Caractérisation et détection d'insertions constitutionnelles de grande taille dans le cadre d'un usage médical

Wesley Delage

UNIVERSITE BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1
COMUE UNIVERSITÉ BRETAGNE LOIRE

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Info

Par
Wesley DELAGE

Caractérisation et détection d'insertions constitutionnelles de grande taille dans le cadre d'un usage médical

Thèse présentée et soutenue à Rennes, le 11 décembre 2020
Unité de recherche : Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), UMR 6074
Thèse N° :

Rapporteurs avant soutenance :

SCHLUTH-BOLARD, Caroline	Maîtresse de conférence - Praticienne hospitalière	HDR, Université de Lyon
VIARI, Alain	Directeur de recherche	Inria, Centre Grenoble Rhône-Alpes

Composition du Jury :

Président :	DAMERON, Olivier	Professeur d'Université	Irisa, HDR, Université de Rennes
Examinateurs :	CHIKHI, Rayan	Chargé de recherche	CNRS, Institut Pasteur, Paris
	DAMERON, Olivier	Professeur d'Université	Irisa, HDR, Université de Rennes
Invitée :	DE TAYRAC, Marie	Maîtresse de conférence - Praticienne hospitalière	IGDR, Université de Rennes
Directrice de thèse :	LEMAITRE, Claire	Chargée de recherche	IRISA/Inria, Rennes
Co-directeur de thèse :	THEVENON, Julien	Professeur d'Université - Praticien hospitalier	Inserm, HDR, Université de Grenoble

Remerciements

De tout temps les femmes et les hommes réalisant une thèse s'adonnent à remercier les personnes ayant jouées un rôle durant les trois années de leur thèse (ou parfois plus). Bien que nous soyons les personnes ayant rédigées et défendues une thèse, son accomplissement ne peut être dûe à un seul individu, à un seul doctorant.

Il est ainsi évident et néanmoins difficile de remercier la totalité des personnes qui ont contribué de près ou de très loin à cette thèse. Comme nombre de confrères et consoeurs qui ont réalisé des remerciements avant moi, je m'excuse auprès des personnes que j'aurais oublié dans ces remericements. Cet oubli n'est pas une offense personnelle à l'exception de ceux ou celles qui se poseront la question. Même sans votre nom dans cette section, je vous autorise à vous octroyer un fragment de cet accomplissement car, sans l'ensemble de ces facteurs (et factrices, celles-ci est pour vous Arnaud et Lucas), cette thèse n'aurait pas aboutie.

Les premières personnes que je souhaite remercier sont celles qui m'ont donné l'opportunité de réaliser cette thèse.

A Claire Lemaitre, qui a eu l'audace et le courage de m'accepter en tant que doctorant. Je sais que mon esprit de contradiction systématique et borné n'ont pas toujours facilité le déroulement de la thèse. J'ai beaucoup apprécié travailler avec toi et te remercie de m'avoir appris les rouages de la recherche dans un esprit de bienséance. Je remercie l'exemple que tu as été à travers ta rigueur dans la recherche ainsi que ta volonté systématique de comprendre tout et dans le moindre détail.

A Julien Thevenon, qui s'est vu chargé d'un doctorant et d'un rôle de codirecteur en supplément de toute les casquettes qu'il portes déjà (tu pourrais ouvrir une boutique de chapeau si tu le souhaites une fois le confinement fini). Malgrè la distance, ta présence et ton encadrement ont eu un grand impact qui nous a permis de garder les pieds sur terre avec Claire. J'ai compris grâce à toi l'importance de connecter les différents acteurs, biologistes, bioinformaticiens et informaticiens pour parvenir à faire avancer la recherche.

La personne qui se doit d'être remerciée ensuite est la personne que l'on peut entendre dans les bureaux de l'INRIA et ce même si vous avez un casque anti-bruit. Je ne parle pas de l'alarme à incendie dont le son est couvert par la voix de la personne que je m'apprête à remercier. A Méline, collègue et soeur siamoise (non siamoise avant 10h30 car elle dort) de thèse. Soyons

honnête, heureusement que nous étions là, heureusement que tu étais là (on calme l'égo par contre). Il y aurait beaucoup trop de private joke, de clin d'oeil à citer que ce manuscrit ferait pâle figure à côté. Donc Mel assieds toi faut que je te parle, on est arrivé à la fin d'une histoire. C'est le début d'une tout autre aventure, j'espère qu'elle contiendra tout ce que tu souhaites.

A Hugo aka Dieu dans le milieu, collègue de bureau qui a réussi à me supporter malgrè mes interprétations musicales et mes besoins d'interactions incessantes. J'admets être responsable de ton manque de productivité (pardon François) à certains moment de notre thèse. Mais il faut reconnaître qu'il était important d'interagir ensemble pour identifier quelle fenêtre était encore fonctionnelle ou laquelle allait tomber en première.

A Lolita, première cunicultrice de France et prophète à ses heures. J'ai apprécié les soirées que tu organisais et auxquelles tu ne venais pas. Nous te remercions tous, des trois commandements du thésard :

1. Une extension de ta thèse tu ne seras pas
2. Aux autres tu ne te compareras pas
3. A l'autocensure tu ne te soumettras pas

Ces derniers se sont montrés importants lors des périodes de démotivation pendant la thèse.

Je (ne) remercie (pas) Arnaud et Cervin d'avoir abusé de ma naïveté. Je (ne) remercie (pas non plus) ce compère parti trop tôt, Lucas, qui mit aux côtés de toi Arnaud, ont rendu toutes les conversations illogiques et incompréhensibles sauf pour vous. Merci à Marine de nous avoir initié à la culture pop' durant nos déjeuners. Merci à Pierre de m'avoir fait découvrir le LIIT et de vivre sa soutenance en mode survivaliste, à Rouen, le jour de l'explosion de Lubrisol.

Un merci que dis-je une reconnaissance éternelle pour Marie sans qui nous ne serions rien. Ta capacité à résoudre les problèmes, à nous expliquer l'univers administratif, combinée à ta personnalité enjouée ont été d'un grand soutien durant cette thèse. Blanche Neige peut se reposer car ses nains sont devenus grands !

Merci à Catherine de ne pas avoir mis Méline dans le même bureau que le mien, sans quoi notre thèse respective n'aurait pas abouti. À Stéphanie d'avoir été là pour partager son expérience, proposer son aide et ses conseils durant la thèse. À Pierre P, lobbyiste-né de l'escalade. Pour les futurs doctorants prenez garde à ne pas vous faire embrigader ! On essaye et après on ne s'en passe plus pour certains, une vraie secte ! À Anthony et à son attitude "anti-thèsard" qui ne fonctionne pas si bien que ça (au final tu nous aimes bien, on le sait, mais chut on ne dira rien). À Dominique qui s'est montré présent lorsqu'il y avait besoin d'un regard extérieur durant la thèse. À Emmanuelle et Olivier qui m'ont appris beaucoup sur l'enseignement à travers leurs conseils et vécus.

