en Javascript pour une utilisation directe dans le navigateur. Malheureusement, cela s'est avéré plus compliqué qu'espéré car la compilation de la bibliothèque GATB (Drezen *et al.*, 2014) que nous utilisons dans FiLT3r pose des soucis encore non résolus avec `emscripten`.

Dans notre article (Boudry *et al.* (2022)), nous montrons sur une cohorte de 185 patientes et patients, que le logiciel est bien plus efficace en ressources de calcul (temps et mémoire) et obtient des résultats légèrement meilleurs que la meilleure approche identifiée par Yuan *et al.* (2021) (FLT3-ITD-Ext) et que km. Les résultats sont à la fois meilleurs en terme de quantification : les logarithmes des quantifications de FiLT3r corrèlent à 94 % avec la méthode de référence, contre 90 % pour la méthode qui suit, FLT3-ITD-Ext ; d'autre part, FiLT3r n'a aucun

15. https://gitlab.univ-lille.fr/filt3r/filt3r

faux négatif, ce qui n'est pas le cas des autres méthodes testées. Aucune méthode n'a de faux positif.

Concernant les heuristiques introduites pour éviter les faux positifs, nous constatons que désactiver l'heuristique qui retire les substitutions entraîne la détection d'un nombre accru d'événements, en particulier de très courts événements qui semblent être des faux positifs. En revanche, l'effet de la simulation des *k*-mers allongés ne présente pas d'intérêt, ni d'inconvénient : sur notre cohorte, les résultats restent identiques avec ou sans cette heuristique.

2.3.5 Impacts des recherches

Le projet est encore récent pour avoir un retour pertinent. Néanmoins, le logiciel est utilisé chaque semaine au CHU de Lille pour l'analyse des échantillons de patients atteints de LAM (environ 80 par semaine). Augustin Boudry a également présenté FiLT3r et nos résultats à l'*European LeukemiaNet*, un réseau de plus de 200 centres répartis dans 44 pays pour le diagnostic et le traitement des leucémies. Il a également présenté ce travail à la société française des biologistes moléculaires et est chargé de faire le contrôle qualité de plus de 50 laboratoires en utilisant FiLT3r.

Chapitre 3

Indexation de données nucléiques

3.1 Méthodes d'indexation

L'analyse de données issues des séquenceurs à haut débit n'est envisageable qu'avec des méthodes capables de rechercher dans des volumes de données conséquents, qu'il s'agisse des séquences brutes ou des séquences assemblées, comme des génomes. De telles recherches sont le cas d'usage parfait pour la construction de structures d'indexation puisque les données dans lesquelles rechercher (des génomes, des jeux de données brutes) sont statiques ou changent peu souvent et sont très fréquemment requêtées. À l'arrivée des séquenceurs à haut débit, les équipes de bioinformatique ont pu piocher dans les structures d'indexation pré-existantes, qui avaient été conçues dans la communauté de l'algorithmique du texte.

3.1.1 Indexation d'une seule séquence de référence

Ainsi, les premiers outils de mapping ne s'appuyant pas sur de « banales » tables de hachage se sont naturellement portés sur le FM-index, une structure d'indexation compressée publiée en 2000 (Ferragina et Manzini, 2000) et reposant sur une technique de compression (la transformée de Burrows-Wheeler) ayant de très fortes similitudes avec une autre structure d'indexation, la table des suffixes (Manber et Myers, 1990). L'arbre et la table des suffixes étaient, jusqu'au début du XXI^{e} siècle, les structures d'indexation de référence pour indexer intégralement un texte. Leur inconvénient majeur est leur consommation mémoire, plusieurs fois supérieure à l'espace mémoire nécessaire pour le texte lui-même (Lecroq et Salson, 2022).

La transformée de Burrows-Wheeler (BWT), introduite par Burrows et Wheeler (1994) dans un rapport interne à leur entreprise, est une opération visant à réorganiser les lettres du texte et qui a tendance à rapprocher les lettres identiques (voir figure 3.1).

La transformée de Burrows-Wheeler désigne à la fois l'opération et son résultat. Afin de distinguer les deux, il est parfois question de « transformation » de Burrows-Wheeler pour l'opération qui permet d'obtenir la transformée. La transformation de Burrows-Wheeler d'un texte T consiste à concaténer (conceptuellement) la dernière lettre de chacune des permutations circulaires de T triées dans l'ordre alphabétique (voir figure 3.2 page suivante). À partir du moment où le texte T finit par un terminateur différent de tous les autres caractères, le tri des permutations circulaires est équivalent au tri des suffixes, auquel on procède pour le calcul de la table des suffixes, d'où la proximité entre les deux concepts. La transformation permet une compression plus efficace grâce à la propension de la BWT à rassembler les lettres identiques. Une propriété de la BWT, utilisée pour inverser la transformation, sert également à rechercher des occurrences d'un motif dans un texte. C'est l'utilisation de cette propriété pour la recherche de motifs qui

Suite de caractères identiques (*runs*) sur le chromosome 1 humain (N supprimés)

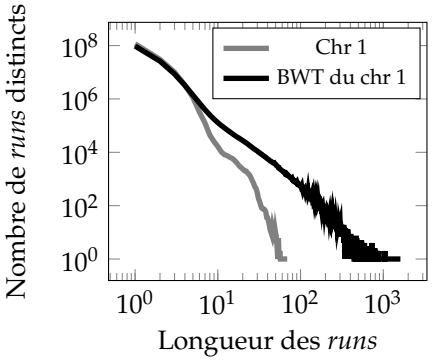

FIGURE 3.1 – Dans la transformée de Burrows-Wheeler du chromosome 1 chez l'humain, on observe des suites de lettres identiques bien plus longues que sur la séquence d'origine.

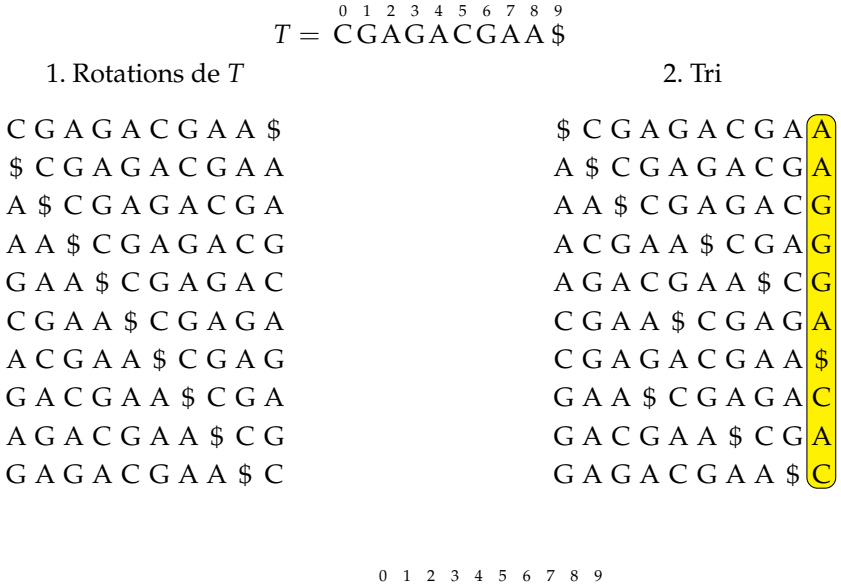

FIGURE 3.2 – Illustration du calcul d'une transformée de Burrows-Wheeler (BWT). Toutes les rotations cycliques du texte sont calculées, puis elles sont triées dans l'ordre alphabétique. La concaténation de la dernière lettre de chacune de ces rotations triées constitue la transformée de Burrows-Wheeler. On constate que dans la BWT obtenue, les trois G sont consécutifs : c'est la conséquence de la répétition de GA dans T. Il s'agit ici d'une illustration de l'obtention de la transformée de Burrows-Wheeler, mais pas d'un algorithme de calcul efficace puisque celui-ci serait quadratique en temps et en espace.

constitue la principale avancée apportée par Ferragina et Manzini (2000). La structure d'indexation qui en découle, le FM-index, repose sur une transformée de Burrows-Wheeler et diverses structures de données annexes, la principale permettant de calculer efficacement le rang d'une lettre dans la BWT.

Des structures de données fondamentales, auxquelles je me référerai plus loin, permettent de connaître en temps constant le rang d'une lettre quelconque dans un texte (opération `rank`) où la position de la i-ème lettre c dans un texte (opération `select`). Dans un premier temps, abordons la version binaire, où le texte est une suite de bits. La solution pour connaître le nombre de 1 (ou de 0) jusqu'à une position quelconque est de précalculer un certain nombre de rangs, mais suffisamment peu pour que l'espace supplémentaire nécessaire soit négligeable en regard de la taille du texte (voir Navarro et Mäkinen (2007)). Le *wavelet tree* est une structure d'indexation pour répondre à de telles requêtes sur des alphabets plus grands. Succinctement, le principe est de se ramener au cas binaire. L'alphabet du texte est divisé en deux, les lettres de la première moitié se voient attribuer un 0 et l'autre moitié un 1. Le processus est poursuivi récursivement sur les sous-mots obtenus en ne gardant que les lettres de la première moitié (sous-arbre gauche) puis celles de la deuxième moitié (sous-arbre droit). Une feuille de l'arbre est atteinte lorsque le sous-mot n'est composé que d'une seule lettre distincte, qui correspond à l'étiquette de la feuille (voir figure 3.3 page suivante). Avec cette structure, les requêtes sont en temps logarithmique dans la taille de l'alphabet, ce qui, dans le cas de l'ADN, peut être assimilé à du temps constant. La structure permet en plus de représenter le texte lui-même dans un espace succinct (c'est-à-dire asymptotiquement proche du minimum théorique), en $n \log \sigma + o(n \log \sigma)$ bits, avec n la taille du texte et σ la taille de l'alphabet.

3.1.2 Indexation de plusieurs séquences de référence

Depuis, en algorithmique du texte, la communauté a continué à proposer d'autres solutions d'indexation, en particulier afin d'indexer des textes fortement similaires (des données versionnées, par exemple sur un wiki ou un dépôt Git, ou des génomes d'individus). Certaines continuent à s'appuyer, en totalité ou en partie, sur la transformée de Burrows-Wheeler,

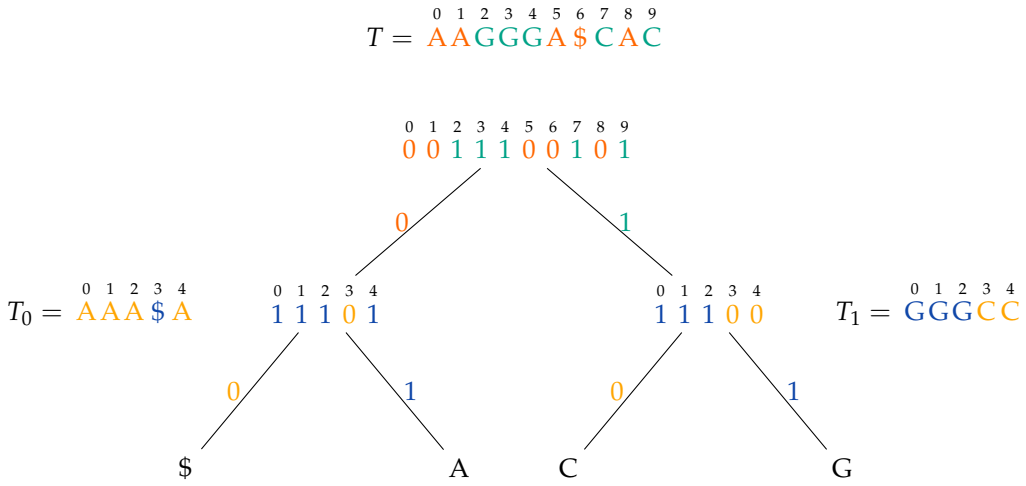

FIGURE 3.3 – Le *wavelet tree* de AAGGGA$CAC. Initialement l'alphabet de taille 4 ($, A, C, G) est divisé en deux. On attribue un 0 aux lettres $ et A et un 1 aux lettres C et G. C'est cette version binaire du texte qui est stockée à la racine du *wavelet tree*. Ensuite, le processus est renouvelé dans les sous-arbres gauche et droit sur, respectivement, le sous-mot de T composé des lettres $ et A (T_0), et sur celui composé des lettres C et G (T_1). Pour connaître le rang du A en position 5 dans T, il suffit de faire un `rank` de 0 à la racine jusqu'à la position 5, ce qui nous donne 3 (il y a 3 zéros de la position 0 à la position 5). Nous savons donc qu'il y a 3 A et $ entre la position 0 et 5 de T. Ensuite, nous renouvelons le processus dans le sous-arbre gauche, en faisant un `rank` de 1 dans le sous-arbre gauche à la position 2 (le troisième bit), ce qui nous donne 3 (il y a 3 uns de la position 0 à 2). Ensuite, le sous-arbre droit de ce sous-arbre nous amène à une feuille, ce qui achève la requête. Nous savons donc qu'en position 5 de T, il s'agissait du troisième A.

d'autres sur une autre technique de compression : la compression Lempel-Ziv et enfin certaines ont exploré l'indexation compressée en utilisant des grammaires (par exemple Mäkinen *et al.* (2010a) ; Navarro (2012) ; Valenzuela (2016) ; Belazzougui *et al.* (2017) ; Cobas *et al.* (2021), et voir la synthèse de Navarro (2021)). Pour autant, ces solutions n'ont pas encore suscité autant d'enthousiasme de la part de la communauté de la bioinformatique des séquences que le FM-index de Ferragina et Manzini. Cela peut s'expliquer d'une part en raison d'un problème de construction de certaines de ces structures, qui peuvent demander d'avoir à un instant t toutes les données en mémoire, ce qui est inenvisageable lorsqu'il s'agit d'indexer, par exemple, des milliers de génomes humains. Une autre explication tient à l'efficacité en mémoire des index obtenus. Bien que les niveaux de compression de ces structures d'indexation peuvent être très appréciables, ils restent notablement insuffisants en regard du volume de données en jeu. Ainsi, dans les expériences de Cobas *et al.* (2021), les meilleures compressions atteintes sur des jeux de données d'ADN sont au-dessus de 0,1 bit par symbole, tandis que sur les jeux de données de séquençage bruts on obtenait au mieux un décevant 2 bits par symbole. L'approche hybride de Valenzuela (2016), CHICO, qui associe une compression Lempel-Ziv avec l'indexation classique d'un texte « noyau », semble la plus efficace parmi ces solutions. L'approche hybride a en outre l'intérêt de pouvoir utiliser des indexations existantes, comme celles de BWA ou Bowtie, afin d'être compatible avec ces outils. Pour l'indexation de 2 000 chromosomes de 1 à 5 (2,4 Tpb), CHICO utilise environ 0,02 bits par symbole. Néanmoins, le problème de CHICO est sa construction, dont la consommation mémoire, proche du téraoctet sur de gros jeux de données, le rend peu adapté en pratique (Kuhnle *et al.*, 2020), ce qui a conduit à la proposition d'une construction distribuée (Maarala *et al.*, 2021), qui reste contraignante. De plus, lorsque l'index CHICO a été adapté pour être intégré dans un aligneur de données pan-génomiques, les performances se sont notablement dégradées par rapport à ce qui était présenté dans l'article princeps : 200 génomes humains étaient indexés à environ 2 bits par symbole (Valenzuela et Mäkinen, 2017).

Du côté bioinformatique des séquences, des propositions ont rapidement été formulées afin d'indexer les quantités massives de génomes ou de reads produits. Puisque les approches

d'indexation mentionnées précédemment ne donnent pas complètement satisfaction, il s'agit d'être moins ambitieux. En effet, ces approches réalisent une indexation intégrale du texte, laissant la possibilité de rechercher des séquences de n'importe quelle longueur. Or, il est envisageable de concevoir des index moins flexibles, qui seraient moins gourmands en mémoire. Une alternative a donc été d'indexer des *k*-mers plutôt que le texte intégral.

Par exemple, dès 2014, Danek *et al.* proposent une solution qui indexe un génome de référence, puis qui indexe des informations sur les *k*-mers spécifiques de certains génomes individuels. Leur solution permet aussi de définir un compromis entre l'espace de stockage pour l'indexation et la flexibilité de recherche : il est possible de n'indexer qu'une fraction des *k*-mers des génomes. Ils arrivent ainsi à indexer 2 000 génomes humains en moins de 16 Go (voire moins de 8 en jouant sur certains paramètres), soit environ 0,01 bit par nucléotide [1] en utilisant une quarantaine de gigaoctets pendant la construction. Cette solution, en dégradant la flexibilité de recherche, obtient des performances très satisfaisantes pour une utilisation en pratique sur des milliers de génomes individuels. Plus récemment, Břinda *et al.* (2023) parviennent, en optimisant le stockage grâce à la phylogénie des espèces, à indexer une collection de génomes de SARS-CoV-2 en utilisant 18 octets par génome (0,005 bit/nt) et plus de 600 000 génomes microbiens, bien plus divers que des génomes humains, en utilisant 0,06 bit/nt (ou 4 bits par *k*-mer distinct). De manière plus générale, l'idée d'indexer des *k*-mers est également la solution privilégiée pour l'indexation de données de séquençage.

3.1.3 Indexation de données de séquençage

Dans ce cas d'application, l'indexation de milliers de jeux de données est un défi, car la quantité de données se compte rapidement en téranucléotides. Néanmoins, le nombre de *k*-mers distincts peut être comparable aux quantités observées lors de l'indexation de génomes humains. À titre d'exemple, un jeu de données couramment utilisé dans la communauté, et introduit par Solomon et Kingsford (2016), consiste en plus de 2 500 RNA-seq humains, représentant 14 téranucléotides et 3,8 milliards de *k*-mers distincts (après filtrage des *k*-mers les moins fréquents). Pour autant, par rapport à l'indexation de génomes d'une même espèce, l'indexation de jeux de séquençage à haut débit présente une plus grande diversité, en raison des erreurs de séquençage qui peuvent rester même après filtrage et en raison de la diversité transcriptomique, s'agissant de l'indexation de RNA-seq.