Le dernier remerciement est évidemment celui qui vient du coeur et est destiné à la personne qui m'a accompagné tout au long de cette période de thèse. À Tiphaine qui s'est montrée d'une patience et d'une aide qui sans quoi l'aboutissement de cette thèse aurait été difficile. Tu es et resteras un exemple vers lequel je tendrais pour me permettre à mon tour d'être la personne qui te soutiendra et t'aidera dans les épreuves que tu rencontreras dans le futur.

A l'image de Snoopy D., je me remercie moi-même d'avoir réalisé cette thèse, non pas pour prouver quelque chose à quelqu'un mais tout simplement parce que la recherche c'est marrant.

TABLE DES MATIÈRES

Nomenclature

ADN : acide désoxyribonucléique
ARN : acide ribonucléique
BAM : Binary Alignment/MAP
BER : base-excision repair
CGH : Comparative Genomic hybridization
CNV : copy number variation
DSBR : double-strand break repair
DupDispers : duplication dispersée
DupSeg : duplication segmentaire
DupTandem : duplication en tandem
FoSTeS : fork stalling and template switching
Indel : insertion/délétion (inférieure à 50 paires de bases)
ME : élément mobile
MMEJ : microhomology-mediated end joining
MMR : mismatch repair
NAHR : non-allelic homologous recombination
NER : nucleotide excision repair
NHEJ : non-homologous end joining
OLC : Overlap Layout Consensus
ONT : Oxford Nanopores Technologies
PacBio : Pacific Biosciences
PCR : polymerase chain reaction
Pb : paire de base
RepTandem : répétition en tandem
SAM : Sequence Alignment/MAP
SDSA : synthesis-dependant strand annealing
SimpleRep : répétition simple
SMRT : Single Molecule Real Time
SNP : single nucleotide polymorphism
SSA : single-strand annealing
SV : structural variation
VCF : variant call format

Chapitre 1

Introduction

Courant 2016, le premier ministre annonce le Plan France Médecine Génomique 2025. Son objectif est de placer le séquençage de génomes au centre des pratiques cliniques d'ici 10 ans. Ces dernières décennies ont montré l'enjeu de santé public qu'est l'accès à l'information génétique des individus pour comprendre, prévenir et traiter des maladies qui les affectent. Le coût réduit des technologies de séquençage a permis d'ouvrir la voie vers la création de plateformes où le séquençage est industrialisé, le rendant accessible à l'ensemble de la population. Ce plan vise également à homogénéiser, à rendre réplicables et reproductibles les méthodes d'analyses qui sont actuellement hétérogènes entre laboratoires. C'est enfin un enjeu économique auquel veut répondre ce plan, où une médecine plus personnalisée permettra de limiter des tests et médicaments inadaptés avec effets indésirables ou encore d'accélérer les analyses. Cette thèse s'ancre dans ce contexte où l'objectif est d'améliorer notre compréhension de la détection de variations génétiques pour permettre l'élaboration de meilleures analyses.

Ce chapitre d'introduction vise à établir le contexte de cette thèse. Dans un premier temps, nous aborderons le contexte biologique qui exposera la structure du génome, l'origine des variations génétiques, ainsi que leur rôle et leur impact sur notre modèle d'étude qu'est le génome humain. Nous explorerons ensuite les techniques qui ont permis l'accès à l'information génétique, ainsi qu'à la production des premiers génomes de références humains. Nous poursuivrons par la présentation des méthodes et des protocoles utilisés pour détecter les variations génétiques dans un usage médical. Enfin, nous présenterons comment les limites identifiées dans ces méthodes ont conduit à la mise en place de la problématique et des différents objectifs de la thèse.

1.1 Le génome et les variations génétiques

1.1.1 Le génome, support de l'information génétique

L'acide désoxyribonucléique (ADN) est le support de l'information génétique héréditaire de l'ensemble des organismes vivants, mais également de structures particulières comme les virus. Cet homopolymère est composé de quatre acides nucléiques, chacun composé d'une base nucléique, d'un pentose et d'un groupe phosphate. Les acides nucléiques se différencient par leur base azotée qui sont l'adénine (A), la cytosine (C), la guanine (G) et la thymine (T). Chaque nucléotide est capable de se lier de manière spécifique à un autre nucléotide à travers des liaisons hydrogènes (H), A se liant avec T (2 liaisons H) et G avec C (3 liaisons H). Une chaine de nucléotides, appelée brin d'ADN, est constituée d'une extrémité possédant un groupe hydroxyl libre, notée 3', et d'une extrémité avec un groupe phosphoryl libre, notée 5'. Cette asymétrie donne une orientation à l'ADN, dont la lecture se fait de l'extrémité 5' vers l'extrémité 3'. L'ADN est composé de deux brins d'ADN qui s'hybrident par paire, attachés par des liaisons hydrogènes. Chaque brin possède la même longueur et forment ensemble une structure hélicoïdale. Les brins sont complémentaires et antiparallèles, chaque brin est donc le complément inverse de l'autre.

L'enchaînement des différents nucléotides forme un texte qui contient des informations permettant le fonctionnement de la cellule et de l'organisme. Une partie de ce texte est constituée d'éléments, appelés gènes, dont la transcription permet la synthèse d'acides ribonucléiques (ARN). La traduction de l'ARN conduit à la synthèse de protéines, où chaque trinucléotide de l'ARN forme un codon qui code pour un acide aminé. Les gènes sont divisables en sections codantes et non codantes appelées respectivement exons et introns. Lors de la transcription standard d'un gène, les introns sont épissés conduisant les ARNs à être composés uniquement d'exons. Un mécanisme d'épissage alternatif existe pouvant induire la rétention ou l'excision d'introns ou d'exons résultant à la synthèse de protéines structurellement différentes. L'ADN est également composé d'autres informations qui ne sont pas des gènes mais qui permettent l'expression ou la répression des gènes.