La première solution, introduite par Solomon et Kingsford (2016), consiste en une collection de filtres de Bloom, stockée dans un arbre binaire. Chaque filtre de Bloom correspond à un jeu de séquençage et enregistre la trace des *k*-mers présents dans chaque jeu de données. Cette structure est probabiliste : il existe une probabilité non nulle de faux positifs. Là aussi, la solution pour faire face au volume de données a été de dégrader la recherche : seuls des *k*-mers sont indexés, ils sont indexés de manière probabiliste et la structure ne répond qu'aux requêtes d'existence. La structure de Solomon et Kingsford (2016) utilise 200 Go pour les 2 500 RNA-seq humains évoqués plus haut, soit environ 0,1 bit par nucléotide ou plus de 400 bits par *k*-mer distinct ($k = 20$). L'idée continuera à être exploitée et améliorée par plusieurs équipes (voir la synthèse de Marchet *et al.* (2021)). La structure la plus économe, HowDeSBT (Harris et Medvedev, 2020), une évolution de la structure de Solomon et Kingsford, utilise moins d'un dixième de la place requise par la structure d'origine. Les 2 500 jeux de données RNA-seq sont indexés en 15 Go [2], soit moins de 0,01 bit par nucléotide ou une trentaine de bits par 20-mer distinct. Cette idée d'indexation *via* des filtres de Bloom a également été exploitée avec succès pour l'indexation de données de séquençage microbien. Mes excellents collègues Camille Marchet et Antoine Limasset ont proposé une approche simplifiée, toujours à base de filtres de Bloom (Marchet et Limasset, 2023). La méthode ne permet pas de diminuer la taille de l'index mais de gagner nettement en ressources (temps et disque) nécessaires lors de la construction, ce qui leur a permis de construire leur index sur 32 768 RNA-seq humains soit plus de 15 % des RNA-seq humains disponibles publiquement sur SRA.

L'indexation par agrégation des filtres de Bloom de jeux de données n'est pas la seule approche qui existe. L'autre type d'approche consiste à indexer tous les *k*-mers ensemble, dans une même structure. Une information est ajoutée à ces *k*-mers, on dit alors qu'ils sont colorés,

1. Une quantification par *k*-mer unique serait plus pertinente, mais le nombre de *k*-mers distincts de leur jeu de données n'est pas communiqué.

2. Pour remettre les choses en contexte, c'est environ la place requise pour une « banale » table des suffixes sur un génome humain.

afin de savoir à quels jeux de données ils appartiennent. La structure de données correspond souvent, mais pas systématiquement, à un graphe de de Bruijn. Ce type d'approche a tendance à être un peu moins efficace en mémoire, mais à passer plus facilement à l'échelle pour la construction. Je montrerai en section 3.4 page 38 comment nous avons utilisé ce type de structure pour indexer des jeux de données en rendant possible des requêtes de comptage.

Enfin, en juillet, le NCBI a lancé le site PebbleScout permettant de rechercher dans différentes collections de jeux de séquençage à haut débit composées, pour les plus grandes, de quelques pétanucléotides (Shiryev et Agarwala, 2023). Méthodologiquement, l'approche est relativement basique. Il s'agit d'indexer des minimiseurs dans un arbre B+, mais en pratique l'avancée est majeure : pour la première fois, la recherche dans des millions de jeux de données stockés sur SRA est rendue possible pour tous et toutes. Pour reprendre les critères de Douglas *et al.* (2011), l'utilisation de ce site coche de nombreuses cases en terme d'accessibilité, d'utilité et de portabilité. Néanmoins, les requêtes sont limitées à l'existence de certaines séquences dans des jeux de données.

3.1.4 Indexation pour la recherche approchée

Les outils de mapping, sur un seul ou sur de multiples séquences de référence, recherchent les reads en tolérant des erreurs entre celles-ci et la région correspondante dans une des références. Pour réussir à trouver de telles occurrences, le plus simple serait intuitivement que la structure d'indexation soit capable d'énumérer toutes les occurrences d'un motif à ε erreurs près, dans le texte indexé. Si de telles structures d'indexation existent, elles ont l'inconvénient d'être exponentielles en ε en espace ou en temps de requête (Chan *et al.*, 2010).

Une autre approche est préférée : celle du filtrage. L'idée est d'utiliser une structure d'indexation qui ne permette qu'une recherche exacte. Grâce à elle, on pourra rechercher des facteurs de la séquence requêtée, c'est-à-dire certaines sous-chaînes de cette séquence. Même s'il y a des différences entre la séquence requêtée et la séquence de référence, certains facteurs des deux séquences seront identiques. Il s'agit donc d'une heuristique qui permet de trouver les occurrences, et même toutes si le taux d'erreurs est limité.

La recherche se produit en deux temps : dans un premier temps, il faut trouver les occurrences de certains facteurs de la requête ; dans un second temps, vérifier (généralement avec un algorithme de programmation dynamique) si ce qui précède ou suit chacune des occurrences est suffisamment similaire entre la requête et la référence (le principe *seed and extend*). Dans ce cas, inutile d'avoir une structure d'indexation capable de rechercher une séquence en tolérant des erreurs. Notons néanmoins que Bowtie (Langmead *et al.*, 2009), par exemple, tolère des erreurs directement dans la première phase. Pour ce faire, les auteurs n'utilisent pas une structure d'indexation atypique mais un FM-index classique. Ils se contentent d'énumérer exhaustivement toutes les possibilités à ε erreurs près et de les rechercher.

Sur ces thématiques de l'indexation, qui sont particulièrement actives dans notre domaine en raison de leurs implications, j'ai co-encadré deux thèses et un post-doctorat. La première thèse, réalisée par Christophe Vroland, avait pour but de trouver toutes les occurrences de courtes séquences (d'une vingtaine de nucléotides) dans des génomes complets, en autorisant jusqu'à trois erreurs, soit environ 15 % d'erreurs (voir section 3.2). La seconde thèse, réalisée par Tatiana Rocher, portait sur l'indexation combinée de séquences ADN et de méta-données qui y sont associées (voir section 3.3 page 36). Enfin le post-doctorat, réalisé par Camille Marchet, se plaçait dans le cadre d'un projet ANR qui visait à indexer des milliers de jeux de données de séquençage (voir section 3.4 page 38).

3.2 De nouvelles graines pour la recherche approchée

Comme indiqué précédemment, les structures d'indexation classiquement utilisées en bioinformatique, qu'il s'agisse de tables de hachage ou de transformée de Burrows-Wheeler (1994), ne donnent pas accès directement à l'ensemble des occurrences d'un mot à ε erreurs près. Ces structures sont adaptées à la recherche exacte de mots, mais devoir tolérer des erreurs lors de la recherche conduit à modifier ces algorithmes de recherche.

Dès les années 1990, le principe utilisé par Blast (Altschul *et al.*, 1990) était de découper la séquence à rechercher en blocs qui pourraient chacun être recherchés de manière exacte[3]. Ce principe est celui du pigeonnier (ou du tiroir à chaussettes, selon les passions de chacun·e), voir figure 3.4. En bioinformatique, il convient de remplacer les cases du pigeonnier par des blocs non chevauchants sur une séquence et les pigeons par des erreurs. Dans ce contexte, la propriété la plus intéressante est d'avoir une case vide[4], c'est-à-dire un bloc sans erreur. Il suffit pour cela que le nombre de cases soit strictement supérieur au nombre de pigeons. C'est-à-dire que pour rechercher une séquence avec ε erreurs, il suffit de la découper en $\varepsilon + 1$ blocs. Ce découpage suffira à garantir qu'au moins un bloc sera conservé et pourra être recherché sans erreur avec notre structure d'indexation favorite. Le principe du pigeonnier se place dans le cadre des stratégies *seed and extend*, où les graines peuvent être des séquences consécutives sur la séquence, c'est-à-dire des k-mers. Avec ce découpage en blocs, plus le taux d'erreurs augmente, plus le nombre de blocs doit également augmenter. Ces blocs deviennent donc de plus en plus courts, ce qui les rend peu spécifiques (pour filer la comparaison : plus un même pigeonnier contient de cases, plus ses cases sont petites et moins les pigeons peuvent y rentrer, ce qui les rend peu utiles).

FIGURE 3.4 – Un pigeonnier à neuf cases. Avec dix pigeons, il y a forcément au moins une case occupée par deux pigeons.

Lorsqu'il s'agit de rechercher de courtes séquences de vingt nucléotides avec un fort taux d'erreurs, par exemple des séquences de nucléotides avec 15 % d'erreurs (soit trois erreurs), nous nous trouvons justement dans la situation où les blocs obtenus seraient trop petits. Notre séquence devrait être découpée en quatre blocs, chacun de cinq nucléotides. Or, puisqu'il n'y a que 1 024 séquences de cinq nucléotides différentes, une telle séquence est trop peu spécifique pour pouvoir être recherchée directement dans un génome complet.

Dans une telle situation, les solutions envisageables ne sont pas pléthoriques. Des outils génériques de mapping, comme Bowtie, ne sont pas adaptés à de tels taux d'erreurs. Il existe en revanche des outils plus spécialisés, RazerS3 en est un exemple (Weese *et al.*, 2012). Son but est d'offrir une sensibilité parfaite, au contraire des heuristiques à la Bowtie qui ne peuvent le garantir. Ce logiciel ne s'appuie pas sur une séquence de référence préalablement indexée, ce qui ne l'empêche pas d'utiliser une stratégie de type *seed and extend*. L'étape d'identification de graines repose également sur le principe du pigeonnier. L'étape d'extension s'appuie sur une optimisation de l'algorithme de Myers, utilisant des vecteurs de bits. Cette méthode rend RazerS3 relativement efficace en dépit de sa capacité à identifier toutes les occurrences qui existent. Il existe également des approches reposant sur un FM-index. readaligner (Mäkinen *et al.*, 2010b) découpe le texte en k-mers et indexe chacun de ces k-mers, concaténés à eux-mêmes, dans une transformée de Burrows-Wheeler. Cette stratégie leur permet de rechercher directement, par exemple, le début d'un k-mer suivi de la fin de ce k-mer, et d'ensuite s'intéresser au milieu entre ce début et cette fin. Dans l'hypothèse où c'est le milieu du k-mer qui comporte des erreurs, cela permet de reporter la phase de prise en compte des erreurs le plus tardivement possible (afin d'éviter d'avoir trop de possibilités à considérer). La contrainte de cette approche est que la taille des motifs que l'on peut rechercher est fixée dès l'indexation : k. Enfin, on peut également s'appuyer sur des méthodes éprouvées reposant sur des approches « *seed and extend* » à la Blast, comme exonerate (Slater et Birney, 2005), où la phase d'extension est optimisée en élaguant les zones à considérer dans la programmation dynamique.

Cette problématique ne se pose pas uniquement pour l'intérêt théorique de la question : nous l'avons formulée dans le cadre de la thèse de Christophe Vroland (de 2012 à 2016) sur la recherche de cibles de micro-ARN dans des plantes, co-dirigée par Hélène Touzet (Bonsai) et Vincent Castric (unité Évolution, Écologie et Paléontologie, à Lille). Ces micro-ARN, une fois matures, sont de courtes séquences nucléiques d'une vingtaine de nucléotides qui en général s'apparient imparfaitement à des ARN messagers (leurs cibles). Du point de vue de l'algorithmique du texte, l'imperfection de l'appariement se modélise en autorisant des erreurs lors

3. Dans sa version nucléique, en tout cas.
4. Une fois n'est pas coutume…

de la recherche (Dai *et al.*, 2011). Bien entendu, cette simplification du problème n'est pas suffisante pour ne détecter que des cibles de micro-ARN, mais il s'agit d'une étape préalable, d'autres filtres pouvant être ensuite ajoutés pour raffiner les résultats (par exemple, à partir des positions des erreurs, de l'énergie d'hybridation, de résultats d'expériences de séquençage de complexes micro-ARN – ARN messager, etc.).

À partir de cette théorisation du problème, nous allons voir comment faire évoluer le principe du pigeonnier au cas de figure où le taux d'erreurs est important et tirer parti de structures d'indexation efficaces.

3.2.1 Adaptation du principe du pigeonnier

L'application du principe du pigeonnier permet de se ramener à la situation où une partie au moins de la séquence à rechercher est conservée. Cette partie conservée est généralement appelée une graine. L'ensemble des occurrences d'une telle graine ne correspondront pas nécessairement à des occurrences de notre séquence d'origine à ε erreurs près : toutes les graines ne germent pas. Il s'agit en réalité d'un filtre : l'emploi d'une telle graine évitera de considérer de nombreuses portions du génome. Appliqué tel que décrit précédemment, le principe du pigeonnier est un filtre exact dans le sens où il ne produit pas de faux négatif. Mais une autre caractéristique importante du filtre est son *pouvoir filtrant*, c'est-à-dire la précision du filtre : le ratio entre le nombre d'occurrences de la séquence complète à ε erreurs près et le nombre d'occurrences totales de l'ensemble des graines possibles. Le problème auquel nous nous intéressons est le suivant :

Nous garderons en tête que, dans cette section, « courte séquence » correspond à une séquence d'une vingtaine de nucléotides et que nous visons $\varepsilon = 3$

Problème

Identifier un filtre exact ayant un meilleur pouvoir filtrant que celui offert par le principe du pigeonnier, afin de trouver toutes les occurrences d'une courte séquence avec au plus ε erreurs.

Afin de proposer une solution à ce problème nous allons nous intéresser à la situation où une séquence quelconque n'est pas découpée en $\varepsilon + 1$ blocs, mais en $\varepsilon + 2$ blocs. Bien que le changement paraisse mineur, il fait apparaître des propriétés intéressantes que nous allons exploiter.

Par définition, lorsqu'une séquence est séparée en $\varepsilon + 2$ blocs et que ε erreurs sont acceptées, il y a nécessairement au moins deux blocs qui n'ont pas d'erreur. Au-delà de cette propriété évidente, la répartition des erreurs garantit de trouver systématiquement au moins un bloc avec 0 erreur suivi par un nombre indéfini de blocs avec exactement 1 erreur, suivis par un bloc avec 0 erreur. Une telle suite de blocs sera appelée graine 01^*0 (voir la figure 3.5 page suivante pour des exemples). Plus formellement, soit P un mot de longueur m découpé en $p = \varepsilon + 2$ blocs, et $P[0], \ldots, P[p-1]$ chacun de ces blocs. Une graine 01^*0 correspond à une suite de blocs $P[i], \ldots, P[j]$ avec $0 \leq i < j < p$, où le bloc $P[i]$ contient 0 erreur, chacun des blocs $P[i']$ contient exactement une erreur, avec $i < i' < j$, et le bloc $P[j]$ contient 0 erreur.

3.2.2 Avantages et inconvénients des graines 01^*0

Le premier avantage des graines 01^*0 est de contraindre l'espace de recherche. Par rapport à une application classique du principe du pigeonnier, où une seule partie avait 0 erreur, les graines 01^*0 garantissent d'avoir deux blocs avec 0 erreur tout en contraignant le nombre d'erreurs entre ces deux blocs. En effet, en appliquant uniquement le principe du pigeonnier, il est possible de rechercher un bloc avec 0 erreur immédiatement suivi par un bloc avec 3 erreurs, ce qui fait exploser l'espace de recherche. Avec des graines 01^*0, ce n'est plus le cas. Indépendamment du nombre d'erreurs recherché (ε), un bloc à 0 erreur sera forcément suivi par un autre bloc à 0 erreur (cas 00) ou par une série de blocs avec 1 erreur suivis par un bloc à 0 erreur (cas 01^+0).

D'autre part, la définition des graines 01^*0 peut encore être raffinée pour contraindre également le nombre d'erreurs qui précèdent et suivent la graine. Dans la figure 3.5 page suivante, lorsqu'il y avait plusieurs possibilités, nous avons placé les graines 01^*0 naïvement, c'est-à-dire en les mettant les plus à gauche. À la place, nous contraignons la définition de ces graines pour limiter l'espace de recherche. Ainsi si les blocs $P[i], \ldots, P[j]$ correspondent à une graine 01^*0, alors les blocs $P[0], \ldots, P[i-1]$ totalisent au plus i erreurs. Par exemple, dans la répartition suivante des erreurs | 2 | 0 | 0 | 0 | 1 |, la graine 01^*0 ne sera pas constituée des deux

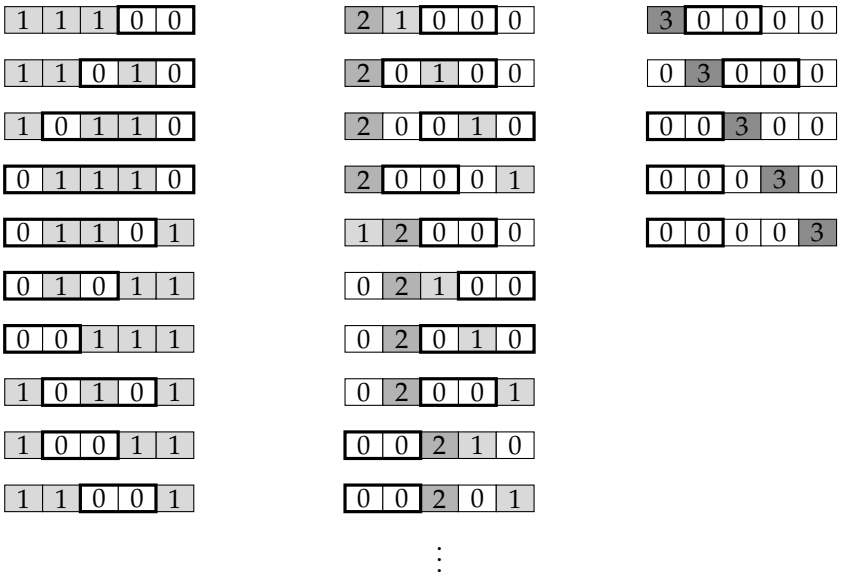

FIGURE 3.5 – Répartition possible de trois erreurs parmi cinq blocs. Pour la colonne du milieu, où un bloc contient deux erreurs, seule une moitié des répartitions possibles a été représentée. Il existe également les répartitions symétriques. Sont encadrés en gras les blocs correspondant à un schéma 01^*0. Dans certaines situations, en particulier dans la colonne de droite, il existe plusieurs possibilités pour avoir une suite de blocs du type 01^*0, seul le plus à gauche a été mis en évidence.

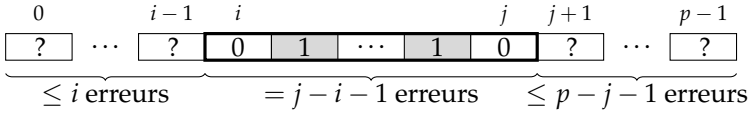

FIGURE 3.6 – Répartition des erreurs dans une séquence découpée en p blocs, à partir de l'occurrence d'une graine 01^*0. Par définition la graine 01^*0 contient exactement $j - i - 1$ erreurs (1 erreur dans chaque bloc intermédiaire). Les blocs qui précèdent et qui suivent ont au maximum autant d'erreurs qu'il y a de blocs.

premiers blocs à 0 erreur (puisqu'ils sont précédés par un seul bloc avec 2 erreurs), mais par les deux derniers blocs à 0 erreur (qui sont bien précédés par deux blocs totalisant 2 erreurs). Cette contrainte est symétrique, et donc les blocs $P[j+1], \ldots, P[p-1]$ contiennent au plus $p - j - 1$ erreurs. L'ensemble de ces contraintes sont résumées dans la figure 3.6. L'ajout de ces contraintes garantit une unicité de la graine 01^*0 pour une répartition donnée des erreurs, lorsque le nombre d'erreurs est maximal. Cette répartition des erreurs peut désormais être utilisée afin d'optimiser la manière de rechercher les occurrences via un index.

3.2.3 Indexation pour l'emploi de graines 01^*0

Idéalement, la recherche d'une graine 01^*0 se ferait par ses deux extrémités, qui doivent être exactes. Néanmoins, à l'heure actuelle et à ma connaissance, il n'existe pas de structure d'indexation de texte intégral qui permette de rechercher deux séquences séparées par une longueur bornée. Sans une telle structure d'indexation, il faut alors commencer par une des extrémités de la graine 01^*0 (sans tolérer d'erreur) et passer aux blocs adjacents, en autorisant au maximum une erreur. Si un bloc est trouvé sans erreur, c'est donc la fin de la graine 01^*0, qui est délimitée par deux blocs sans erreur. Pour les occurrences avec erreurs, la recherche des blocs adjacents se poursuit avec au moins une erreur jusqu'à tomber sur un bloc sans erreur. Ici, c'est bien dans la structure d'indexation que la recherche avec erreurs est (en partie) effectuée, de la même manière que ce qui est fait pour Bowtie (Langmead *et al.*, 2009).

Une fois les occurrences des graines 01^*0 identifiées, il reste à procéder à l'étape d'extension. Là aussi, grâce aux propriétés des graines 01^*0, le nombre d'erreurs pendant la phase

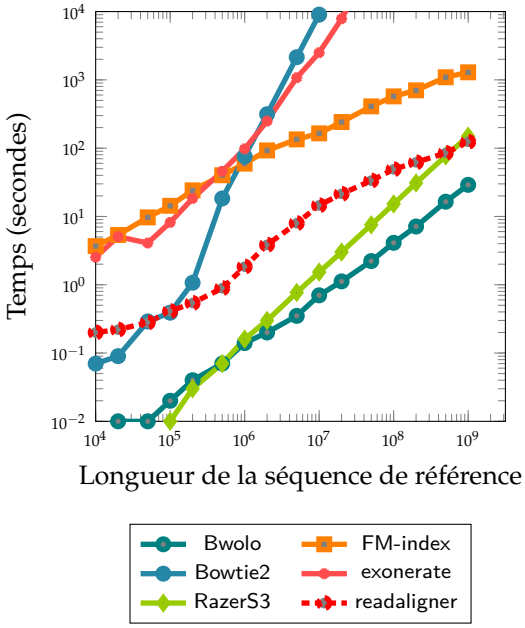

FIGURE 3.7 – Temps pour rechercher 100 séquences de taille 20 avec 3 erreurs dans des séquences de longueur 10^4 à 10^9. Toutes les séquences ont été générées pseudo-aléatoirement. Figure issue de Vroland *et al.* (2016).

d'extension est contraint, ce qui rend plus efficace la phase d'alignement par programmation dynamique. Notons qu'avec une structure d'indexation bidirectionnelle, permettant de rechercher une séquence aussi bien de la gauche vers la droite que dans l'autre direction, et permettant de changer de direction en cours de recherche, il serait possible de faire la phase d'extension exclusivement dans l'index. En effet, une fois trouvées les occurrences des graines 01^*0, il suffirait de poursuivre la recherche dans l'index pour rechercher ce qui précède et ce qui suit la graine.

3.2.4 Mise en œuvre et résultats

Cette méthode a été mise en œuvre en C++ par Christophe Vroland, en s'appuyant sur la bibliothèque SeqAn (Döring *et al.*, 2008), dans un prototype sous licence libre appelé bwolo. Elle a été publiée en conférence (Vroland *et al.*, 2014) et, en version étendue, en journal (Vroland *et al.*, 2016). La structure d'indexation utilisée est un FM-index, afin de limiter l'espace mémoire nécessaire à l'index car le but est d'utiliser cette approche sur un génome de référence, pouvant donc faire quelques giganucléotides. En théorie, nous aurions pu utiliser un FM-index bidirectionnel. Néanmoins, cela doublerait l'espace mémoire requis par la structure d'indexation (il faut indexer le texte dans les deux directions), pour un gain en temps relativement modeste en pratique.

Nous avons comparé bwolo aux autres logiciels mentionnés précédemment : Bowtie2 (dont je rappelle qu'il n'est pas adapté à cette tâche), RazerS3, exonerate, readaligner, ainsi qu'une version naïve d'un FM-index dans laquelle on recherche exhaustivement les erreurs. Cette comparaison a notamment été faite sur des séquences aléatoires et montre que bwolo est systématiquement plus rapide pour rechercher des séquences de longueur 20 avec 3 erreurs lorsque la séquence de référence est grande (voir figure 3.7).

3.2.5 Impacts des recherches

Cette thèse étant co-encadrée avec l'unité Évolution, Écologie et Paléontologie, le but n'était pas uniquement de proposer une solution pour la recherche de courtes séquences avec de forts taux d'erreurs mais également d'adapter cette solution à la recherche de cibles de micro-ARN. La difficulté, pour ce type de thèse, est d'arriver jusqu'à proposer un outil bioinformatique

répondant aux besoins des biologistes, qui puisse ensuite être valorisé. Dans cette thèse, nous n'avons pas pu aller jusqu'à cette étape par manque de temps.

Simultanément à notre publication en conférence, Kucherov *et al.* (2014) ont publié une méthode consistant à prioriser les blocs à rechercher pour de la recherche approchée (autorisant uniquement les substitutions) dans un index bidirectionnel. Suite à cela, l'équipe de Knut Reinert à l'université libre de Berlin a proposé un schéma optimal de recherche approchée (Kianfar *et al.*, 2017) s'inspirant des travaux de Kucherov *et al.* et montrant que notre approche était proche de l'optimale.

Sur l'aspect bioinformatique, les graines 01^*0 ont été intégrées dans miRkwood, un outil pour la recherche de cibles de micro-ARN dans les génomes de plantes, créé à Lille sous la supervision d'Hélène Touzet (Guigon *et al.*, 2019).

3.3 Indexation de recombinaisons V(D)J

En lien avec le projet Vidjil, la question de l'indexation des milliers de jeux de données que nous traitons se pose afin de répondre à des problématiques biologiques, ou même cliniques de suivi des patients, comme expliqué en section 2.2.7 page 19.

Avec Mathieu Giraud, nous avons proposé un sujet de thèse destiné à indexer des recombinaisons V(D)J. L'enjeu est à la fois d'indexer les séquences nucléiques de ces recombinaisons, ce que l'on sait faire, et d'indexer en même temps des informations positionnelles sur ces séquences comme le fait que tel gène V est présent de la position x à y de la séquence. De manière plus générale, nous cherchons à indexer des séquences annotées, en ajoutant la contrainte qu'une position ne peut avoir au plus qu'une seule annotation. Tatiana Rocher a travaillé sur ce sujet de thèse de 2014 à 2017.

Soit un texte T de longueur n et A un vecteur de taille n contenant les annotations de T, tel que $A[i]$ est l'annotation de $T[i]$. Un facteur f de T, apparaissant en position i, est annoté avec x si et seulement si $A[i] = x$.

Problème

> *Indexer un texte T et son vecteur d'annotations A de manière à pouvoir répondre à des requêtes associant des éléments de T et de A, avec une complexité en temps asymptotiquement meilleure qu'une approche naïve ou avec un espace mémoire inférieur à $|T| + |A|$.*

Indexer une séquence annotée signifie indexer T et A ensemble de manière à répondre à un ensemble de requêtes en un temps sous-linéaire :

1. Quelles sont les occurrences d'un motif P dans T ?
2. Quelle est l'annotation de $T[i]$?
3. Combien de fois l'annotation x est-elle utilisée ?
4. Quelles sont les occurrences de x dans A ?
5. Combien de fois la séquence P est-elle annotée avec x ?
6. Quelles sont les occurrences de P annotées avec x ?

Les quatre premières requêtes sont triviales avec des structures d'indexation pour le texte que nous avons l'habitude d'utiliser : une table des suffixes ou une transformée de Burrows-Wheeler suffirait pour T et pour A une table de hachage conviendrait (Lecroq et Salson, 2022). Le cœur de la thèse de Tatiana a été de se concentrer sur les deux dernières requêtes qui allient des informations de T avec des informations de A. Notons que nous considérons un problème un peu simplifié, où nous recherchons les occurrences de P dont la première lettre est annotée avec x, ce qui évite d'avoir à traiter les situations dans lesquelles seule une partie de P est annotée avec x. Comment associer le texte et le vecteur d'annotations de manière à optimiser l'espace utilisé ou le temps de requête ?

La solution la plus évidente pour indexer le texte est de passer par un FM-index (voir section 3.1.1 page 27). Nous avons en effet une quantité de données (des dizaines de milliers de jeux de données) qui justifie de passer par une structure compressée. D'autre part, les annotations seront indexées dans un *wavelet tree* (Ferragina *et al.*, 2009), ce qui garantit de n'utiliser qu'un espace succinct tout en ayant un accès aux annotations en un temps qui est logarithmique dans le nombre d'annotations différentes (voir section 3.1.1 page 27, où le *wavelet tree* est détaillé).

Il reste à pouvoir passer d'une structure à l'autre, afin que des occurrences dans le texte, identifiées à l'aide du FM-index, puissent trouver une correspondance dans le *wavelet tree*. Le FM-index s'appuie sur une transformée de Burrows-Wheeler qui a pour effet de réarranger les lettres du texte. Les positions des annotations dans le texte ne correspondent donc pas aux positions dans la transformée. Afin de simplifier ce passage d'une structure à l'autre, nous réordonnons les annotations dans un tableau imaginaire A', de manière à ce que leurs positions correspondent à celles dans la transformée de Burrows-Wheeler. D'autre part, les annotations ayant une sémantique liée à leur séquence — au moins dans notre application aux recombinaisons V(D)J — les mettre dans l'ordre de la transformée peut avoir pour effet de rassembler des annotations identiques. De cette façon, les annotations restent liées aux lettres auxquelles elles sont attachées (voir figure 3.8 page suivante). Ainsi, connaître l'annotation d'une occurrence dans la transformée de Burrows-Wheeler à la position i consiste uniquement à consulter $A'[i]$.

Plutôt que de stocker explicitement A', ce qui serait inefficace étant donné que nous nous attendons à avoir de nombreuses annotations identiques consécutives, un vecteur de bits B stocke l'information de l'identité entre deux annotations consécutives. Nous avons $B[i] = 0$ si et seulement si $A'[i] = A'[i-1]$, pour $0 < i < |A'|$, et $B[0] = 1$. Les annotations $A'[i]$ telles que $B[i] = 1$ sont ensuite indexées à l'aide du *wavelet tree* (voir figure 3.8 page suivante).

Lors d'une requête combinée, où on cherche toutes les séquences P annotées avec x, le FM-index permet d'identifier l'intervalle $[d, f]$ de la transformée de Burrows-Wheeler dans lequel se situent toutes les occurrences de P. Ensuite, il suffit d'identifier le nombre de 1 dans B jusqu'à la position $B[d]$ (noté u_d) et jusqu'à la position $B[f]$ (noté u_f). Il suffit ensuite de récupérer dans le *wavelet tree*, parmi les positions comprises entre $u_d - 1$ et $u_f - 1$, celles qui correspondent à l'annotation x, ce qui se fait avec un aller-retour dans l'arbre de la racine à la feuille qui correspond à l'annotation x. La remontée permet d'identifier les intervalles éventuellement disjoints qui correspondent à l'annotation x. La complexité de la recherche est en $\mathcal{O}(|P| + occ_P \log^{1+\varepsilon} n + occ_{P,x} \log a)$, avec $\varepsilon > 0$, occ_P le nombre d'occurrences de P dans T, $occ_{P,x}$ le nombre d'occurrences de P dont la première lettre est annotée x et a le nombre d'annotations distinctes. Le facteur $\log^{1+\varepsilon} n$ est dû au temps nécessaire pour identifier la position de chaque occurrence dans le FM-index. La question se pose néanmoins de savoir si une solution qui ne dépendrait pas de occ_P est envisageable. En effet, si la proportion d'annotations x est très faible parmi les occurrences de P, nous récupérons les positions de nombreuses occurrences dans T qui se révèlent inutiles *a posteriori*.

3.3.1 Mise en œuvre et résultats

Cette structure de données a été codée en C++ par Tatiana Rocher, en s'appuyant sur la bibliothèque SDSL-lite (Gog *et al.*, 2014), et mise à disposition sous licence libre sur un dépôt Git [5] et publiée (Rocher *et al.*, 2018). Dans les expériences conduites, nous avons montré que la requête combinée était entre un et deux ordres de grandeur plus rapide que des approches plus naïves.

3.3.2 Impacts des recherches

Cette capacité à rechercher des séquences ADN combinées à leur annotation n'a malheureusement pas rencontré l'écho escompté. Dans le cadre de Vidjil, nous avions interrogé les biologistes utilisant Vidjil à propos de l'intérêt qu'il pourrait y avoir à rechercher dans leurs données de séquençage. Globalement, nos collègues étaient enthousiastes à l'idée de réinterroger ces données, ou alors d'interroger d'autres jeux de données (publics ou partagés entre différents hôpitaux).

Pour répondre à leur besoin, avant que la thèse de Tatiana soit achevée, nous avons mis en place un prototype, à l'aide de BWA (Li et Durbin, 2009), qui se contente de rechercher dans les jeux de données de chaque laboratoire. Le prototype n'a pas connu l'engouement attendu, les requêtes y sont rarissimes bien que nous ayons développé des fonctionnalités afin de rendre son interrogation facile à travers l'interface de Vidjil.

Un interfaçage relativement simple ne suffit pas. Sur ce projet, je considère que nous avons manqué de collaborations actives avec des laboratoires concernés. Cela aurait permis d'une

5. https://gitlab.inria.fr/vidjil/tl-index/

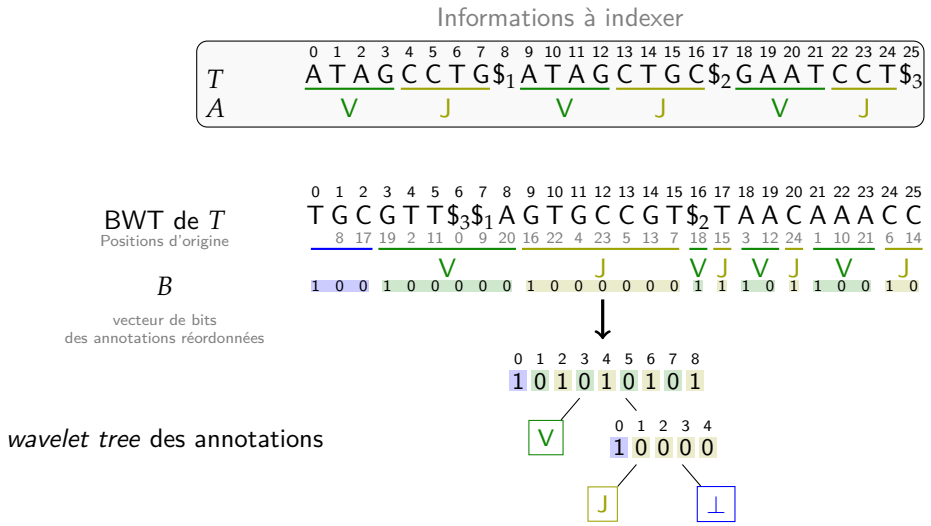

FIGURE 3.8 – Trois séquences sont indexées (T), chacune avec ses annotations V et J. À partir de T, on calcule la transformée de Burrows-Wheeler. Les annotations sont alors transférées aux positions correspondantes. À titre d'information, les positions d'origine de chaque annotation sont indiquées (les lettres ne correspondent pas à la position d'origine car la transformée de Burrows-Wheeler est construite sur la dernière lettre de chaque permutation circulaire, là où dans T c'est la première lettre qui est représentée). Les intervalles d'annotations identiques sont représentés par une suite de bits B commençant par un 1, et des 0 pour les positions suivantes possédant la même annotation. Un *wavelet tree* permet de retrouver l'annotation de chaque position. L'association de B et du *wavelet tree* servent à indexer la suite d'annotations sur la transformée de Burrows-Wheeler.

part de construire ensemble une solution qui réponde à leur besoin mais cela aurait également permis de les engager plus fermement dans ce projet qui, pour l'instant, peut ressembler à une fonctionnalité un peu vague dont ils n'ont qu'une connaissance parcellaire.

À la suite d'une réunion d'usagers de Vidjil en mai 2023, nous comptons lancer une collaboration plus appuyée avec différents laboratoires intéressés afin d'étudier comment nous pourrions exploiter les dizaines de milliers de jeux de données qu'ils ont déjà séquencés.

3.4 Indexation de jeux de séquençage

Lors de mon travail sur le logiciel CRAC avec mes collègues à Montpellier, nous avons conçu une structure d'indexation pour une collection de reads : Gk-arrays (Philippe *et al.*, 2011). Cette solution, permettant de retrouver les occurrences de n'importe quel k-mer dans un jeu de reads, ne passe pas à l'échelle de plusieurs jeux de reads. Ce n'était d'ailleurs pas son ambition.