Cette molécule d'ADN est stockée différemment selon que l'organisme soit eucaroyte ou procaryote. Les eucaroytes (*eu* : bien/vrai et *karuon* : noyau) stockent leur ADN dans un compartiement cellulaire appelé noyau, ce qui n'est pas le cas des procaryotes (*pro* avant et *karuon* : noyau). Au sein de ce noyau, l'ADN est extrêmement condensé via un mécanisme de compaction. Ce dernier est basé sur un enroulement de l'ADN autour de structures à base d'histones. L'ensemble d'un double brins d'ADN est appelé chromosome et l'ensemble des chromosomes

d'une cellule est appelé génome (Figure 1.1). Les organismes peuvent être caractérisés par le nombre de copies de chromosomes qu'ils portent. Un organisme haploïde possède une seule copie de chaque chromosome, tandis qu'un organisme avec un nombre plus important de copie est qualifié d'organisme polyploïde. L'être humain est par exemple un organisme diploïde, portant deux copies de chaque chromosome.

L'être humain est un eucaryote qui possède 23 paires de chromosomes, où chaque chromosome est composé de deux chromatides. Chaque chromatide d'une paire de chromosome est hérité de chaque parent de l'individu. Ces 23 paires de chromosomes sont divisées en deux types : 22 paires d'autosomes et une paire qualifiée de chromosomes sexuels (XX pour une femme et XY pour un homme). Ces paires de chromosomes possèdent trois grandes régions répétées, une à chaque extrémité du chromosome appelée télomère et une au centre appelée centromère. L'ensemble de ces chromosomes représente environ 3,2 milliards de paires de bases (pb) et seulement 2% sont occupés par des exons. Le nombre de gènes codant pour des protéines est estimé à 20 000, mais le mécanisme d'épissage alternatif augmente considérablement le nombre de protéines pouvant être synthétisées. Il est est ainsi estimé que plus de 84 000 protéines pourraient être synthétisées chez l'homme[41].

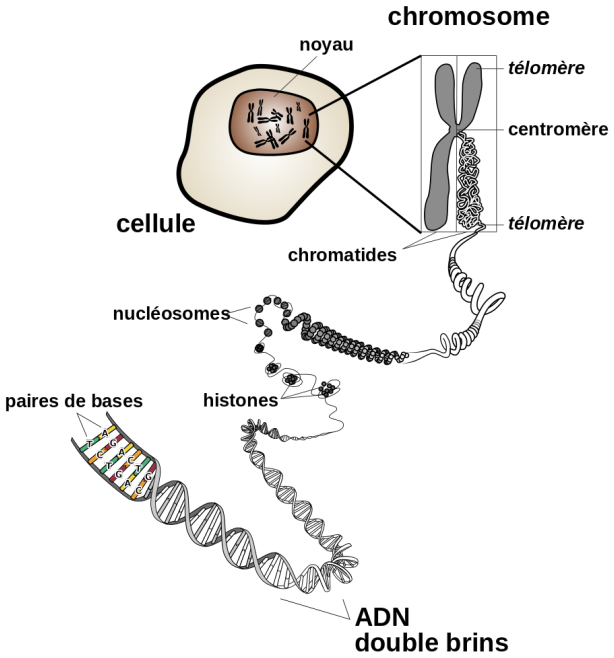

FIGURE 1.1 – Structure du génome des eucaryotes. Figure adaptée de [27].

Le génome humain ne figure pas parmi les plus gros génomes ou ceux ayant le plus de

gènes. Par exemple, la plante modèle *Arabidposis thaliana* dont le génome n'est que de 135 millions de paires de bases contient 27 655 gènes[22]. L'être humain n'est également pas le génome de vertébré le plus gros identifié à ce jour. Ce titre revient au Protoptère éthiopien (*Protopterus aethiopicus*) avec un génome de 130 milliards de paires de bases. Néanmoins, le génome humain reste un génome complexe dont plus de la moitié est associé à des fragments répétées ou dupliqués. Ces régions répétées seront au coeur de la problématique de la thèse.

1.1.2 Définitions des variations génétiques

Chaque individu partage des caractères phénotypiques avec un autre individu de son espèce, tout en possédant des caractères qui lui sont propres. Cette variabilité phénotypique peut être associée à une variabilité dans le génome des individus, appelée variabilité génétique. Les variations génétiques sont définies comme des portions d'ADN différentes entre deux entités (cellules, organismes, population) à une même position du génome (locus), dont l'une est identifiée comme référence. Différents types de variations sont identifiables selon le type cellulaire dans lequel elles se produisent, selon la taille de ces variations ou encore selon la nature de la variation.

Type cellulaire.

Deux types de variations peuvent être caractérisées selon le type cellulaire dans lequel elles se produisent. Les variations dites somatiques sont décrites comme des variations portées par une ou quelques lignées cellulaires d'un organisme. Les variations germinales quant à elle, trouvent leur origine dans les cellules germinales et seront portées par toutes les cellules des descendants de l'individu.

La taille.

Les variations impliquant le changement d'une base par une autre sont appelés des SNP (single nucleotide polymorphism). Entre 1 et 50 paires de bases, les variations sont décrites comme indel (concaténation de insertion et délétion). Au dessus de 50 paires de bases, les variations sont qualifiées de variations de structure (SV)[3]. La plus récente estimation du nombre d'indel et de variations de structure par génome s'élève à un ordre de 10^6 et 10^5 respectivement[17]. Malgré leur occurence plus faible que les indel et les SNP, les variations de structure restent les variations modifiant le plus de paires de bases dans le génome[34] (Table 1.1).

Classe	Taille en paires de bases	Quantité par génome	Taille de la région affectée en Mpb	Pourcentage du génome
SNP	1	4 000 000-5 000 000	4-5	0,078
Indel	1–49	700 000–800 000	3–50	0,069
SV	>50	23 000–28 000	10–12	0,19
Inversions	>50	153	23	0,397
CNV	>1000	environ 500	12–15	0,232

TABLE 1.1 – Caractéristiques des variations génétiques humaines. Table adaptée de [34]. CNV : variation du nombre de copies, SV : variation de structure.

La nature.

Plusieurs définitions de variations ont été développées en fonction de leur nature et sont séparées en deux grands types (Figure 1.2). Le premier grand type, appelé équilibré, regroupe l'ensemble des variations qui ne modifient pas la quantité de nucléotides dans un génome. Les SNP représentent la modification d'une base par une autre. Les inversions représentent une variation dont la séquence est inversée chez un individu en comparaison avec un génome de référence. La translocation réciproque décrit l'échange d'un fragment avec un autre fragment, tandis que la transposition décrit le déplacement d'un fragment vers un autre locus.

Le second grand type, appelée déséquilibré, correspond aux variations qui modifient la quantité en nucléotides conduisant à un gain ou à une perte de nucléotides par rapport à un génome de référence. Les délétions sont définies comme l'absence d'une séquence d'ADN chez un individu par rapport à un individu de référence. Son opposé, l'insertion est la présence d'une séquence d'ADN présente chez l'individu qui est absente chez l'individu de référence, au même locus. Les insertions sont divisibles en plusieurs sous types selon la nature de l'insertion. Les nouvelles insertions ou *de novo* sont des insertions dont la séquence n'est pas présente dans le génome de référence. Les duplications impliquent l'insertion d'une copie d'une séquence déjà présente dans le génome de référence. La duplication en tandem implique que le site de duplication est à côté de la séquence copiée, alors que la duplication dispersée s'insère à une autre position. Le type de variant appelé répétition en tandem est un cas particulier de la duplication en tandem puisqu'il s'agit d'une duplication en très grand nombre d'une séquence (graine) les unes à côté des autres[3].

Le chromothripsis est un type de variation induisant une fragmentation en plusieurs segments d'un chromosome qui sont ensuite réarrangés lors de la réparation de cette fragmentation. Cette réparation peut être imparfaite, changeant l'ordre des segments ou omettant certains frag-

ments. Cette forme de variation a principalement été découverte dans des maladies cancereuses.

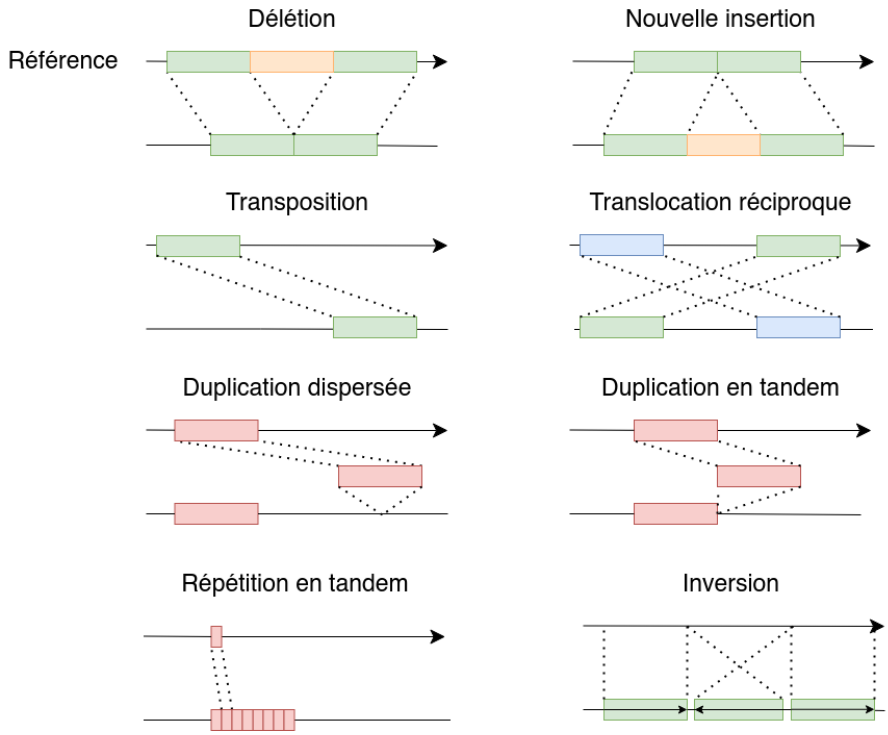

FIGURE 1.2 – Structure des différentes variations de structure. Figure adaptée de [3].

1.1.3 Origine des variations génétiques

Historiquement, le premier mécanisme engendrant de la variabilité génétique d'un individu, décrit par Mendel, concerne l'hérédité et l'indépendance de la transmission de caractères[90]. Un individu peut présenter des variations observées au sein de chaque parent et résulter à de nouvelles caracteristiques phénotypiques propres à l'enfant. Le second mécanisme, décrit par Thomas Hunt Morgan, basé sur les travaux de Frans Alfons Janssens, révèle l'existence de recombinaison entre chromosomes homologues[54]. Cet événement, nommé crossing over, a permis de mettre en lumière la recombinaison de gènes et de caractères. Le troisième mécanisme est associé aux variations qui vont directement altérer le contenu du génome et trouvent leurs origines dans divers événements. Ce dernier prend sa source dans la réparation de l'ADN ayant subit une cassure d'un ou de ses deux brins. Il est important de noter que les mécanismes décrits par la suite impliquent principalement les variants qui ne sont pas des SNP. La génération de SNP résulte principalement en des erreurs lors de réplication de l'ADN.

Les cassures de l'ADN

Le nombre de cassures double brins (DSB) est estimé à dix par jour et par cellule, sur la base des cassures de chromosomes en métaphase et de chromatides dans des fibroblastes primaires d'humains ou de souris[81]. Les cassures simple brin (SSB) quant à elles seraient 10^3 plus fréquentes que les cassures double brins[14]. Les causes induisant la formation de cassures d'un ou des deux brins de l'ADN sont diverses. Des dérivés réactifs de l'oxygène ou ROS (reactive oxygen species) peuvent être produits par la mitochondrie lors de la respiration oxydative. Ces ROS, convertis par des superoxydes dismutases en radicaux libres hydroxylés, peuvent réagir avec l'ADN et provoquer des cassures simples brins[131, 56]. Les radiations ionisantes sont également un facteur pouvant générer des cassures de brins de l'ADN. Environ 300 millions de particules de radiations ionisantes pénètrent à travers chaque personne toute les heures. Ces dernières peuvent générer la formation de radicaux libres, qui mis en contact avec l'ADN, peuvent conduire à des cassures simples et double brins de l'ADN[135]. Des erreurs d'actions de la part d'enzymes nucléaires sont capables de conduire à des cassures de l'ADN. La topoisomérase de type II est une enzyme qui a pour fonction de briser, de façon transitoire, les deux brins d'ADN. Une défaillance dans cette fonction peut conduire à une incapacité à réunir les brins d'ADN préalablement découpés conduisant à une cassure de l'ADN[147]. Enfin des contraintes physiques et mécaniques de l'ADN peuvent également conduire à la formation de cassures de l'ADN.

Afin de préserver l'intégrité de la cellule ainsi que son bon fonctionnement, de multiples mécanismes moléculaires ont été sélectionnés au cours de l'évolution permettant de réparer les cassures de l'ADN. C'est lors de cette réparation que des modifications de l'ADN peut avoir lieu et conduire à la formation de variations génétiques autres que des SNP.