Néanmoins, au fur et à mesure de notre collaboration avec l'équipe de Thérèse Commes, à l'*Institute for Regenerative Medicine and Biotherapy* de Montpellier, nous avons acquis la conviction que l'accès et la recherche dans des dizaines, centaines, voire milliers de jeux de données de séquençage serait une fonctionnalité extrêmement importante, notamment afin de confirmer l'existence d'événements rares sur de grandes cohortes (par exemple certains transcrits de fusion).

3.4.1 Comparaison différentielle de jeux de séquençage

Avec l'équipe de Thérèse Commes et celle de Daniel Gautheret (Institut de biologie intégrative de la cellule, Paris-Saclay), nous avons commencé par proposer une approche pour comparer deux ensembles de jeux de données et identifier ce qui différencie les uns par rapport aux autres.

Problème

À partir d'un ensemble de jeux de séquençage contrôle et expérimentaux, identifier les séquences significativement différentiellement exprimées entre les deux conditions.

Le principe est relativement simple. Il serait évidemment possible de réaliser cela avec des méthodes d'alignement qui quantifient le niveau d'expression. Ainsi que mentionné dans l'introduction, des approches sans alignement ont aussi été développées pour la quantification d'ARN. Néanmoins, si ces approches ne font pas de l'alignement au nucléotide près, elles font néanmoins un alignement « grossier » afin de pouvoir déterminer de quel transcrit provient la séquence en question. Cette nécessité de comparer les séquences à une référence peut poser problème : certaines séquences s'alignent difficilement, par exemple parce que ce sont des régions de faible complexité, ou parce que ce sont des régions fortement mutées. Si des différences d'expression existent pour des séquences qu'on n'arrive pas à aligner, elles ne pourront pas être détectées. Notre approche a plutôt été de ne faire aucun alignement (même grossier) et de se restreindre à un comptage de k-mers :

1. nous comptons les k-mers séparément dans les deux conditions ;
2. nous filtrons les k-mers erronés (faible occurrence) ou de référence (afin de retirer les événements déjà probablement connus) ;
3. nous ne conservons que les k-mers différentiellement exprimés entre les deux conditions ;
4. nous assemblons ces k-mers différentiellement exprimés afin de constituer des contigs un peu plus longs pour les aligner et les annoter.

Cet outil, appelé DE-KUPL, a été mis à disposition sous licence libre sur un dépôt Git [6] par Jérôme Audoux, alors doctorant à Montpellier, et publié (Audoux *et al.*, 2017). Il a servi de preuve de concept pour une demande de financement ANR afin de développer une structure d'indexation pour des milliers de jeux de reads. Ce projet, appelé Transipedia et dans lequel j'étais impliqué avec Rayan Chikhi, qui était alors chercheur dans notre équipe, a été financé et a donné lieu au travail décrit dans la section suivante.

3.4.2 Index pour la quantification de séquences dans des collections de jeux de reads

Une limite importante des structures d'indexation de jeux de reads qui ont été présentées dans l'introduction de ce chapitre (voir section 3.1.3 page 30) tient au type de requêtes qu'elles permettent. Ces structures sont capables d'indiquer la présence ou l'absence de k-mers dans des jeux de données, éventuellement de manière probabiliste. Or, le projet Transipedia porte sur des données de séquençage transcriptomique (RNA-seq), dans lesquels le niveau d'expression des gènes est une information fondamentale. D'ailleurs, une approche à la DE-KUPL montre bien l'intérêt de connaître la quantification puisque ce sont des différentiels de quantification qui sont sélectionnés, pas seulement des différentiels de présence. Cette différence permet de caractériser des événements transcriptomiques pertinents.

Dans le cadre de ce projet nous avons recruté Camille Marchet en post-doctorat pour travailler sur une structure d'indexation de jeux de reads et qui a commencé par réaliser une synthèse des approches existantes (Marchet *et al.*, 2021).

Problème

Avoir une structure d'indexation qui réponde à la fois à des requêtes sur l'existence et le comptage de k-mers dans des milliers de jeux de reads RNA-seq.

L'approche qu'elle a conçue pour indexer des jeux de reads, en répondant à la problématique posée, a été la suivante (voir figure 3.9 page suivante) :

1. construire un graphe de de Bruijn (avec comptage) pour chaque jeu de reads ;
2. identifier les *monotigs* de l'union (implicite) des graphes de de Bruijn ;
3. associer les *monotigs* à un vecteur de comptage des abondances dans chaque jeu de données.

Nous définissons un *monotig* comme un chemin dans un graphe de de Bruijn, dont chaque k-mer partage le même minimiseur [7] et le même vecteur de comptages.

6. https://github.com/Transipedia/dekupl

7. Un minimiseur d'un k-mer est le plus petit k'-mer ($k' < k$) selon une fonction de hachage donnée. Il s'agit de sélectionner une partie d'un k-mer de manière déterministe mais arbitraire.

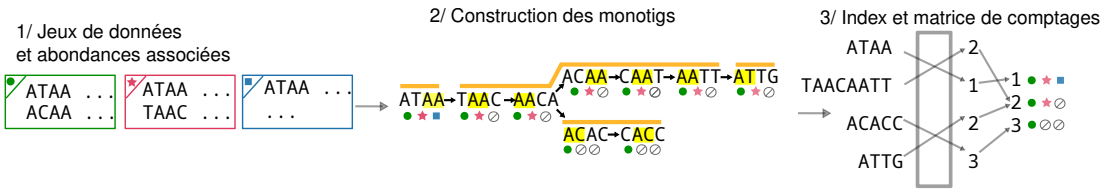

FIGURE 3.9 – Indexation de jeux de données de séquençage. 1. Nous avons trois jeux de données, dont les abondances sont supposées uniformes dans chacun et représentées par des symboles (● ★ ■). 2. Les graphes de de Bruijn de chaque jeu de données sont (implicitement) fusionnés, et les monotigs y sont identifiés par des traits orange. 3. Chaque monotig est associé à une ligne dans la matrice des comptages (plusieurs monotigs peuvent être associés à une même ligne). Figure traduite de Marchet *et al.* (2020).

Ensuite lors d'une requête avec une séquence Q, il suffit d'itérer sur les k-mers de Q, d'identifier pour chacun son monotig et de récupérer le vecteur de comptages correspondant. La requête ne retourne pas un vecteur de comptages unique mais un vecteur de comptages par monotig. Ensuite, c'est à la personne utilisant la structure de définir si elle préfère obtenir la moyenne, la médiane, le maximum, des comptages obtenus (ce qui dépendra notamment de la question posée). Cela en fait la première structure d'indexation pour des jeux de reads à répondre aux requêtes de comptage.

3.4.3 Mise en œuvre et résultats

La structure d'indexation, REINDEER, a été codée en C++ par Camille Marchet, mise à disposition sous licence libre sur un dépôt Git [8], et publiée (Marchet *et al.*, 2020).

Avec REINDEER, Camille a indexé la collection habituellement utilisée dans l'évaluation des autres structures d'indexation. Cette collection est composée de plus de 2 500 jeux de données RNA-seq rassemblant près de 14 téranucléotides. L'index, avec les quantifications, consomme 52 Go, soit 0,03 bit par nucléotide ou 110 bits par k-mer unique ($k = 21$). L'index de REINDEER est plus gourmand que la plus frugale des structures existantes (HowDeSBT arrive à une trentaine de bits par k-mer distinct), mais REINDEER n'est pas probabiliste et permet des requêtes de comptage.

3.4.4 Impacts des recherches

REINDEER a suscité l'intérêt de la communauté. Nous avons également fait en sorte de rendre cet index exploitable facilement. Rayan Chikhi a mis en place, au début de la pandémie de Covid-19, une page web pour rechercher des séquences dans les génomes de SARS-CoV-2 connus. À Montpellier, l'équipe a créé des index sur divers RNA-seq publics, notamment 1 019 jeux provenant de CCLE [9], et a mis en place un site web [10] pour faciliter l'exploitation de l'index. Depuis, le projet ANR se poursuit avec un nouveau financement jusqu'en 2026 (FullRNA). Pour la partie indexation, le but de ce projet sera de faire passer REINDEER à l'échelle pour indexer un ou deux ordres de grandeur de plus.

3.4.5 Pistes de recherche

Depuis la publication de REINDEER, une autre structure a été publiée afin d'indexer des jeux de données de séquençage tout en permettant des requêtes de comptage (Karasikov *et al.*, 2022). Cette structure affirme gagner un facteur cinq en espace par rapport à REINDEER. Dans le mode de requête portant sur l'existence uniquement, selon les expériences des auteurs, la structure utilise même moins d'espace que HowDeSBT.

De plus, leur approche est flexible sur le type d'annotations à stocker en lien avec les k-mers. L'intérêt est qu'on peut stocker des comptages, afin de répondre aux requêtes correspondantes, mais il est également possible de stocker, à la place, la localisation des k-mers afin

8. https://github.com/kamimrcht/REINDEER/
9. Cancer Cell Line Encyclopedia, des lignées cellulaires cancéreuses très étudiées dont on possède notamment de nombreuses données de séquençage à haut débit.
10. https://transipedia.org/

de répondre aux requêtes de localisation. Il s'agit du premier index à permettre des requêtes de localisation sur des milliers de jeux de données. C'est une avancée significative mais qui ne résout pas tout.

Nous pensons avoir des marges d'amélioration significatives sur REINDEER, qui méritent d'être explorées.

Séparer les données

La pertinence de mettre des centaines de milliers (ou — rêvons un peu — des millions) de jeux de données dans une même structure d'indexation pose question. Plus le nombre de jeux de données est important, plus on risque d'avoir un mélange disparate de données. En terme de compression, mais aussi en terme de requêtes, il est plus facile de gérer des données homogènes qu'hétérogènes. Séparer les données, en fonction de leur contenu en k-mers et de la co-occurrence de ces k-mers, garantirait d'avoir plusieurs index, chacun plus petits et plus homogènes. L'homogénéité facilite la compression et offre également la possibilité que lors d'une requête un ensemble restreint d'index soit sollicité.

Compresser plus efficacement les quantifications

Le stockage des quantifications pourrait être optimisé. Comme indiqué précédemment, chaque monotig est associé à un vecteur de comptages. Tous les comptages sont donc représentés dans une matrice où chaque ligne correspond à un monotig et chaque colonne correspond à un jeu de données.

Or, ni l'ordre des lignes, ni l'ordre des colonnes n'importe. Plutôt que de prendre un ordre arbitraire, cet ordre pourrait être choisi afin d'améliorer la compression de données. Concernant le réordonnement des lignes, il s'agit de tirer bénéfice du fait que des comptages de monotigs peuvent être très similaires. Nous nous attendons logiquement à avoir des k-mers co-occurrents. Ils sont déjà traités uniformément lorsqu'ils sont dans un même monotig. Mais cette co-occurrence peut également se produire au-delà de ces monotigs. Par exemple, hors épissage alternatif, on s'attend à ce que la quantification des k-mers (ou des monotigs) des exons d'un même transcrit soit corrélée (sans être uniforme, notamment en raison de la variation de couverture de séquençage). Aussi, nous pourrions rassembler dans la matrice de comptage les lignes les plus similaires, afin que les comptages d'une ligne puissent être exprimés en fonction de ceux de la ligne précédente.

Concernant les colonnes, le principe reste similaire : il s'agirait de les réordonner afin que les jeux de données les plus similaires soient consécutifs. Cela permettrait d'exprimer une colonne (ou une partie) en fonction d'une autre.

Là également, la séparation en index plus homogènes pourrait aider à ce que les lignes et colonnes des matrices de comptages soient globalement plus similaires et donc, puissent plus facilement être décrites en fonction de celles qui précèdent.

Dépasser les monotigs

Actuellement, REINDEER repose sur le concept de monotig. Or, plus le nombre de jeux croît, plus la définition d'un monotig est contraignante puisqu'elle implique une identité de quantification sur tous les jeux de données. La séparation des données en paquets cohérents distincts vise à éviter cette limitation. D'autre part, la contrainte que tous les k-mers d'un monotig partagent le même minimiseur empêche d'avoir des séquences très longues, même si cela simplifie les requêtes. Il y a donc une réflexion à mener sur d'autres types de séquences à indexer plutôt que des monotigs. Relâcher les définitions, pour rassembler des comptages similaires et non identiques et pour ne pas imposer la conservation d'un minimiseur peut sembler bénéfique. Néanmoins, il ne faut pas perdre de vue que cela induit aussi du stockage supplémentaire, afin de savoir comment les quantifications varient entre les comptages rassemblés sous un même chapeau, et une complexification des requêtes s'il n'y a plus un point d'accroche commun (comme un minimiseur). Des évaluations devront être menées afin de définir si les avantages dépassent les inconvénients. Cela fera notamment partie du travail que débute un post-doctorant en septembre 2023, co-encadré avec Camille Marchet.

Chapitre 4

Diffusion de la culture scientifique

En tant qu'enseignant-chercheur, je considère avoir une responsabilité dans la transmission de la culture scientifique [1]. Cela fait partie de ce que l'on appelle parfois, un peu pompeusement, la *responsabilité sociale du savant*. En effet, au sortir de la deuxième guerre mondiale, et de l'utilisation de la bombe atomique, une partie du monde scientifique a pris conscience de sa responsabilité à informer voire alerter le public le cas échéant (Laurens, 2019). Il reste néanmoins des réticences, car pour certains ce serait contradictoire avec la « neutralité » qu'on attendrait des scientifiques.

Cette volonté de diffusion de la culture scientifique correspond à une conviction de ma part qu'une connaissance plus affûtée de certaines savoirs spécifiques d'une part, ou de grands principes scientifiques d'autre part serait de nature à permettre d'éclairer le débat démocratique. Il ne s'agit pas pour autant de tomber dans la vision naïve selon laquelle il suffirait d'éduquer le peuple pour que celui-ci, illuminé par les connaissances qu'on lui apporterait, se comporte de manière rationnelle (ce qui signifie, le plus souvent, conforme à nos attentes). Cette vision est celle dite du *information-deficit model*, qui est désormais battue en brèche. Ce modèle du déficit d'information part de l'hypothèse que la croyance dans des thèses loufoques ou la non adhésion à des connaissances éprouvées (comme « *le réchauffement climatique est réel et l'humanité en est responsable* ») est due à un déficit d'informations. Il suffirait donc de combler ce déficit pour résoudre ces problèmes. Comme le montrent divers travaux de psychologie, ce n'est évidemment pas suffisant (Ecker *et al.*, 2022). Nous ne sommes pas des vases qu'il suffit de remplir de connaissances. Nous avons des affects, des croyances, des vécus qui jouent un rôle dans nos raisonnements et qui interfèrent avec notre adhésion à telle ou telle hypothèse. Si une information rentre en conflit trop direct avec ce qu'on est, ce qu'on croit viscéralement, il y a un fort risque qu'on se retrouve dans une situation de dissonance cognitive qui nous la fasse rejeter (Drummond et Fischhoff, 2017). Le paradoxe étant qu'avoir une bonne culture scientifique donne justement des outils permettant de trouver un prétexte pour rejeter une information dérangeante. Sur des sujets sensibles (comme le dérèglement climatique, par exemple), apporter de l'information ne suffira pas à entraîner des modifications de croyance et, *a fortiori*, de comportement. Par exemple, des sondages conduits par l'Ademe (l'Agence de l'environnement et de la maîtrise de l'énergie) de 2000 à 2021 montrent de manière régulière que les personnes ayant un diplôme universitaire « scientifique » adhèrent moins à l'origine humaine du changement climatique que les personnes ayant un diplôme universitaire « non-scientifique » [2] (Boy et Conseil, 2022). Dans une étude internationale dans 24 pays, le déni de l'origine humaine du changement climatique n'était pas corrélé à la culture scientifique (Rutjens *et al.*, 2021). En revanche, il existait une corrélation entre ce déni et le conservatisme politique, illustrant le rôle des convictions pré-existantes dans le rejet de certaines informations. Au-delà de ces verrous, dont il faut avoir conscience, les besoins de connaissances scientifiques autour des enjeux contemporains me paraissent majeurs. Y apporter ma contribution, à la hauteur de mes compétences, me semble alors un devoir.

1. D'un point de vue plus légaliste, cela figure parmi les missions des enseignants-chercheurs dans l'article 3 du décret n° 84-431 sur les enseignants-chercheurs : « *Ils contribuent au dialogue entre sciences et sociétés, notamment par la diffusion de la culture et de l'information scientifique et technique* ».

2. La terminologie « scientifique » et « non-scientifique » est celle de l'Ademe, sans qu'elle soit bien définie. Il semble que des diplômes de psychologie ou sociologie soient classés dans « non-scientifique », ce qui semble assez contestable pour des sciences humaines…

Dans ce domaine, mes contributions se sont divisées en deux thématiques : l'enseignement autour du développement de l'esprit critique et des activités de médiation scientifique.

4.1 Enseignement de l'esprit critique

Attiré par la démarche critique, dans sa notion d'autodéfense intellectuelle, j'avais à cœur, dès mon recrutement comme maître de conférences, de proposer un enseignement autour de cette thématique car je considérais qu'il s'agissait d'un pré-requis important dans notre rôle de formation des étudiant·e·s qui vise aussi à les aider à se construire en tant que citoyennes et citoyens.

4.1.1 Dans une formation de journalistes

En discutant avec différents collègues de ce projet, j'ai pu rencontrer un responsable du M2 « journalisme scientifique » qui était conjoint entre l'université Lille 1 et l'École Supérieure de Journalisme (ESJ) de Lille. Il était intéressé par ce projet et m'a donc confié quelques heures d'enseignement pendant l'année scolaire 2013-2014.

Ce premier cours abordait les bases de l'esprit critique, en mettant en lumière les biais qui peuvent affecter nos opinions (y compris celles des chercheurs et chercheuses), en proposant des exemples de paradoxes statistiques, en examinant la conception de protocoles expérimentaux, en discutant des études scientifiques et en présentant divers artifices rhétoriques.

Ces différentes thématiques avaient pour but de stimuler le doute face à des informations scientifiques auxquelles ces futurs journalistes feront face. Néanmoins, stimuler le doute seul serait vain. Faire preuve d'esprit critique face à certaines affirmations, ce n'est pas tout remettre en cause, tout critiquer. En effet, une définition de l'esprit critique est : « *la capacité à aboutir à des conclusions raisonnables fondées sur des preuves, la logique et l'honnêteté intellectuelle* » (Rowe *et al.*, 2015). Or, la remise en cause systématique d'informations au prétexte de doutes ou d'incertitudes ne permet pas d'arriver à des conclusions raisonnables mais mène simplement au déni ou à l'inaction comme, par exemple, pour le dérèglement climatique[3]. L'épidémiologiste Bradford Hill le formulait clairement dans les années 1960 : « *Tout travail scientifique est incomplet, qu'il soit observationnel ou expérimental. Tout travail scientifique peut être remis en question ou modifié par le savoir qui progresse. Cela ne nous donne pas la liberté d'ignorer les connaissances que nous avons déjà, ou de reporter l'action qu'elles semblent exiger à un moment donné.* » (Hill, 1965).

L'industrie du tabac ne s'y est pas trompée : c'est dans ses mémos internes qu'on peut lire la phrase, devenue fameuse, d'un haut dirigeant de l'industrie : « *le doute est notre produit* »[4] (figure 4.1).

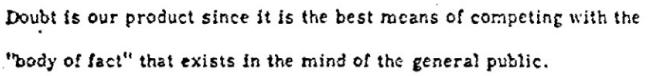

FIGURE 4.1 – « *Le doute est notre produit* », extrait d'un mémo interne de l'industrie du tabac, en 1969.
ID : `hhvf0191`, Truth Tobacco Industry Documents, UCSF

Afin d'alerter les futurs journalistes sur cette instrumentalisation du doute, j'avais également intégré à ce premier cours une partie sur les stratégies du doute ainsi que sur le biais de financement. Le biais de financement est un biais observé dans les résultats ou conclusions des études scientifiques : celles financées ou en lien avec l'industrie ont plus de chances d'avoir un résultat favorable à l'industrie en question. Sa non prise en compte peut laisser penser que sur certains sujets la littérature scientifique est encore controversée alors qu'il s'agit seulement des recherches financées par l'industrie qui contribuent à semer le doute. Deborah Barnes et Lisa Bero en ont offert l'illustration la plus saillante en analysant une centaine de *reviews* à propos du tabagisme passif. La seule différence statistiquement significative entre celles qui identifiaient un risque et celles qui n'en identifiaient pas était de savoir si des auteurs étaient liés à l'industrie du tabac (Barnes et Bero, 1998). Cette présentation du biais est également accompagnée de ces explications : le plus souvent, il ne s'agit pas de scientifiques verreux prêts à publier ce que l'industriel leur commande afin de bénéficier de généreux financements. Au contraire, il s'agit d'un méta-biais

3. À ce titre, il se trouve certains défenseurs de la pensée critique ont sombré dans un doute systématique, doutant du réchauffement climatique, voire de l'Holocauste.

4. Document disponible sur le site de l'UCSF dédié aux mémos internes de l'industrie (ID : hhvf0191).

qui rassemble différentes influences subtiles pouvant interférer sur le protocole expérimental, l'analyse des résultats, entrainer des biais de publication, etc.

Suite aux retours positifs des étudiant·e·s, on m'a confié un volume horaire allant croissant au fil des années, jusqu'à atteindre 24 h en 2021-2022. À la rentrée 2022, un changement s'opère dans la formation des journalistes scientifiques : elle ne se fait plus en une seule année, mais en deux ans. La première année est alors conjointe avec les journalistes généralistes. La formation devient partagée entre l'ESJ de Lille et Sciences Po Lille. On me propose 6 h de cours à Sciences Po lors de cette première année conjointe, avec le défi de parler des notions de preuve et de doute en sciences à un public qui n'est pas forcément friand de sciences.

La notion de doute me tient à cœur, pour les raisons explicitées précédemment. L'équilibre à trouver entre un excès de doute (hypercriticisme) et une absence de doute (naïveté) est impossible à tenir. Nous pouvons facilement pencher, selon les sujets, tantôt d'un côté ou tantôt de l'autre. Je choisis, comme fil rouge pour ce cours, un sujet sur lequel les étudiant·e·s n'ont probablement pas déjà un avis forgé, afin d'éviter des effets de réactance qui iraient contre leurs croyances. Nous discutons de la (fumeuse) étude d'Andrew Wakefield, désormais rétractée, qui entendait montrer un lien entre la vaccination contre la rougeole, les oreillons et la rubéole (ROR) et un trouble du développement [5]. L'étude était trop faible pour raisonnablement semer le doute sur la sécurité de la vaccination ROR (DeStefano et Shimabukuro, 2019). En plus de cela, elle se révélera ensuite frauduleuse. Pour autant, celle-ci défraiera la chronique outre-Manche, où réside Wakefield, ce qui entraînera une défiance de la population envers la vaccination ROR, puis une baisse de la couverture vaccinale de près de 10 points (Lewis et Speers, 2003). Dans cette polémique, le fait de ne pas avoir suffisamment mis en doute cette étude l'a semé sur la vaccination. D'une certaine manière, le doute peut protéger du doute.

Cet épisode Wakefield et ses conséquences sur la couverture vaccinale ont aussi été instrumentalisés par des personnes qui entendent alerter sur les rhétoriques d'anti-vaccins. En effet, diverses personnes, notamment le sociologue Gérald Bronner, vont affirmer que l'étude de Wakefield est aussi à l'origine d'un mouvement de défiance vaccinale en France qui aurait causé les épidémies de rougeole que nous avons connues dans les années 2000 et 2010. Le raisonnement semble cohérent : il y a effectivement eu des épidémies de rougeole importantes ces dernières années et l'étude conduite par Wakefield a eu un effet délétère sur la confiance dans la vaccination ROR au Royaume-Uni, on pourrait alors penser qu'il en a été de même chez nous. Pourtant, cette causalité n'est pas établie. La prémisse elle-même ne l'est pas : la couverture vaccinale de la vaccination ROR n'a jamais baissé en France et finit par être assez haute, quand les enfants grandissent [6], ce qui signifie que les enfants tardent à être vaccinés, pas qu'ils ne le sont pas. D'autre part, les foyers de l'épidémie de rougeole de 2008–2012 ont été identifiés par une géographe et il s'agit de communautés historiquement anti-vaccins, qui n'ont pas attendu Wakefield pour l'être (Guimier, 2017).

Cet exemple est assez emblématique d'une situation où les arguments opposés les plus vocaux ne sont pas nécessairement les plus pertinents. Le doute est de mise des deux côtés, d'une part avec les résultats de Wakefield, d'autre part avec l'instrumentalisation qui est faite des conséquences de ses fausses informations, sans non plus renvoyer dos à dos les deux parties : dans un cas, il s'agit de fraude, dans un autre de déformation.

4.1.2 En licence informatique

En plus de cet enseignement auprès de futurs journalistes, j'ai proposé un enseignement similaire au département d'informatique. Celui-ci a ouvert à la rentrée 2014, sous forme d'option, rassemblant en général de 25 à 30 étudiant·e·s. Ce cours se situait plus résolument dans le cadre de « l'autodéfense intellectuelle » que le cours auprès des futurs journalistes. Ce terme d'autodéfense intellectuelle vient d'un professeur de sciences de l'éducation, auteur d'un *Petit cours d'autodéfense intellectuelle*, Normand Baillargeon (Baillargeon et Charb, 2006).

Le principe de l'esprit critique est d'aboutir à des conclusions raisonnables. Or, afin d'arriver à ces conclusions raisonnables, encore faut-il déconstruire notre manière de penser et la façon dont elle a été modelée par notre *cultivation* aux médias, et en particulier à la télévision (Morgan et Shanahan, 2010).

5. Je reste volontairement vague afin de ne pas contribuer à véhiculer la fausse information.

6. Données disponibles sur le site de Santé Publique France : https://www.santepubliquefrance.fr/determinants-de-sante/vaccination/articles/donnees-de-couverture-vaccinale-rougeole-rubeole-oreillons-par-groupe-d-age

Le cours entendait déconstruire nos propres biais sociaux ou cognitifs, puis aborder les différentes sources d'influence externe et la manière de les analyser (utilisation de statistiques, étude de sondages, influence des médias). Je présentais également des aspects plus spécifiques de la démarche scientifique, comme l'établissement de protocoles expérimentaux.

Au profit d'une nouvelle maquette, l'option a été promue au rang de cours obligatoire. Néanmoins, le volume horaire étant plus faible, il a fallu faire des choix. C'est en particulier les aspects de démarche scientifique qui ont été conservés. En m'appuyant sur la littérature sur le sujet, j'ai développé une partie plus axée sur l'argumentation, qui est également un élément clé de l'esprit critique (Groarke et Tindale, 2012 ; Dwyer, 2017).

4.2 Activités de médiation scientifique

Un autre axe de diffusion de la culture scientifique consiste à entreprendre des activités de médiation scientifique. J'ai mené des actions à la fois sur des thématiques en lien direct avec mes recherches mais également en lien avec les cours que je viens d'évoquer.

4.2.1 Autour de la bioinformatique

À mon arrivée dans l'équipe Bonsai, il existait déjà un matériel de médiation conséquent et créé par l'équipe que nous avons utilisé à de nombreuses reprises pour expliquer les rudiments de l'alignement, de l'assemblage ou du repliement des ARN. Nous avons notamment fait des actions de médiation avec ce matériel au Palais de la Découverte, à Paris, et à de nombreuses reprises auprès de lycéens.

Avec Mathieu Giraud, nous avons également conçu une activité sur le même modèle afin d'expliquer notre algorithme de recherche de recombinaisons V(D)J. Nous avons choisi de commencer par illustrer un algorithme naïf, puis de montrer notre heuristique afin de comprendre le gain de temps qu'elle peut représenter. L'activité est composée d'une série de courtes séquences correspondant à des gènes V ou J. On montre ensuite un court read et on demande d'identifier de quels gènes V et J il est composé. Cette étape demande du temps puisqu'elle nécessite de comparer « à l'œil » tous les gènes au read. Ensuite, nous passons à l'illustration de notre heuristique. Nous utilisons une fenêtre que l'on va faire glisser le long du read. Dans ce cas, la seule question qu'on se pose est de savoir si la séquence présente dans la fenêtre est présente parmi les gènes V ou parmi les gènes J. À la fin du read, on sait donc où se trouve la région V dans le read et où se trouve la région J. Cela permet d'illustrer notre heuristique (détaillée en section 2.2.3.1, page 11). Ce petit jeu permet aussi de toucher du doigt la notion de complexité puisqu'on perçoit bien que cette seconde méthode a demandé bien moins de comparaisons que la première. Nous avons eu l'occasion de pratiquer ces jeux avec des élèves de collège ou lycée, mais aussi avec nos collègues à l'Inria ou dans notre unité.

D'autre part, pour un public plus avancé, nous avons également rédigé des articles de médiation scientifique sur la bioinformatique. Avec Mathieu Giraud, nous avons rédigé un article à propos du séquençage à haut débit et de l'indexation de séquences. Avec Hélène Touzet et Claire Lemaitre (chercheuse dans l'équipe Genscale, à l'Irisa, Rennes), nous avons rédigé un chapitre dans un rapport sur le rôle qu'a joué l'algorithmique des séquences dans l'identification, la compréhension et le suivi de la pandémie de Covid-19 (Alizon *et al.*, 2021). Ce chapitre a ensuite donné lieu à la publication d'articles sur le site d'Interstices.

4.2.2 Autour de l'esprit critique

Mon engagement dans l'enseignement de l'esprit critique m'a conduit à recevoir des sollicitations pour réaliser des interventions en lien avec ces thématiques. C'est une situation toujours délicate pour moi, car je ne suis pas didacticien des sciences et ne suis donc pas un expert du sujet (ce que je tâche de rappeler). J'ai néanmoins une connaissance du sujet par la lecture d'ouvrages et articles scientifiques du domaine et je peux également faire part de mon retour d'expériences dans mes enseignements.

Je suis donc intervenu pour l'événement de médiation scientifique *Pint of Science* en 2017, qui vise à faire intervenir des scientifiques dans des bars pour présenter des sujets scientifiques de manière didactique. J'ai également donné une conférence pour *Skeptics in the Pub* en 2019 à propos de l'enseignement de l'esprit critique. Une journaliste, ancienne étudiante, m'a

sollicité afin d'intervenir dans une vidéo pour les Éditions Belin, destinée aux enseignants en lycée, pour parler du doute. Enfin, l'association *Sciences Citoyennes* m'a invité à participer à un webinaire autour des *enjeux politiques de l'information scientifique*.

D'autre part, j'ai également produit des documents écrits autour de ces problématiques. Tout d'abord, j'ai créé et largement alimenté la page Wikipédia francophone autour du biais de financement [7]. Ce biais est parfois dénigré voire minimisé, car pour certains ayant une vision positiviste voire idéaliste, il faudrait ne considérer que les arguments et non d'où ils viennent. Si cela semble louable en théorie, confrontés à la réalité nous réalisons que ce positionnement n'est pas tenable puisqu'il aurait pour effet de ne pas prendre en compte ce biais de financement. Il n'est évidemment pas raisonnable d'ignorer un biais susceptible d'affecter des résultats scientifiques. Il me semblait donc important qu'un document résume les connaissances sur ce biais afin de percevoir l'importance qu'il peut avoir. D'autre part, dans un contexte où en tant que chercheurs et chercheuses nous sommes de plus en plus incité·e·s à solliciter des financements de l'industrie, il est important de ne pas perdre de vue ce que cela peut impliquer en terme d'influence inconsciente. J'ai également produit des critiques [8] des contenus de l'AFIS (association pour l'information scientifique). Bien qu'ayant des motivations laissant penser à un désir de médiation scientifique comparable au mien, cette association a tendance à rejeter certaines connaissances. En lien avec les stratégies du doute, dont j'ai déjà parlé, j'ai montré que l'association évoque très peu ces aspects, notamment les influences de certains lobbys sur les connaissances scientifiques alors que, dans le même temps, elle accuse un lobby en particulier, le « lobby vert », de divers maux peu étayés. Cette critique à deux vitesses est questionnable, quand on cherche à faire de l'information scientifique. Là aussi, il faut prendre garde à l'interprétation de ce fait, une absence de critique d'un lobby puissant (comme le lobby pétrolier ou le lobby chimique) ne signe évidemment pas une collusion avec celui-ci. Pour les raisons explicitées précédemment sur le déni du changement climatique, certains positionnements politiques, certaines valeurs peuvent suffire à expliquer ce positionnement.

7. https://fr.wikipedia.org/w/index.php?title=Biais_de_financement

8. https://www.fil.univ-lille.fr/~salson/zetetique/afis.html et https://www.fil.univ-lille.fr/~salson/zetetique/afis-lobbying.html

Conclusions et perspectives

J'ai résumé dans les pages qui précèdent quelques-uns des travaux auxquels j'ai contribué ou que j'ai co-encadrés depuis mon arrivé en tant que maître de conférences dans l'équipe Bonsai, en 2010. J'ai également brièvement rappelé l'état de l'art des domaines concernés.

Nous avons à notre disposition des méthodes sans alignement qui parviennent à obtenir des résultats comparables (voire meilleurs) aux approches avec alignement en n'utilisant qu'une fraction des ressources que ces dernières sollicitent.

En terme d'indexation, nous avons à l'heure actuelle des solutions pour indexer des dizaines de milliers de jeux de données de séquençage et y faire des requêtes de comptage. À n'en pas douter, les travaux actuels, dans notre équipe et dans le monde, vont encore repousser ces limites. Indexer des pétanucléotides, pour obtenir la quantification des séquences requêtées, est à portée de main. À la fin de ma thèse (en 2009), je n'aurais probablement pas osé pronostiquer qu'on parviendrait à construire des index sur un téranucléotides.

Des données qui ne sont pas données

Comme je l'ai rappelé en introduction, le contexte dans lequel nous évoluons est celui d'une croissance exponentielle ininterrompue de la quantité de séquences nucléiques accessibles, depuis l'avènement du séquençage à haut débit. Ce défi, dont la difficulté croît au fil du temps, est extrêmement stimulant intellectuellement. Des solutions originales et pertinentes apportées à un moment donné se retrouvent ensuite dépassées par le volume de données qu'elles ne sont plus adaptées à traiter. Si on réussit à indexer un pétanucléotides, saura-t-on le faire pour un exanucléotides ?

Mais ces données, qui nous abreuvent, portent bien mal leur nom. Le *Trésor de la langue française* nous propose comme définition par extension du mot *donnée* : « *Ce qui est connu et admis, et qui sert de base, à un raisonnement* ». Nous considérons comme acquis, presque comme une loi de la nature, le fait que ces données soient produites et que leur quantité croisse exponentiellement. Or, ces données sont le plus souvent produites par nos collègues. Prenons le cas de la santé, auquel j'ai consacré le chapitre 2 page 5. Afin d'améliorer les traitements des patients et *in fine* leur santé, il paraît parfaitement légitime d'avoir l'information la plus précise possible. Une façon d'y parvenir est d'accéder à l'information génétique, avec des ambitions allant croissant. Cela concernait d'abord le séquençage de quelques locus d'intérêt, puis de quelques gènes, puis de l'exome, ensuite du transcriptome et enfin du génome complet, voire du métagénome intestinal (puis des métagénomes ?). C'est d'ailleurs la nature du projet France médecine génomique 2025 dont le but était de parvenir à séquencer annuellement 235 000 génomes à l'horizon 2020 [9]. Au niveau international, se développe actuellement le diagnostic précoce de maladies chez les nouveaux-nés en utilisant du séquençage à haut débit (Kingsmore *et al.*, 2022). D'un point de vue bioinformatique, un tel projet est certainement très intéressant : il est nécessaire d'obtenir des résultats précis dans un temps court afin que les nouveaux-nés puissent bénéficier au plus vite des soins nécessaires. Mais le recours à ces données, leur utilisation, leur conservation soulève de nombreuses questions.

Jusqu'où pousser ce besoin de données ? Toute donnée est-elle nécessairement pertinente à acquérir ? C'est dans ce sens qu'il ne faut pas forcément prendre pour acquise une donnée.