Réparations des cassures simple brin de l'ADN

Le modèle global de réparation des cassures simple brin de l'ADN se divise en trois étapes : une reconnaissance de l'erreur, suivie d'une excision des nucléotides impliqués, puis d'une synthèse des nucléotides en se basant sur le brin complémentaire intact. Les mécanismes biologiques décrits pour appliquer ce modèle sont multiples. La réparation par excision de base (*base-excision repair*, BER) est par exemple décrite pour réparer les cassures induites par des nucléotides endommagés, appelés AP site (apurinic/apyrimidinic site)[14]. Cette réparation se limite uniquement aux AP sites détectés et retirés par des glycosylases d'ADN. Les radiations sont capables d'endommager un ensemble de nucléotides d'un brin d'ADN qui ne sont pas ré-

solvables avec la réparation par excision de base. Cette réparation est réalisée par le mécanisme de réparation par excision de nucléotide (NER). Lorsque ce type de dommage est reconnu, 12 à 24 nucléotides en amont et en aval de la région endommagée sont retirés pour permettre la synthèse du brin. Enfin des mésappariements de nucléotides entre brins peuvent avoir lieu lors de la réplication de l'ADN. Un mésappariement correspond à la présence d'un nucléotide qui n'est pas complémentaire du nucléotide du second brin (A/C, A/G, T/C, T/G). La réparation de mésappariements (MMR) met en jeu de nombreuses protéines de reconnaissance et de lyse de l'ADN, appartenant à la famille Mut. Cette réparation conduit également au découpage d'un des brins qui est par la suite resynthétisé en se basant sur le brin complémentaire. L'ensemble de ces réparations permettent une résolution précise et rapide des dommages causés à l'ADN. Elles ne se présentent pas comme le mécanisme générant le plus de variants de structure. Néanmoins des conséquences sont observées lorsque la fonction des protéines impliquées dans la réparation est altérée. La mutation de *SCAN1* ou de *AOA1* conduisent à une altération de la fonction de réparation de cassures simple brins impliquant un dysfonctionnement neurologique progressif des individus portant cette mutation[14].

Réparations des cassures double brins de l'ADN

Recombinaison homologue

La recombinaison homologue est définie comme un échange de nucléotides entre chromosomes homologues pouvant avoir lieu avant l'entrée de la cellule en mitose, peu de temps après la réplication de l'ADN[2]. Cette période est propice à la réparation de l'ADN puisque la chromatine soeur formée suite à la réplication de l'ADN est facilement disponible. Deux modèles ont été développés afin d'expliquer comment la recombinaison homologue permet la réparation de cassures de l'ADN[126]. Les premières étapes de chaque modèle sont similaires : le complexe MRN va se lier à l'ADN au niveau de la coupure, puis un découpage va avoir lieu de 5' vers 3' pour chaque brin d'ADN cassé via les helicases Sgs1 ainsi que les nucléases Exo1 et Exo2. Des filaments de nucléoprotéines vont rapprocher les brins d'ADN, dont le contenu en nucléotide est similaire aux brins cassés. Ces brins intacts vont servir de modèle pour la réparation des brins cassés. L'ADN modèle utilisé est souvent issu du chromosome homologue permettant une réparation quasi à l'identique du fragment perdu (Figure 1.3). Mais il est toutefois possible que l'ADN provienne, non pas du chromosome homologue, mais d'une région d'ADN similaire en terme de séquences nucléotidiques. Il est donc ainsi possible d'observer des variations génétiques dues à ce phénomène.