D'ailleurs, cette inflation des données génétiques commence à susciter des remises en cause (Owens, 2022). D'une part, l'accès à davantage de données, même si elles sont plus

9. https://www.inserm.fr/wp-content/uploads/2017-11/aviesan-planfrancemedecinegenomique-2025.pdf, un esprit taquin remarquera que l'objectif de séquencer 235 000 génomes a été rempli… pour les génomes du SARS-CoV-2.

précises, fait peser le risque d'un surdiagnostic dans certaines situations. Les dépistages systématiques de cancers du sein et de la prostate (ne reposant pas sur des données de séquençage) en ont été un exemple : la balance bénéfice-risque du dépistage systématique est discutée (Gøtzsche et Jørgensen, 2013 ; Ilic *et al.*, 2013). Owens (2022) mentionne le cas de généticiens échaudés après avoir annoncé à des patients qu'ils possédaient des variants pathogéniques, qui ont ensuite été reconsidérés comme bénins. De telles annonces ne sont pas sans conséquence sur la santé mentale, voire physique (en raison du recours à des examens supplémentaires) des personnes concernées.

La croissance du recours aux tests génétiques pose d'autres problèmes éthiques. À l'heure actuelle, les bases de données génétiques ont un important biais ethnique : toutes les populations ne sont pas aussi bien représentées (Sirugo *et al.*, 2019). Notre niveau de connaissance sur l'impact des variants dépend donc fortement des populations dans lesquelles ils sont les plus fréquents.

D'autre part, les données génétiques accessibles peuvent conduire à une instrumentalisation politique dangereuse. Dans ce sens, la société étatsunienne de génétique humaine a publié un communiqué, pendant la présidence de D. Trump, afin de rappeler qu'il était impossible de séparer les humains en sous-catégories biologiques, qu'il n'y a pas de bons ou de mauvais gènes et que la génétique n'a pas une part prépondérante dans les qualités ou défauts humains (American Society of Human Genetics, 2020). Or, l'existence de ces données de séquençage et les corrélations que certains identifient entre des ensembles de gènes humains et des caractéristiques comportementales pourraient être exploitées par des gouvernements autoritaires afin de discriminer certaines personnes en fonction de leur génétique. Les séquençages de génomes dès la naissance accroît ce risque.

De plus, la génération de grandes quantités de données nécessite des infrastructures adéquates afin de lancer les logiciels d'analyse ainsi que pour stocker de manière pérenne les données brutes. Or, la construction d'appareils électroniques a des impacts environnementaux qui ne sont pas anodins (Bordage, 2019). Ils sont énergivores et utilisent des ressources qui ne sont pas renouvelables et qui sont peu recyclées. Les solliciter pour l'analyse et le stockage de données dont on ne sait pas, *a priori*, si elles seront utiles est questionnable.

Ainsi, certains médecins préfèrent séquencer le nombre minimal de gènes pertinents plutôt que de partir à la pêche avec le plus gros filet possible. Dans notre position de (bio)informaticien·ne·s, nous pourrions confortablement estimer que la production de données n'est pas entre nos mains. Nous pouvons néanmoins discuter avec nos collègues de ce qui serait utile ou non et de l'intérêt de séquencer plus largement ou plus profondément. D'autre part, par la production d'algorithmes plus frugaux, nous nourrissons également cette avalanche de données. Les données sont produites car elles peuvent être analysées. Tentons une expérience de pensée dans laquelle les développements méthodologiques seraient restés bloqués dans les années 1970. Nous avions certes des méthodes pour indexer et comparer des séquences entre elles, mais de manière beaucoup moins frugale qu'actuellement. Dans ces conditions, le développement du séquençage à haut débit, notamment pour le diagnostic, aurait été beaucoup plus complexe car nécessitant des infrastructures extrêmement lourdes. C'est une illustration de l'*effet rebond* (Jevons, 1866) : une amélioration de l'efficacité des algorithmes a conduit à produire plus de données et, *in fine*, n'a probablement pas contribué à décroître la consommation globale d'énergie ou de ressources. Si on souhaite réduire cet impact, cela devrait donc passer par une sélection plus affutée des données à séquencer. Une telle approche d'ignorance volontaire est contre-intuitive dans notre métier où nous cherchons à contribuer aux connaissances. Mais il faut se poser la question plus générale de l'intérêt pour la société : des avantages que fournissent ces données et de leurs inconvénients autant à court terme qu'à long terme. Il s'agit-là d'une remise en question des données auxquelles on peut avoir accès. Elles ne devraient plus être considérées comme « *admises* » mais interrogées : de quoi a-t-on réellement besoin ?

Des méthodes diversement adoptées

À la lumière des différents projets auxquels j'ai contribué, je peux tenter de dessiner les raisons pouvant expliquer l'utilisation plus ou moins importante des logiciels qui en ont découlé. C'est néanmoins un exercice à réaliser avec prudence, puisque je n'ai évidemment pas employé un protocole expérimental rigoureux afin de déterminer les variables expliquant l'impact de mes recherches. Pour m'aider dans cet exercice, je vais m'appuyer sur les travaux

		CRAC (§2.1 p.6)	Vidjil (§2.2 p.8)	FiLT3r (§2.3 p.21)	Bwolo (§3.2 p.31)	TL-index (§3.3 p.36)	REINDEER (§3.4 p.38)
accessibilité	ouverture	✓	✓	✓	✓	✓	✓
	compréhension mutuelle	✓	✓	✓	~	~	✓
utilité	intégration	✓	✓	✓	?	✓	~
	interface	✗	✓	✗	✗	✓	✓
portabilité	standards	✓	✓	✓	✗	✗	✗
	maintenance	✓	✓	✓	✗	~	✓
	application hors-contexte	?	✓	✓	✓	?	✓

TABLE 4.1 – Évaluation des logiciels auxquels j'ai contribué (comme co-auteur ou co-encadrant) à la lumière des critères de Douglas *et al.* (2011).

de Douglas *et al.* (2011), mentionnés dans l'introduction de ce document. La table 4.1 présente certains critères de Douglas *et al.* (2011), facilement évaluables par rapport aux recherches que j'ai présentées.

Sans rentrer dans le détail de chaque critère pour chaque logiciel, ce qui risquerait d'être fastidieux, notons que toutes les méthodes sont ouvertes, elles sont sous licence libre et accessibles sur des dépôts git. Il est intéressant de noter que le logiciel qui a le plus de succès (Vidjil) est celui qui coche toutes les cases. À l'inverse, les prototypes développés pendant les thèses sont ceux en cochant le moins, conséquence du manque de temps ou de moyen pour faciliter l'utilisation de ces productions. Néanmoins, il ne faut pas perdre de vue que c'est parfois l'utilisation d'un logiciel qui nous conduit à adopter des pratiques plus vertueuses (comme l'utilisation de standards, le développement d'interface) sans quoi nous n'avons pas les ressources, ou l'énergie, pour aller jusque-là.

Dans le cas du dernier logiciel produit, FiLT3r, nous avons essayé d'appliquer les méthodes qui fonctionnent afin de favoriser son adoption. L'absence d'interface d'utilisation est due à un échec technique, décrit en section 2.3.4 page 25, qui n'a pu être résolu faute de financement dédié.

Pour le TL-index, bien qu'il s'agisse d'un prototype de thèse, le fait qu'il s'inscrive dans le projet Vidjil lui a permis de bénéficier de la communauté d'utilisateurs et d'un développement spécifique, et ainsi d'améliorer son utilisabilité. Pour autant, l'index utilisé comme prototype est très rarement utilisé. À mon sens, cela pose la question de l'utilisation hors du contexte habituel. Pour les personnes qui utilisent Vidjil, cela nécessite de se poser des questions qu'elles ne se posaient pas habituellement, cela ne rentre pas dans leur pratique habituelle, à l'inverse des tâches réalisées avec Vidjil qui ressemblent pour partie à ce qui se faisait en biologie moléculaire plus classique. Je pense que c'est dans ce type de situation qu'une collaboration forte et suivie avec des laboratoires est nécessaire afin d'inscrire l'utilisation de l'index dans une démarche qui a du sens dans leur travail et qui apporte des réponses que nos collègues peuvent s'approprier. Dans ce sens, REINDEER répond parfaitement aux besoins que nous avions imaginés avec nos partenaires d'ANR lors de la rédaction de celle-ci. REINDEER est donc bien utilisé au sein du consortium. Pour autant, REINDEER ne s'inscrit pas réellement dans une démarche existante pour des laboratoires extérieurs à notre consortium. Il est nécessaire de sortir de sa routine habituelle, de sa manière classique d'analyser les données afin d'apprécier pleinement le rôle que peut jouer REINDEER.

À partir de mon expérience, j'identifie comme condition nécessaire, mais non suffisante, à ce qu'un logiciel de bioinformatique soit utilisé relativement largement, qu'il soit issu d'une collaboration exigeante avec de futurs utilisateurs potentiels. Sans cela, le risque est grand que le logiciel ne réponde qu'à ce qu'on s'imaginait être pertinent pour des biologistes, des médecins, des bioanalystes, mais qu'en réalité le logiciel ne s'inscrive pas dans la pratique de ces

personnes ou ne réponde que de manière partielle et insatisfaisante aux questions qu'elles se posent. Comme je l'indique, il ne s'agit évidemment pas d'une condition suffisante. Notamment si le logiciel change les pratiques habituelles. Cela peut nécessiter du temps pour que des laboratoires changent leurs habitudes d'analyse. À titre d'exemple, le logiciel TopHat2 (Kim *et al.*, 2013), qui analysait des données de séquençage d'ARN (RNA-seq), continue à être utilisé longtemps après l'arrêt de son développement et après que ses auteurs ont recommandé d'utiliser un autre logiciel (qu'ils avaient également développé). Sur la seule année 2022, six ans après la recommandation des auteurs, Tophat2 était encore cité plus de 1 000 fois. Pourtant, passer d'un logiciel à son successeur n'est qu'un changement mineur. Cela illuste combien de nouvelles approches changeant les pratiques plus en profondeur peuvent mettre du temps à percoler dans les différents laboratoires susceptibles d'être intéressés.

Méthodes génériques sans alignement

Nous l'avons vu, les méthodes sans alignement constituent une part importante de mes contributions. Pour CRAC, Vidjil, FiLT3r, les principes généraux sont comparables : il s'agit de découper un read en *k*-mers et de rechercher chacun de ces *k*-mers dans une séquence de référence afin d'en inférer différentes informations (épissage, recombinaisons V(D)J, duplications, etc.). La question de la généricisation se pose évidemment : pourrait-on produire un logiciel qui permette, de manière générique, de produire les résultats que produisent ces trois logiciels ? La question se pose évidemment au-delà de ces trois logiciels, sans toutefois vouloir regrouper toutes les approches sans alignement. En effet, il existe des différences importantes entre une approche à la km (Audemard *et al.*, 2019) qui construit un graphe de de Bruijn d'une séquence de référence afin d'identifier des variations par rapport à celle-ci et une approche à la Salmon pour la quantification d'ARN (Patro *et al.*, 2017) qui fait du quasi-*mapping*.

Une approche générique consisterait à rechercher chaque *k*-mer (et, même, de manière plus générique, chaque graine) de chaque read. Cette recherche pourrait n'être qu'une information de présence/absence (ce dont nous avons besoin pour Vidjil), de comptage (comme dans l'heuristique pour la détection de sous-clonotypes, page 20) ou de localisation (ce que nous utilisons pour CRAC et FiLT3r). À partir d'une liste des occurrences des *k*-mers pour chaque read, il s'agit de déterminer si un événement d'intérêt est présent ou non dans le read en question. Cette approche dépend naturellement de la question posée mais peut être en partie généricisée. Dans le cas de CRAC et de FiLT3r, l'intérêt est dans le différentiel de localisation avant et après des séries de *k*-mers non localisés. Pour Vidjil, c'est la présence de *k*-mers provenant de différentes séquences de référence qui importe. Ensuite, des ensembles de reads doivent être définis, partageant les mêmes propriétés : des clonotypes pour Vidjil, partageant une même fenêtre, des duplications en tandem pour FiLT3r, partageant une duplication de même longueur aux mêmes positions.

Une généricisation de ce type d'approches sans alignement, pour en faire une bibliothèque permettrait de simplifier le développement d'outils recourant à des méthodes sans alignement. De plus, l'optimisation des étapes critiques de cette bibliothèque permettrait de créer plus facilement des logiciels efficaces. Il resterait ensuite à développer plus spécifiquement pour chaque logiciel les étapes aval, qui semblent moins génériques.

Concrètement, une telle bibliothèque aurait simplifié le développement d'un logiciel conçu au début de la pandémie de Covid-19, avec l'entreprise SeqOne, pour identifier les amorces de PCR détectées dans des échantillons de patients, afin de déterminer ceux positifs et négatifs au SARS-CoV-2. D'autre part, une telle bibliothèque aurait également un intérêt pédagogique : en simplifiant le développement de programmes bioinformatiques, cela pourrait rendre ces approches plus accessibles à des TP, ce qui permettrait à des étudiant·e·s de plus facilement s'approprier ces concepts.

Limites en temps des méthodes sans alignement

L'efficacité en temps des méthodes sans alignement est telle qu'elle pose la question de l'effet rebond, mentionné ci-dessus. Si les programmes sont plus efficaces, nous pouvons réaliser plus d'analyses ou des analyses sur davantage de données. Naturellement, nous n'allons pas parsemer nos codes de `sleep(1)` afin d'éviter un effet rebond ! À mon sens, s'il faut éviter un effet rebond, l'effort est plutôt à cibler sur la quantité de données séquencées (comme évoqué

précédemment). Des gains en efficacité sur les algorithmes d'analyse sont de toute façon intéressants à prendre (de même que des gains en efficacité sur une ampoule, une machine à laver ou un véhicule sont intéressants).

L'efficacité des méthodes sans alignement soulève d'autres questions. Par exemple, FiLT3r, lancé sur un seul thread, n'est que trois fois plus lent que la seule décompression par gunzip de ces mêmes données. La décompression des données devient donc une étape prenant une part non négligeable de l'analyse. De plus, cela nuit à la parallélisation du traitement car la parallélisation de gunzip n'est pas triviale (Kerbiriou et Chikhi, 2019).

Bien évidemment, d'autres formats de compression, plus adaptés au stockage de reads et permettant une décompression en parallèle, seraient souhaitables. Malgré les nombreuses propositions pertinentes qui existent (Hernaez *et al.*, 2019), et la standardisation ISO d'un format en 2019 (MPEG-G), il faut se rendre à l'évidence : aucune solution n'a pris le pas et gzip reste la méthode de compression de référence pour les données que nous traitons.

Si gzip reste le standard de fait et que la décompression en parallèle reste un défi, alors nous n'avons plus qu'une seule possibilité pour lever cette limite du temps de décompression : ne pas décompresser les données. L'analyse de données compressées présente un avantage : la redondance exploitée par gzip [10] pourrait être mise à profit afin d'éviter de requêter à nouveau des *k*-mers qui l'ont déjà été.

Je resterais néanmoins prudent sur l'intérêt d'une telle approche : à moins de jeux de données avec une forte redondance, il y a peu de chance que la proportion de *k*-mers identiques dans un contexte de taille relativement modeste soit importante. Le gain risque alors d'être limité. De plus, l'approche serait dépendante de la méthode de compression ce qui la rendrait caduque si gzip finissait (enfin) par être remplacé par une méthode plus appropriée.

Finalement, il est possible que pour certaines problématiques (comme pour FiLT3r), nous nous approchions de ce qui peut être fait de plus efficace, à la fois en temps et en mémoire, à partir de ces données. Il ne serait pas forcément pertinent de chercher à grapiller quelques pourcentages d'amélioration sur des temps d'exécution et de consommation mémoire qui sont déjà très faibles.

Au-delà de l'automatisation

La bioinformatique a longtemps consisté à automatiser des tâches qu'il devenait fastidieux de réaliser manuellement, ou dont les résultats manquaient d'exhaustivité. Mes travaux n'échappent pas à ce constat.

Des logiciels comme Vidjil et FiLT3r sont la conséquence du passage de techniques classiques de biologie moléculaire avec l'utilisation de Genescan ou de séquençages Sanger, qui ne se concentrent au final que sur une poignée de séquences, au séquençage à haut débit. Pour autant, le type de résultat obtenu est le même. La différence réside principalement dans la quantité de données produites, avec des résultats qui ne se restreignent pas à quelques molécules.

Or, puisque la technique change, les critères employés pour prendre des décisions sur la base des ces résultats devraient également évoluer. Ce n'est pourtant pas ce que je constate dans ces deux exemples. Les mêmes critères cliniques sont employés (taux de mutations, quantification, etc.) pour caractériser les résultats. Évidemment, des critères qui ont été établis à la suite de multiples études cliniques ne peuvent être renversés immédiatement à l'arrivée d'une nouvelle technique. Pour autant, l'accès à des données plus vastes qu'auparavant peut fournir des critères plus pertinents afin de juger du niveau de sévérité de la maladie étudiée (par exemple *via* la prise en compte de la diversité du répertoire immunitaire (Kotrova *et al.*, 2015)). En tant que bioinformaticien, j'ai évidemment peu de marge de manœuvre pour influer sur les critères cliniques pris en compte. Néanmoins, nous pouvons collectivement fournir des outils qui aident les médecins et biologistes à définir de meilleurs critères.

Aussi, concernant Vidjil, nous avons l'ambition de proposer aux différents hôpitaux français de rassembler les informations contenues dans les dizaines de milliers d'échantillons de répertoires immunitaires de leucémies qu'ils ont séquencés afin d'en tirer des informations pertinentes. Il s'agit nécessairement d'un travail de longue haleine, nécessitant en amont de

10. La méthode de compression consiste, succinctement, à stocker des informations en faisant référence à celles qui précèdent. Il s'agit plus concrètement de stocker des références vers une suite identique de caractères préalablement compressée, dans un contexte borné de quelques dizaines de kilo-octets.

définir des nomenclatures communes. Un tel travail ne demandera pas forcément un travail algorithmique stimulant mais il semble néanmoins indispensable pour une bonne exploitation des données existantes.

En effet, la réutilisation des données existantes pourrait être un impératif éthique. Comme je l'ai indiqué précédemment, la production et le stockage de ces données a des impacts. Aussi, une fois celles-ci produites, nous devrions exploiter tout leur potentiel et ne pas les considérer comme des produits jetables qui ne serviraient qu'une seule fois et dormiraient ensuite au fond d'un disque de sauvegarde.