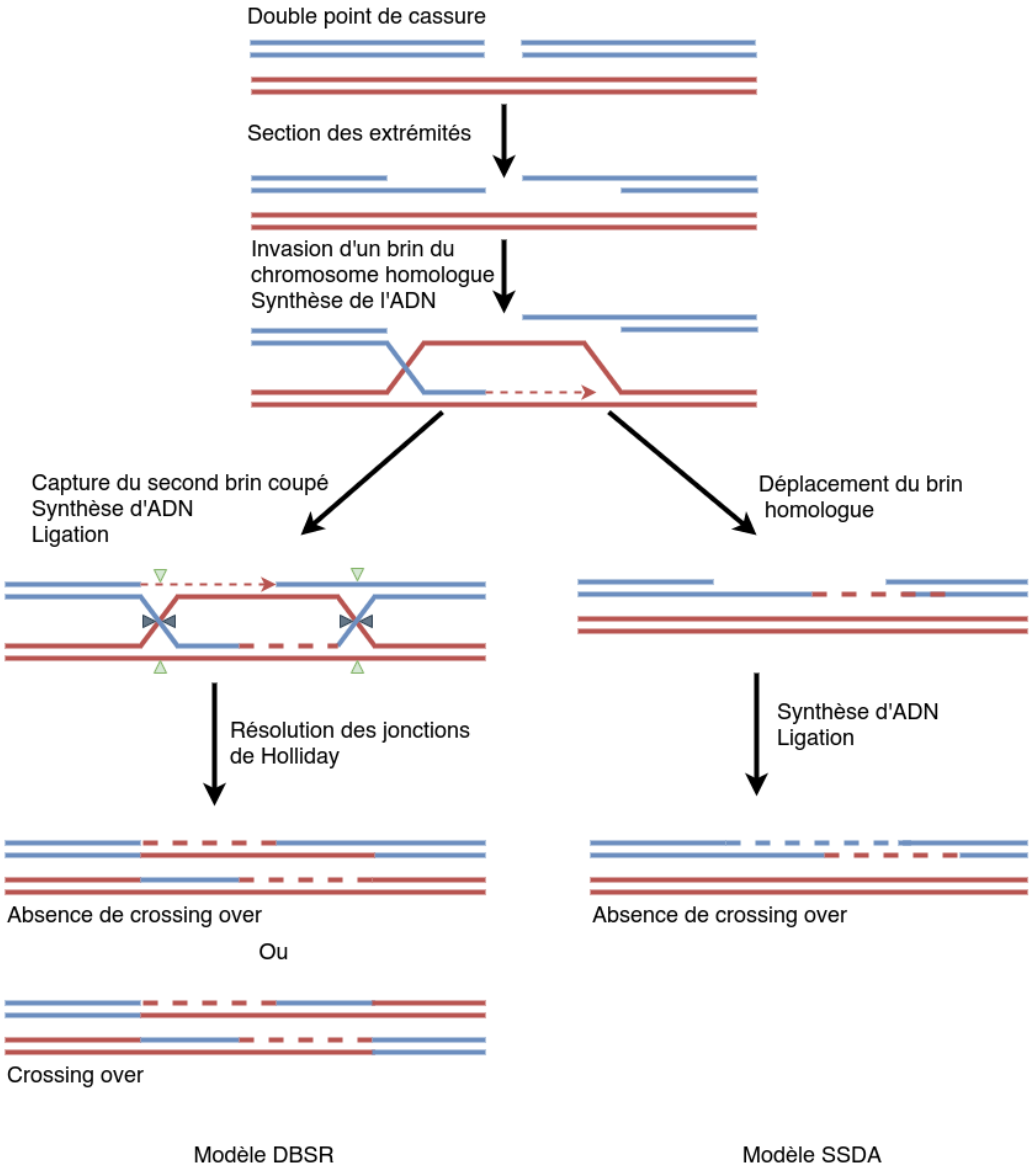

FIGURE 1.3 – Modèle de recombinaison homologue. Figure adaptée de [126]. Les cassures double brins (DSB) peuvent être réparées par plusieurs voies de recombinaison homologues. Deux voies principales sont la réparation des cassures double brins (DSBR) et la *Synthesis-Dependent Strand Annealing* (SDSA). Dans les deux voies, la réparation est initiée par le découpage des extrémités des brins cassés. L'invasion du brin homologue est suivie par la synthèse d'ADN à une extrémité d'un brin cassé. Pour le modèle DSBR, après l'invasion et la synthèse du brin, le deuxième brin cassé peut être capturée pour former des jonctions de Holliday (HJ). Après la synthèse et la ligature de l'ADN manquant, la structure est résolue au niveau des HJ en mode non croisé (pointes de flèches noires) ou croisé (pointes de flèches vertes à une HJ et pointes de flèches noires à l'autre HJ). Pour le modèle SDSA, le brin homologue est éloigné et le brin cassé récemment réparé est utilisé pour réparer le second brin. Ce dernier modèle, contrairement au modèle DBSR, ne permet pas la génération de croisement entre les brins homologues (*crossing over*).

Une boucle de déplacement (D-loop) est formée lors du rapprochement entre le brin cassé en 3' et le chromosome homologue. Une ADN polymérase va ensuite prolonger l'extrémité du brin 3' en synthétisant un nouvel ADN. Cela transforme la boucle D en une structure en forme de croix appelée jonction de Holliday. Suite à cette étape les deux modèles divergent, dans le cas du premier modèle appelé DSBR (Double-Strand Break Repair), une seconde jonction de Holliday se produit. Cette double jonction de Holliday peut être à l'origine du phénomène de crossing over, selon les coupures des brins impliqués dans ces jonctions par les endonucléases (Figure 1.3). Dans le second modèle appelé SDSA (Synthesis-Dependent Strand Annealing), la jonction de Holliday est résolue. Puis l'extrémité d'ADN non syntétisée peut l'être grâce à la séquence de l'autre brin récemment synthétisée (Figure 1.3). Ce modèle ne permet pas la génération de crossing over contrairement au premier modèle.

Recombinaison non allèlique

Un modèle alternatif, appelé recombinaison homologue non allélique (NAHR), a été développé pour expliquer la recombinaison homologue qui n'a pas lieu entre allèles[45]. L'ADN modèle utilisé pour la recombinaison sont des LCR (low copy repeats), appelés également duplications segmentaires. Ces LCR sont des fragments de plusieurs kilobases qui partagent plus de 90% d'identité de séquence avec d'autres copies dans le génome[115]. Dans de plus rares cas, des éléments transposables tels que des éléments *Alu* et des SINE sont impliqués dans ce mécanisme. Une recombinaison déféctueuse de type NAHR peut induire la formation de variations génétiques.

Par exemple, des microdélétions localisés dans le gène *NF1*, dont la cause serait une recombinaison déféctueuse de type NAHR, induisent l'apparition d'une neurofibromatose de type 1[133].

Single-strand annealing

Le modèle single-strand annealing (SSA) est un autre mécanisme de recombinaison homologue. Ce dernier n'utilise pas un chromosome homologue pour réparer l'ADN mais la présence de séquences répétées au niveau des points de cassures. Lors d'une cassure double brins de l'ADN, un rapprochement entre deux séquences répétées, proches du point de cassure a lieu. Les brins "volants" non homologues sont ensuite digérés pour permettre la synthèse et la ligation d'ADN en se basant sur la complémentarité des brins (Figure 1.4 à droite)[117].

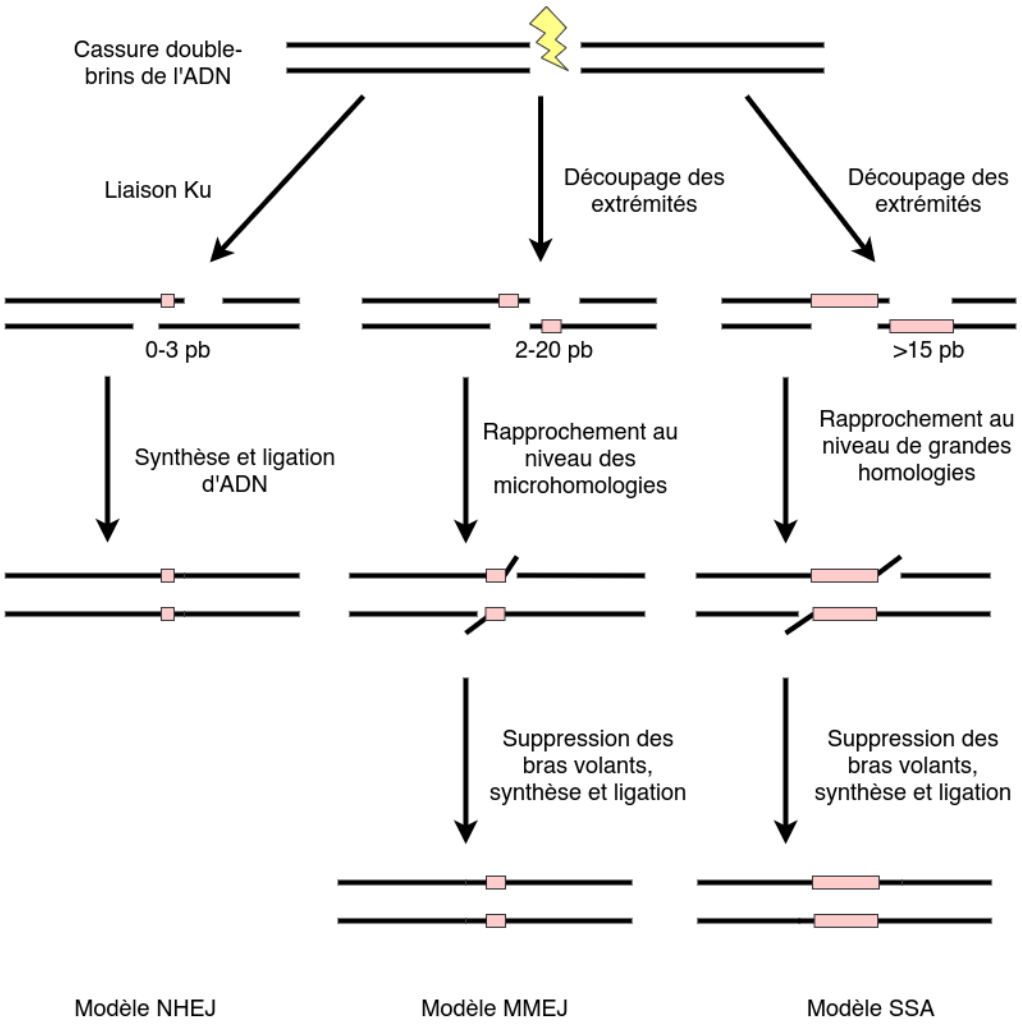

FIGURE 1.4 – Modèle NHEJ, MMEJ, SSA. Figure adaptée de [117]. La NHEJ répare de préférence les cassures de l'ADN avec un redécoupage des brins limité dû à la liaison d'hétérodimères de type Ku qui empêchent ce redécoupage. Des microhomologies (0-3 bp) aident à la juxtaposition des extrémités de l'ADN, ce qui ce qui peut conduire à des délétions/insertions de 1-5 bp aux jonctions de réparation. Contrairement au NHEJ, un redécoupage de l'ADN est réalisé dans les modèles MMEJ et SSA. Une homologie de 2 à 20 pb (MMEJ) ou supérieure à 15 pb (SSA) est utilisée pour rapprocher les deux brins cassés. Dans le MMEJ et le SSA, ce rapprochement est suivi par un découpage des bras d'ADN volants en 3. La synthèse et la ligature de l'ADN, ces mécanismes peuvent conduire à la génération de délétions ou d'insertions de taille variable pour le modèle MMEJ. Seules de grandes délétions peuvent être produites par le modèle SSA.

Non-homologous end joining

Décrit par Moore et Haber en 1996 chez *Saccharomyces cerevisae*, la jonction d'extrémités non homologues ou NHEJ, est un mécanisme de réparation de cassures double brins de l'ADN.

Contrairement à la réparation par recombinaison homologue, la réparation par NHEJ n'utilise pas d'étape de redécoupage des brins cassés, ni d'ADN modèle pour synthétiser les brins d'ADN cassés, mais effectue une ligation directe. Les jonctions résultantes de cette réparation se caractérisent par la présence de petites répétitions (0-3 paires de bases) souvent présentes au niveau des points de cassures et qui sont utilisées pour guider la réparation (Figure 1.4). Cette réparation n'est pas sans erreur et est une source de réarrangements. Des délétions, translocations ainsi que des fusions de télomères ont pu être observées suite à une réparation de l'ADN par ce mécanisme. La cause principale serait la présence de multiples cassures double brins qui permettraient des jonctions intra et interchromosomiques causant des réarrangements[117]. Une mauvaise réparation par NHEJ serait responsable de l'apparition d'une délétion de 3 mégabases dans la région flanquante au gène *NF1*. Cette dernière est également responsable de la neurofibromatose de type 1[133]. Bien que la recombinaison homologue soit la principale source de réparation, la NHEJ reste importante. Des mutations dans les gènes codant pour des protéines impliquées dans le mécanisme NHEJ conduisent à des microcéphalies, déformations faciales et un retard de croissance[140].

Microhomology-mediated end joining

Aussi connu comme un mécanisme alternatif au NHEJ (Alt-NHEJ), le mécanisme de *microhomology mediated end joining* (MMEJ) a été décrit pour la première fois par Nussenzweig et Nussenzweig en 2007[100]. Ce modèle possède néanmoins des caractéristiques propres qui le distinguent du modèle NHEJ. Les tailles des microhomologies comprises entre 2 et 20 paires de bases, sont plus grandes que celles du NHEJ (0-3 paires de bases) mais plus petites que celles du modèle SSA ($>$ 15 paires de bases). Le MMEJ suit de manière similaire le modèle SSA concernant la réparation de l'ADN (Figure 1.4)[117]. Ce mécanisme est suspecté d'être responsable de microdélétion dans le gène *foxl2* conduisant au syndrome BPES (*Blepharophimosis, ptosis, and epicanthus inversus*)[134].

Fork Stalling and Template Switching

Ce modèle a été proposé en 2007 par Lee, Carvalho et Lupski afin d'expliquer la formation de réarrangement complexes, non réccurents, de type délétion et duplication dans le contexte de la maladie de Pelizaeus-Merzbacher (PMD)[73]. La présence non systématique de microhomologies au niveau des points de cassure laissait suggérer à une potentielle réparation par le mécanisme NHEJ. Cependant, les duplications induisant la maladie PMD étaient interrompues par des fragments tripliquées. Cette observation n'était pas cohérente avec le mécanisme NHEJ et un nouveau modèle pour expliquer ce phénomène a été proposé. Ce modèle se base sur une réplication ralentie d'un des deux brins d'ADN lors de la réplication de l'ADN. Ce brin d'ADN

ralenti se désengagerait de sa fourche de réplication pour se transférer et se raccrocher à une autre fourche de réplication grâce à des microhomologies entre les brins. La synthèse d'ADN se poursuivrait sur cette nouvelle fourche, qui pourrait de nouveau subir un ou plusieurs "sauts" de fourche de réplication (Figure 1.5)[45].

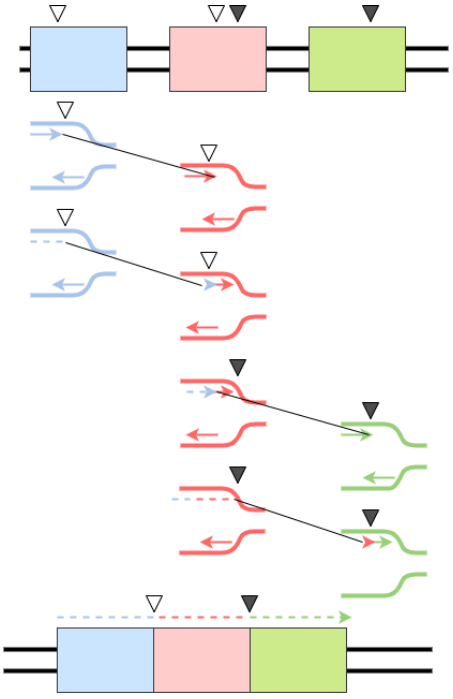

FIGURE 1.5 – Modèle *Fork Stalling and Template Switching* (FoSTeS). Figure adaptée de [45]. L'exemple présente un double FoSTeS conduisant à la délétion de deux fragments. Aucune grande homologie n'est nécessaire entre les séquences représentées par un rectangle bleu, rouge et vert. Toutefois, des microhomologies, représentées par des triangles, sont observées au niveau des jonctions des fragments.