Nos travaux autour de l'indexation de jeux de données de séquençage se placent, à mon sens, dans ce cadre. Ce sont de telles structures d'indexation qui permettent de répondre à de nouveaux types de questions qu'on ne se posait pas (ou qu'on n'osait pas se poser, faute de pouvoir obtenir des réponses). Le projet Serratus (Edgar *et al.*, 2022), qui a analysé des millions de jeux de données publics pour caractériser de dizaines de milliers de nouveaux virus à ARN, est un exemple du type d'informations disponibles dans ces données dormantes. Pour autant, ce projet ne s'appuie pas sur un index de ces jeux de données [11] mais sur une ré-analyse exhaustive. Pour rendre plus accessible ce type d'analyse, il serait souhaitable d'arriver à indexer l'ensemble des données accessibles dans SRA, la banque mentionnée dans l'introduction de ce manuscrit qui stocke les données de millions de séquençages à haut débit.

Au-delà des classiques requêtes à base de k-mers dans des index de téranucléotides, le besoin se porte également sur des requêtes plus sémantiques, comme identifier les jeux de données les plus similaires à un jeu de données (en terme de séquences voire d'expression) ou identifier les séquences qui rendent spécifiques un ensemble de jeux de données par rapport aux autres. Pour dépasser l'automatisation de processus pré-existants, il est nécessaire de réfléchir à des requêtes pertinentes sur ces index et de développer les algorithmes et structures qui y répondront de manière efficace.

Indexation : poser les limites

Nous l'avons vu (section 3.1.3 page 30), de nombreuses solutions existent pour l'indexation de milliers, voire millions, de jeux de données de séquençage, pour répondre à des requêtes d'existence, de comptage ou de localisation. Tout porte à croire que de futures recherches, notamment celles que nous menons dans le cadre du projet ANR FullRNA, permettront d'indexer encore plus de données. Jusqu'où ?

Si les requêtes de localisation sont très utiles lorsqu'il s'agit d'interroger un faible nombre de jeux de données, on peut se demander s'il est encore nécessaire de savoir qu'un k-mer apparaît dans tels reads parmi des dizaines de milliers de jeux de données. La plus-value informationnelle vaut-elle les ressources supplémentaires qu'il faut mobiliser, quand, dans le même temps, nous développons des approches sans alignement montrant que ce type d'information n'est pas indispensable ? L'intérêt des requêtes d'existence ou de comptage (parfois probabilistes) est de permettre d'éviter de stocker une quantité importante d'informations, avantage à ne pas négliger. En effet, quand le volume de données est tel, peut-être vaut-il mieux se contenter d'indexer une information dégradée, plutôt que de vouloir une indexation exhaustive qui, de toute façon, n'apportera qu'une plus-value faible avec un surcoût prohibitif.

Projetons-nous quelques années dans le futur et supposons que nous réussissions à indexer une partie significative des jeux de données de séquençage disponibles dans SRA ou ENA, c'est-à-dire des millions de jeux de données, afin de répondre à des requêtes de comptage. Un problème fait face : la plupart des utilisateurs ou utilisatrices ne seront pas intéressées par la totalité des données indexées. Une personne travaillant sur la génétique humaine aura probablement peu d'intérêt pour les séquençages de métagénomique marine, et inversement pour une spécialiste de biologie marine. Dans ce cas de figure, serait-il pertinent de rechercher dans des millions de jeux de données alors que la grande majorité n'aurait pas d'intérêt en regard de la question posée ?

Une question fondamentale sera donc celle des requêtes restreintes, et des structures d'indexation qui les rendront possibles. Une requête restreinte serait une requête que l'on restreint

11. Une version préliminaire de ce manuscrit affirmait : « *personne n'est parvenu à en indexer ne serait-ce que 10 %* ». Entre temps, Shiryev et Agarwala (2023) m'ont démenti, ce qui illustre la rapidité de l'avancée de ces solutions, bien que ce soit l'infrastructure plus que la méthodologie qui, ici, ait permis le passage à l'échelle.

dynamiquement à un sous-ensemble de jeux de données [12]. Le but de telles restrictions serait d'avoir des requêtes dont la complexité en temps soit indépendante du nombre de jeux de données indexés mais dépendante du nombre de jeux de données requêtés (et du nombre d'occurrences qu'on y trouve). D'une certaine manière, cela peut faire le lien avec l'indexation annotée, abordée en section 3.3 page 36, mais avec des annotations de nature différente. Ici, nous voudrions que seuls les jeux de données présentant certaines annotations d'intérêt soient requêtés. La solution que nous avons proposée pour les recombinaisons V(D)J ne serait pas adaptée à cette nouvelle question. Au-delà de la solution apportée, il n'est même pas évident qu'une telle structure d'indexation soit possible. Par essence, une structure d'indexation va rassembler les séquences similaires afin de les retrouver plus facilement. Cette propriété, qu'on retrouve dans des structures aussi diverses que les arbres de suffixes, les transformées de Burrows-Wheeler, les graphes de de Bruijn, les listes inversées, semble contradictoire avec la capacité à faire des requêtes restreintes. Je pense néanmoins que ce serait un axe de recherche intéressant à explorer, mais risqué.

Pour relever ces nouveaux défis stimulants (sur la généricisation des méthodes sans alignement, sur des requêtes évoluées pour de grandes structures d'indexation ou de l'indexation restreinte), il faudra continuer à maintenir un lien fort entre les réflexions théoriques sur l'indexation ou la comparaison et leurs applications à des problématiques actuelles, en conservant des collaborations étroites avec les laboratoires intéressés par ces développements. Tout ceci en gardant un regard critique sur nos recherches afin d'avoir conscience de leurs impacts positifs ou négatifs et de savoir les orienter dans la direction qui semble la plus bénéfique à la lumière des connaissances actuelles.

12. Si la restriction est statique, donc définie à l'avance, le problème devient simple : il suffit de construire un index par sous-ensemble.

Bibliographie

A. V. AHO et M. J. CORASICK : Efficient string matching : an aid to bibliographic search. *Communications of the ACM*, 18(6):333–340, 1975.

E. ALAMYAR, V. GIUDICELLI, S. LI, P. DUROUX et M.-P. LEFRANC : IMGT/HighV-QUEST : the IMGT® web portal for immunoglobulin (IG) or antibody and T cell receptor (TR) analysis from NGS high throughput and deep sequencing. *Immunome Research*, 8(1), 2012.

S. ALIZON, F. CAZALS, S. GUINDON, C. LEMAITRE, T. MARY-HUARD, A. NIARAKIS, M. SALSON, C. SCORNAVACCA et H. TOUZET : SARS-CoV-2 Through the Lens of Computational Biology : How bioinformatics is playing a key role in the study of the virus and its origins. Research report, CNRS, 2021.

S. F. ALTSCHUL, W. GISH, W. MILLER, E. W. MYERS et D. J. LIPMAN : Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

AMERICAN SOCIETY OF HUMAN GENETICS : American society of human genetics statement regarding concepts of "good genes" and human genetics, 2020. URL https://www.ashg.org/publications-news/ashg-news/statement-regarding-good-genes-human-genetics/.

E. O. AUDEMARD, P. GENDRON, A. FEGHALY, V.-P. LAVALLÉE, J. HÉBERT, G. SAUVAGEAU et S. LEMIEUX : Targeted variant detection using unaligned RNA-seq reads. *Life Science Alliance*, 2(4), 2019.

J. AUDOUX, N. PHILIPPE, R. CHIKHI, M. SALSON, M. GALLOPIN, M. GABRIEL, J. LE COZ, E. DROUINEAU, T. COMMES et D. GAUTHERET : DE-kupl : exhaustive capture of biological variation in RNA-seq data through k-mer decomposition. *Genome Biology*, 18(243), 2017.

N. BAILLARGEON et CHARB : *Petit cours d'autodéfense intellectuelle*. Lux, 2006.

D. E. BARNES et L. A. BERO : Why Review Articles on the Health Effects of Passive Smoking Reach Different Conclusions. *JAMA*, 279(19):1566–1570, 1998.

A. BEITINJANEH, S. JANG, H. ROUKOZ et N. S. MAJHAIL : Prognostic significance of FLT3 internal tandem duplication and tyrosine kinase domain mutations in acute promyelocytic leukemia : a systematic review. *Leukemia research*, 34(7):831–836, 2010.

D. BELAZZOUGUI, F. CUNIAL, T. GAGIE, N. PREZZA et M. RAFFINOT : Flexible indexing of repetitive collections. *In Unveiling Dynamics and Complexity : 13th Conference on Computability in Europe, CiE 2017, Turku, Finland, June 12-16, 2017, Proceedings 13*, p. 162–174. Springer, 2017.

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

D. A. BOLOTIN, S. POSLAVSKY, I. MITROPHANOV, M. SHUGAY, I. Z. MAMEDOV, E. V. PUTINTSEVA et D. M. CHUDAKOV : MiXCR : software for comprehensive adaptive immunity profiling. *Nature Methods*, 12(5):380–381, 2015.

O. BONHAM-CARTER, J. STEELE et D. BASTOLA : Alignment-free genetic sequence comparisons : a review of recent approaches by word analysis. *Briefings in Bioinformatics*, 15(6):890–905, 2014.

F. BORDAGE : Empreinte environnementale du numérique mondiale. Rap. tech., GreenIT.fr, 2019.

A. BOUDRY, S. DARMON, N. DUPLOYEZ, M. FIGEAC, S. GEFFROY, M. BUCCI, K. CELLI-LEBRAS, M. DUCHMANN, R. JOUDINAUD, L. FENWARTH, O. NIBOUREL, L. GOURSAUD, R. ITZYKSON, H. DOMBRET, M. HUNAULT, C. PREUDHOMME et M. SALSON : Frugal alignment-free identification of FLT3-internal tandem duplications with FiLT3r. *BMC bioinformatics*, 23(1):1–16, 2022.

D. BOY et R. CONSEIL : Les représentations sociales du changement climatique : 2000-2021. Rap. tech., Ademe, 2022.

K. BŘINDA, L. LIMA, S. PIGNOTTI, N. QUINONES-OLVERA, K. SALIKHOV, R. CHIKHI, G. KUCHEROV, Z. IQBAL et M. BAYM : Efficient and robust search of microbial genomes via phylogenetic compression. *bioRxiv*, p. 2023–04, 2023.

B. BRINEY, A. INDERBITZIN, C. JOYCE et D. R. BURTON : Commonality despite exceptional diversity in the baseline human antibody repertoire. *Nature*, 566(7744):393–397, 2019.

M. BRÜGGEMANN, M. KOTROVA, H. KNECHT, J. BARTRAM, M. BOUDJOGRHA, V. BYSTRY, G. FAZIO, E. FROŇKOVÁ, M. GIRAUD, A. GRIONI, J. HANCOCK, D. HERRMANN, C. JIMENEZ, A. KREJCI, J. MOPPETT, T. REIGL, M. SALSON, B. SCHEIJEN, M. SCHWARZ, S. SONGIA, M. SVATON, J. J. M. VAN DONGEN, P. VILLARESE, S. WAKEMAN, G. WRIGHT, G. CAZZANIGA, F. DAVI, R. GARCÍA-SANZ, D. DAVI, P. GROENEN, M. HUMMEL, E. MACINTYRE, K. C. STAMATOPOULOS, C. POTT, J. TRKA, N. DARZENTAS et A. LANGERAK : Standardized next-generation sequencing of immunoglobulin and T-cell receptor gene recombinations for MRD marker identification in acute lymphoblastic leukaemia ; a EuroClonality-NGS validation study. *Leukemia*, 2019.

M. BURROWS et D. J. WHEELER : A block-sorting lossless data compression algorithm. Rap. tech. 124, Digital Equipment Corporation, Palo Alto, California, 1994.

B. CAPPS, R. CHADWICK, Y. JOLY, T. LYSAGHT, C. MILLS, J. J. MULVIHILL et H. ZWART : Statement on bioinformatics and capturing the benefits of genome sequencing for society. *Human genomics*, 13:1–6, 2019.

H.-L. CHAN, T.-W. LAM, W.-K. SUNG, S.-L. TAM et S.-S. WONG : Compressed indexes for approximate string matching. *Algorithmica*, 58:263–281, 2010.

K. CHWALENIA, L. FACEMIRE et H. LI : Chimeric RNAs in cancer and normal physiology. *WIREs RNA*, 8(6):e1427, 2017.

D. COBAS, T. GAGIE et G. NAVARRO : A fast and small subsampled r-index. *arXiv :2103.15329 [cs]*, 2021.

T. H. CORMEN, C. E. LEISERSON, R. L. RIVEST et C. STEIN : *Introduction to algorithms*. MIT press, 2022.

X. DAI, Z. ZHUANG et P. X. ZHAO : Computational analysis of miRNA targets in plants : current status and challenges. *Briefings in bioinformatics*, 12(2):115–121, 2011.

A. DANEK, S. DEOROWICZ et S. GRABOWSKI : Indexes of large genome collections on a PC. *PLoS ONE*, 9(10):e109384, 2014.

A. L. de SEPTENVILLE, M. BOUDJOGHRA, C. BRAVETTI, M. ARMAND, M. SALSON, M. GIRAUD et F. DAVI : Immunoglobulin gene mutational status assessment by next generation sequencing in chronic lymphocytic leukemia. *In* A. LANGERAK, éd. : *Immunogenetics*, vol. 2453, p. 153–167. Methods in molecular biology (Clifton, NJ), 2022.

F. DESTEFANO et T. T. SHIMABUKURO : The MMR vaccine and autism. *Annual review of virology*, 6:585–600, 2019.

H. DÖHNER, E. ESTEY, D. GRIMWADE, S. AMADORI, F. R. APPELBAUM, T. BÜCHNER, H. DOMBRET, B. L. EBERT, P. FENAUX, R. A. LARSON *et al.* : Diagnosis and management of AML in adults : 2017 ELN recommendations from an international expert panel. *Blood*, 129(4):424–447, 2017.

A. DÖRING, D. WEESE, T. RAUSCH et K. REINERT : SeqAn an efficient, generic C++ library for sequence analysis. *BMC bioinformatics*, 9:1–9, 2008.

C. DOUGLAS, R. GOULDING, L. FARRIS et J. ATKINSON-GROSJEAN : Socio-cultural characteristics of usability of bioinformatics databases and tools. *Interdisciplinary Science Reviews*, 36 (1):55–71, 2011.

E. DREZEN, G. RIZK, R. CHIKHI, C. DELTEL, C. LEMAITRE, P. PETERLONGO et D. LAVENIER : GATB : genome assembly & analysis tool box. *Bioinformatics*, 30(20):2959–2961, 2014.

C. DRUMMOND et B. FISCHHOFF : Individuals with greater science literacy and education have more polarized beliefs on controversial science topics. *Proceedings of the National Academy of Sciences*, 114(36):9587–9592, 2017.

M. DUEZ, M. GIRAUD, R. HERBERT, T. ROCHER, M. SALSON et F. THONIER : Vidjil : A web platform for analysis of high-throughput repertoire sequencing. *PLOS One*, 11(11):e0166126, 2016.

C. P. DWYER : *Critical Thinking : Conceptual Perspectives and Practical Guidelines*. Cambridge University Press, 2017. ISBN 978-1-107-14284-8.

U. K. H. ECKER, S. LEWANDOWSKY, J. COOK, P. SCHMID, L. K. FAZIO, N. BRASHIER, P. KENDEOU, E. K. VRAGA et M. A. AMAZEEN : The psychological drivers of misinformation belief and its resistance to correction. *Nature Reviews Psychology*, 1(1):13–29, 2022.

R. C. EDGAR, J. TAYLOR, V. LIN, T. ALTMAN, P. BARBERA, D. MELESHKO, D. LOHR, G. NOVAKOVSKY, B. BUCHFINK, B. AL-SHAYEB *et al.* : Petabase-scale sequence alignment catalyses viral discovery. *Nature*, 602(7895):142–147, 2022.

P. FERRAGINA, R. GIANCARLO et G. MANZINI : The myriad virtues of wavelet trees. *Information and Computation*, 207(8):849–866, 2009.

P. FERRAGINA et G. MANZINI : Opportunistic data structures with applications. *In Proceedings 41st annual symposium on foundations of computer science*, p. 390–398. IEEE, 2000.

Y. FERRET, A. CAILLAULT, S. SEBDA, M. DUEZ, N. GRARDEL, N. DUPLOYEZ, C. VILLENET, M. FIGEAC, C. PREUDHOMME, M. SALSON et M. GIRAUD : Multi-loci diagnosis of acute lymphoblastic leukaemia with high-throughput sequencing and bioinformatics analysis. *British Journal of Haematology*, 2016.

B. A. GAËTA, H. R. MALMING, K. J. L. JACKSON, M. E. BAIN, P. WILSON et A. M. COLLINS : iHMMune-align : hidden Markov model-based alignment and identification of germline genes in rearranged immunoglobulin gene sequences. *Bioinformatics*, 23(13):1580–1587, 2007.

M. GIRAUD, M. SALSON, M. DUEZ, C. VILLENET, S. QUIEF, A. CAILLAULT, N. GRARDEL, C. ROUMIER, C. PREUDHOMME et M. FIGEAC : Fast multiclonal clusterization of V(D)J recombinations from high-throughput sequencing. *BMC Genomics*, 15(1):409, 2014.

V. GIUDICELLI, D. CHAUME et M.-P. LEFRANC : IMGT/GENE-DB : a comprehensive database for human and mouse immunoglobulin and T cell receptor genes. *Nucleic acids research*, 33 (suppl_1):D256–D261, 2005.

V. GIUDICELLI, D. CHAUME, J. BODMER, W. MÜLLER, C. BUSIN, S. MARSH, R. BONTROP, L. MARC, A. MALIK et M.-P. LEFRANC : IMGT, the international ImMunoGeneTics database. *Nucleic Acids Research*, 25(1):206–211, 1997.

S. GOG, T. BELLER, A. MOFFAT et M. PETRI : From theory to practice : Plug and play with succinct data structures. *In 13th International Symposium on Experimental Algorithms*, p. 326–337. Springer, 2014.

P. C. GØTZSCHE et K. J. JØRGENSEN : Screening for breast cancer with mammography. *Cochrane database of systematic reviews*, 6, 2013.

L. A. GROARKE et C. W. TINDALE : *Good Reasoning Matters : A Constructive Approach to Critical Thinking*. Oxford University Press Canada, 2012.

I. GUIGON, S. LEGRAND, J.-F. BERTHELOT, S. BINI, D. LANSELLE, M. BENMOUNAH et H. TOUZET : miRkwood : a tool for the reliable identification of microRNAs in plant genomes. *BMC genomics*, 20(1):1–9, 2019.

L. GUIMIER : La résistance aux vaccinations : d'un défi de santé publique à un enjeu de société. *In MIVILUDES : rapport d'activité 2016 et premier semestre 2017*, 2017.

R. S. HARRIS et P. MEDVEDEV : Improved representation of sequence bloom trees. *Bioinformatics*, 36(3):721–727, 2020.

M. HERNAEZ, D. PAVLICHIN, T. WEISSMAN et I. OCHOA : Genomic data compression. *Annual Review of Biomedical Data Science*, 2:19–37, 2019.

S. A. B. HILL : The Environment and Disease : Association or Causation ? *Proceedings of the Royal Society of Medicine*, 1965.

E. HSU : V(D)J Recombination : Of Mice and Sharks. *In* P. FERRIER, éd. : *V(D)J Recombination*, p. 166–179. Springer New York, 2009.

D. ILIC, M. M. NEUBERGER, M. DJULBEGOVIC et P. DAHM : Screening for prostate cancer. *Cochrane Database of Systematic Reviews*, 1, 2013.

K. R. JAMES et H. W. KING : Germs and germlines : how "public" B-cell clones evolve in the gut. *Immunology and Cell Biology*, 98(6):428–430, 2020.

W. S. JEVONS : The Coal Question ; An Inquiry Concerning the Progress of the Nation, and the Probable Exhaustion of our Coal-Mines. *Fortnightly*, 6(34):505–507, 1866.

M. KARASIKOV, H. MUSTAFA, G. RÄTSCH et A. KAHLES : Lossless Indexing with Counting de Bruijn Graphs. *bioRxiv*, 2022.

M. KERBIRIOU et R. CHIKHI : Parallel decompression of gzip-compressed files and random access to DNA sequences. *In International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, p. 209–217. IEEE, 2019.

K. KIANFAR, C. POCKRANDT, B. TORKAMANDI, H. LUO et K. REINERT : Optimum search schemes for approximate string matching using bidirectional fm-index. *arXiv :1711.02035*, 2017.

D. KIM, G. PERTEA, C. TRAPNELL, H. PIMENTEL, R. KELLEY et S. L. SALZBERG : TopHat2 : accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome biology*, 14(4):1–13, 2013.

S. F. KINGSMORE, L. D. SMITH, C. M. KUNARD, M. BAINBRIDGE, S. BATALOV, W. BENSON, E. BLINCOW, S. CAYLOR, C. CHAMBERS, G. DEL ANGEL *et al.* : A genome sequencing system for universal newborn screening, diagnosis, and precision medicine for severe genetic diseases. *The American Journal of Human Genetics*, 109(9):1605–1619, 2022.

M. KOTROVA, K. MUZIKOVA, E. MEJSTRIKOVA, M. NOVAKOVA, V. BAKARDJIEVA-MIHAYLOVA, K. FISER, J. STUCHLY, M. GIRAUD, M. SALSON, C. POTT, M. BRÜGGEMANN, M. FÜLLGRABE, J. STARY, J. TRKA et E. FRONKOVA : The predictive strength of next-generation sequencing MRD detection for relapse compared with current methods in childhood ALL. *Blood*, 126(8):1045, 2015.

L. KUCHENBECKER, M. NIENEN, J. HECHT, A. U. NEUMANN, N. BABEL, K. REINERT et P. N. ROBINSON : IMSEQ - a fast and error aware approach to immunogenetic sequence analysis. *Bioinformatics*, p. btv309, 2015.

G. KUCHEROV, K. SALIKHOV et D. TSUR : Approximate string matching using a bidirectional index. *In Combinatorial Pattern Matching*, Lecture Notes in Computer Science, p. 222–231. Springer, 2014.

A. KUHNLE, T. MUN, C. BOUCHER, T. GAGIE, B. LANGMEAD et G. MANZINI : Efficient construction of a complete index for pan-genomics read alignment. *Journal of Computational Biology*, 27(4):500–513, 2020.

B. LANGMEAD, C. TRAPNELL, M. POP et S. L. SALZBERG : Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biology*, 10(3):1–10, 2009.

S. LAURENS : *Militer pour la science. Les mouvements rationalistes en France (1930-2005)*. EHESS, 2019.

T. LECROQ et M. SALSON : Sequence indexing. *In From Sequences to Graphs : Discrete Methods and Structures for Bioinformatics*, p. 49–86. John Wiley & Sons, Inc, 2022.

J. LEWIS et T. SPEERS : Misleading media reporting ? the MMR story. *Nature Reviews Immunology*, 3(11):913–918, 2003.

H. LI et R. DURBIN : Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009.

R. LI, Y. LI, K. KRISTIANSEN et J. WANG : SOAP : short oligonucleotide alignment program. *Bioinformatics*, 24(5):713–714, 2008.

A. I. MAARALA, O. ARASALO, D. VALENZUELA, V. MÄKINEN et K. HELJANKO : Distributed hybrid-indexing of compressed pan-genomes for scalable and fast sequence alignment. *Plos one*, 16(8):e0255260, 2021.

V. MÄKINEN, G. NAVARRO, J. SIRÉN et N. VÄLIMÄKI : Storage and retrieval of highly repetitive sequence collections. *Journal of Computational Biology*, 17(3):281–308, 2010a.

V. MÄKINEN, N. VÄLIMÄKI, A. LAAKSONEN et R. KATAINEN : Unified view of backward backtracking in short read mapping. *In* T. ELOMAA, H. MANNILA et P. ORPONEN, éds : *Algorithms and Applications*, p. 182–195. Springer, 2010b.

U. MANBER et G. MYERS : Suffix arrays : a new method for on-line string searches. *In SODA '90 : Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, p. 319–327, Philadelphia, PA, USA, 1990. Industrial and Applied Mathematics.

C. MARCHET, C. BOUCHER, S. J. PUGLISI, P. MEDVEDEV, M. SALSON et R. CHIKHI : Data structures based on k-mers for querying large collections of sequencing data sets. *Genome Research*, 31(1):1–12, 2021.

C. MARCHET, Z. IQBAL, D. GAUTHERET, M. SALSON et R. CHIKHI : REINDEER : efficient indexing of k-mer presence and abundance in sequencing datasets. *Bioinformatics*, 36:i177–i185, 2020.

C. MARCHET et A. LIMASSET : Scalable sequence database search using partitioned aggregated Bloom comb trees. *Bioinformatics*, 39(Supplement_1):i252–i259, 2023.

J. MAURER, E. THIEL, W.-D. LUDWIG, J. JANSSEN, C. BARTRAM, B. HEINZE, J. van DENDEREN, U. AYDEMIR, C. FONATSCH, J. HARBOTT *et al.* : Detection of chimeric BCR-ABL genes in acute lymphoblastic leukaemia by the polymerase chain reaction. *The Lancet*, 337(8749): 1055–1058, 1991.

D. MELTON : Reading tangled RNA sequences. *Nature*, 497, 2013.

M. MORGAN et J. SHANAHAN : The state of cultivation. *Journal of broadcasting & electronic media*, 54(2):337–355, 2010.

W. MÜLLER et H.-H. ALTHAUS : Dnaplot program. Institute for Genetics, Universtiry of Cologne. URL https://web.archive.org/web/19980123163746/http://www.genetik.uni-koeln.de/dnaplot/dnaplot.html.

K. MURPHY et C. WEAVER : *Janeway's immunobiology*. Garland science, 2016.

G. NAVARRO : Indexing highly repetitive collections. *In IWOCA*, 2012.

G. NAVARRO : Indexing highly repetitive string collections, part II : Compressed indexes. *ACM Computing Surveys*, 54(2):26 :1–26 :32, 2021.

G. NAVARRO et V. MÄKINEN : Compressed full-text indexes. *ACM Computing Surveys (CSUR)*, 39(1):2–es, 2007.

L. OHM-LAURSEN, M. NIELSEN, S. R. LARSEN et T. BARINGTON : No evidence for the use of DIR, D–D fusions, chromosome 15 open reading frames or VHreplacement in the peripheral repertoire was found on application of an improved algorithm, JointML, to 6329 human immunoglobulin H rearrangements. *Immunology*, 119:265–277, 2006.

K. OWENS : The passivists : Managing risk through institutionalized ignorance in genomic medicine. *Social Science & Medicine*, 294:114715, 2022.

F. PANE, M. INTRIERI, C. QUINTARELLI, B. IZZO, G. C. MUCCIOLI et F. SALVATORE : BCR/ABL genes and leukemic phenotype : from molecular mechanisms to clinical correlations. *Oncogene*, 21(56):8652–8667, 2002.

R. PATRO, G. DUGGAL, M. I. LOVE, R. A. IRIZARRY et C. KINGSFORD : Salmon provides fast and bias-aware quantification of transcript expression. *Nature methods*, 14(4):417–419, 2017.

N. PHILIPPE, M. SALSON, T. COMMES et E. RIVALS : CRAC : an integrated approach to the analysis of RNA-seq reads. *Genome Biology*, 14:R30, 2013.

N. PHILIPPE, M. SALSON, T. LECROQ, M. LÉONARD, T. COMMES et E. RIVALS : Querying large read collections in main memory : a versatile data structure. *BMC bioinformatics*, 2011.

T. B. POLAK, J. van ROSMALEN, S. DIRVEN, J. K. HERZIG, J. CLOOS, S. MESHINCHI, K. DÖHNER, J. J. JANSSEN et D. G. CUCCHI : Association of FLT3-internal tandem duplication length with overall survival in acute myeloid leukemia : a systematic review and meta-analysis. *Haematologica*, 107(10):2506–2510, 2022.

C. PREUDHOMME, L. CHAMS-EDDINE, C. ROUMIER, N. DUFLOS-GRARDEL, C. DENIS, A. COSSON et P. FENAUX : Detection of BCR-ABL transcripts in chronic myeloid leukemia (CML) using an in situ RT-PCR assay. *Leukemia*, 13(5):818–823, 1999.

D. K. RALPH et F. A. MATSEN IV : Consistency of VDJ rearrangement and substitution parameters enables accurate B cell receptor sequence annotation. *PLoS computational biology*, 12 (1):e1004409, 2016.

A. R. REES : Understanding the human antibody repertoire. *MAbs*, 12(1):1729683, 2020.

J. REN, X. BAI, Y. Y. LU, K. TANG, Y. WANG, G. REINERT et F. SUN : Alignment-free sequence analysis and applications. *Annual Review of Biomedical Data Science*, 1(1):93–114, 2018.

T. ROCHER, M. GIRAUD et M. SALSON : Indexing labeled sequences. *PeerJ CS*, 4(e148), 2018.

M. P. ROWE, B. M. GILLESPIE, K. R. HARRIS, S. D. KOETHER, L.-J. Y. SHANNON et L. A. ROSE : Redesigning a General Education Science Course to Promote Critical Thinking. *CBE—Life Sciences Education*, 14(3):ar30, 2015.

B. T. RUTJENS, N. SENGUPTA, R. v. der LEE, G. M. van KONINGSBRUGGEN, J. P. MARTENS, A. RABELO et R. M. SUTTON : Science Skepticism Across 24 Countries. *Social Psychological and Personality Science*, 2021.

M. SALSON, A. CAILLAULT, M. DUEZ, Y. FERRET, A. FIEVET, M. KOTROVA, F. THONIER, P. VILLARESE, S. WAKEMAN, G. WRIGHT et M. GIRAUD : A dataset of sequences with manually curated V(D)J designations. RepSeq workshop, 2016.

M. SALSON, M. GIRAUD, A. CAILLAULT, N. GRARDEL, N. DUPLOYEZ, Y. FERRET, M. DUEZ, R. HERBERT, T. ROCHER, S. SEBDA, S. QUIEF, C. VILLENET, M. FIGEAC et C. PREUDHOMME : High-throughput sequencing in acute lymphoblastic leukemia : Follow-up of minimal residual disease and emergence of new clones. *Leukemia Research*, 53:1–7, 2017.

S. A. SHIRYEV et R. AGARWALA : Indexing and searching petabyte-scale nucleotide resources. *bioRxiv*, 2023.

A. SHLEMOV, S. BANKEVICH, A. BZIKADZE et Y. SAFONOVA : New algorithmic challenges of adaptive immune repertoire construction. *In RECOMBSeq*, 2016.

G. SIRUGO, S. M. WILLIAMS et S. A. TISHKOFF : The missing diversity in human genetic studies. *Cell*, 177(1):26–31, 2019.

G. S. C. SLATER et E. BIRNEY : Automated generation of heuristics for biological sequence comparison. *BMC bioinformatics*, 6:1–11, 2005.

B. SOLOMON et C. KINGSFORD : Fast search of thousands of short-read sequencing experiments. *Nature biotechnology*, 34(3):300–302, 2016.

M. M. SOUTO-CARNEIRO, N. S. LONGO, D. E. RUSS, H.-w. SUN et P. E. LIPSKY : Characterization of the Human Ig Heavy Chain Antigen Binding Complementarity Determining Region 3 Using a Newly Developed Software Algorithm, JOINSOLVER. *The Journal of Immunology*, 172(11):6790–6802, 2004.

N. STOLER et A. NEKRUTENKO : Sequencing error profiles of Illumina sequencing instruments. *NAR Genomics and Bioinformatics*, 3(1):lqab019, 2021.

S. TONEGAWA : Somatic generation of antibody diversity. *Nature*, 302(5909):575–581, 1983.

D. VALENZUELA : CHICO : A compressed hybrid index for repetitive collections. *In 15th International Symposium on Experimental Algorithms*, num. 9685, p. 326–338. Springer, 2016.

D. VALENZUELA et V. MÄKINEN : CHIC : a short read aligner for pan-genomic references. *bioRxiv*, p. 178129, 2017.

P. VILLARESE, C. ABDO, M. BERTRAND, F. THONIER, M. GIRAUD, M. SALSON et E. MACINTYRE : One-Step Next-Generation Sequencing of Immunoglobulin and T-Cell Receptor Gene Recombinations for MRD Marker Identification in Acute Lymphoblastic Leukemia. *In* L. A.W., éd. : *Immunogenetics*, vol. 2453, p. 43–59. Methods in molecular biology, 2022.

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

J. M. VOLPE, L. G. COWELL et T. B. KEPLER : SoDA : implementation of a 3D alignment algorithm for inference of antigen receptor recombinations. *Bioinformatics*, 22(4):438–444, 2006.

C. VROLAND, M. SALSON, S. BINI et H. TOUZET : Approximate search of short patterns with high error rates using the 01^*0 lossless seeds. *Journal of Discrete Algorithms*, 37:3–16, 2016.

C. VROLAND, M. SALSON et H. TOUZET : Lossless seeds for searching short patterns with high error rates. *In International Workshop on Combinatorial Algorithms*, p. 364–375, 2014.

X. WANG, D. WU, S. ZHENG, J. SUN, L. TAO, Y. LI et Z. CAO : Ab-origin : an enhanced tool to identify the sourcing gene segments in germline for rearranged antibodies. *BMC Bioinformatics*, 9(12):S20, 2008.

D. WEESE, M. HOLTGREWE et K. REINERT : RazerS 3 : Faster, fully sensitive read mapping. *Bioinformatics*, 28(20):2592–2599, 2012.

P. WEINER : Linear pattern matching algorithm. *In 14th IEEE Symp. on Switching and Automata Theory*, p. 1–11, 1973.

X. WU, X. FENG, X. ZHAO, F. MA, N. LIU, H. GUO, C. LI, H. DU et B. ZHANG : Prognostic significance of FLT3-ITD in pediatric acute myeloid leukemia : a meta-analysis of cohort studies. *Molecular and cellular biochemistry*, 420(1):121–128, 2016.

J. YE, N. MA, T. L. MADDEN et J. M. OSTELL : IgBLAST : an immunoglobulin variable domain sequence analysis tool. *Nucleic Acids Research*, 41(W1):W34–W40, 2013.

D. YUAN, X. HE, X. HAN, C. YANG, F. LIU, S. ZHANG, H. LUAN, R. LI, J. HE, X. DUAN, D. WANG, Q. ZHOU, S. GAO et B. NIU : Comprehensive review and evaluation of computational methods for identifying FLT3-internal tandem duplication in acute myeloid leukaemia. *Briefings in Bioinformatics*, 2021.

A. ZAKAI : Emscripten : an LLVM-to-JavaScript compiler. *In Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion*, p. 301–312, 2011.

C. ZHANG, B. ZHANG, L.-L. LIN et S. ZHAO : Evaluation and comparison of computational tools for RNA-seq isoform quantification. *BMC Genomics*, 18(1):583, 2017.

Résumé

Alors que les technologies de séquençage de l'ADN et de l'ARN se démocratisent, les données produites par les laboratoires se démultiplient. Les besoins de méthodes capables de traiter ces données, sans nécessiter le recours à des infrastructures de calcul gigantesques, est alors criant.

Mes recherches, comme maître de conférences dans l'équipe de bioinformatique Bonsai depuis 2010, ont eu pour but d'offrir la capacité aux équipes intéressées de tirer parti, à moindre coût, de ces données produites. Deux axes principaux de recherche correspondent à mes travaux sur la période afin de remplir cet objectif.

L'un de ces axes consiste à proposer des méthodes de comparaison qui évitent de recourir à de coûteuses étapes d'alignement, il s'agit des méthodes sans alignement. Avec mes collègues, j'ai proposé des méthodes d'analyse sans alignement, efficaces en temps et en espace qui fournissent des résultats pertinents pour les recherches ou les soins en oncologie. L'un de ces projets, Vidjil, est un logiciel désormais utilisé par des dizaines d'hôpitaux autour du monde dont certains financent deux ingénieurs qui continuent à maintenir et améliorer le logiciel.

L'autre axe correspond à l'indexation des données de séquençage avec l'objectif de faciliter leur exploitation, voire leur réexploitation. L'indexation de données est une étape indispensable afin d'obtenir rapidement des informations recherchées. Néanmoins, le type d'information qui peut être recherché dépend de la nature de l'indexation. Les méthodes présentées sont de différentes natures : certaines visent à offrir des recherches plus souples, tandis que d'autres ont pour objectif de favoriser la réexploitation de données existantes.

Enfin, j'aborde également succinctement mes activités de diffusion de la culture scientifique auprès du grand public ou dans le cadre de formations de journalisme.