1.1.4 Les éléments mobiles

Les éléments mobiles, composant près de 50% du génome humain, sont des structures génétiques capables de se répliquer et/ou de se déplacer d'une position du génome à une autre. Parmi ces éléments, les transposons sont très répandus au sein des eucaryotes et sont une des causes de l'expansion du génome[6]. Leur fonction se réduit à une duplication ou à une transposition de l'élément, ce qui représente un premier mécanisme de génération de variant de structure. L'insertion d'un élément mobile dans un gène peut, par exemple, conduire à un blocage de la fonction du gène[138]. Puisqu'ils composent plus de 50% du génome, les éléments mobiles représentent également des séquences homologues dispersées dans le génome qui peuvent être

impliquées dans une mauvaise réparation de cassures de l'ADN.

Deux types de transposons ont été décrits, les retrotransposons (classe 1) basés sur une stratégie copier/coller et les transposons à ADN basés sur une stratégie couper/coller[29]. Le premier va tout d'abord être exprimé en ARN grâce au mécanisme d'expression des gènes de l'organisme. Cet ARN va être inversement transcrit en ADN, ce qui va permettre à la copie d'ADN de s'intégrer dans le génome à une nouvelle position. Le second type d'éléments mobiles ne va pas subir d'expression d'ADN mais va être excisé, puis s'introduire à une nouvelle position dans le génome.

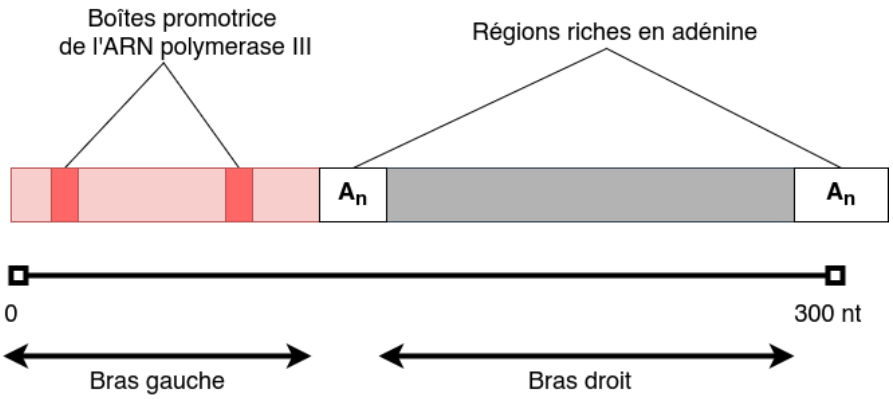

FIGURE 1.6 – Structure des éléments *Alu*. Figure adaptée de [48]. Les éléments *Alu* mesurent environ 300 paires de bases et sont composés de deux bras monomériques. Le bras gauche possède deux boîtes correspondant à des promoteurs internes de l'ARN polymerase III. Les deux bras sont séparés par une région riche en adénine. Le bras droit se termine par une petit queue poly(A).

Les retrotransposons peuvent être divisés en classes, selon leur structure ou leur origine. On distingue les LINE (Long Interspersed Nucleotide Elements), représentant 20% du génome humain et se caractérisant par une taille d'environ 7 kilobases ; les SINE (Short Interspersed Nucleotide Elements) qui possèdent une taille comprise entre 100 et 700 paires de bases. Ils possèdent une région interne de 50 à 500 paires de bases contenant un ARNt nécessaire à leur transposition. La famille *Alu* fait partie des SINE les plus représentés dans le génome humain, occupant 11% de ce dernier. Leur taille est estimée à 300 paires de bases et ont une structure particulière composée de deux bras, séparés par deux régions polyA (Figure 1.6)[48].

1.1.5 Impacts des variations génétiques

L'impact des variations génétiques peut être caractérisé selon la conséquence sur le phénotype de l'individu. Une variation génétique dont l'effet est dit neutre ou silencieux n'induit

pas de variation phénotypique et n'a pas de conséquence sur la vie de l'individu. Au contraire, une variation qui altère des fonctions biologiques de l'indivdu dans un environnement donné, réduisant sa capacité à survivre, est décrite comme pathogène. Ce type de variation est très étudié dans la compréhension des maladies d'origine génétique. Les variations peuvent cependant conférer un avantage sélectif permettant à l'individu une meilleure survie dans un environnement donné. Ce sont ces dernières qui tendent à être sélectionnées et disséminées au sein d'une population.

Comme nous l'avons vu précédemment, les mécanismes de réparation de l'ADN peuvent conduire à des modifications du génome d'un individu. Ces variations peuvent être localisées dans des séquence codantes de gènes et être la cause de maladie génétique[119, 115]. Bien que les indel et les SV ne représentent que 14% des variations génétiques par génome (Table 1.1), leur implication dans des maladies génétiques est plus importante. Une analyse des variations provenant de la base de données de mutations des gènes humains HGMD révèle que 34% des variations causant des maladies sont constituées de variants autres que des SNP (`www.hgmd.cf.ac.uk/`))[34]. Des variations peuvent être localisées dans des régions régulatrices de gènes conduisant à une expression différentielle de gènes. Des délétions dans les régions régulatrices du gène *SOX9* sont associées au syndrome de Pierre Robin[9]. Ces variations restent complexes à identifier puisqu'elles peuvent affecter des éléments de régulation *cis* qui ont un effet sur des gènes localisés à plusieurs centaines de kilobases de ces éléments[121].

Les éléments mobiles sont susceptibles d'être impliqués dans des maladies génétiques, notamment via l'insertion de ces éléments ou leur implication dans des réarrangements complexes. Des maladies induites par l'insertion d'éléments *Alu* dans des régions particulières ont déjà été observées. Par exemple, l'insertion d'un élément *Alu* dans le gène *ALSM1* conduit au syndrome d'Älstrom ou à de l'hypertension artérielle pulmonaire lorsque l'insertion a lieu dans le gène *BMPR2*[128, 59]. Les éléments mobiles sont également un important facteur impliqué dans l'évolution du génome puisqu'ils représentent plus de la moitié du génome humain. En effet, de part leur structure répétée dans le génome, ces éléments représentent d'importantes régions propices à la génération de variations génétiques via la réparation de l'ADN. Un enrichissment en *Alu* a notamment été trouvé aux abords de 2,366 duplications segmentaires chez l'homme. Il est ainsi supposé que les *AluY* et *AluS* seraient fortement impliqués dans la génération de duplications segmentaires conduisant à une expansion du génome humain[36, 6].

Il est important de noter que, en théorie, la localisation de la variation serait plus importante que le type de variation à un locus donné. Ainsi ce serait la disruption d'un gène par un variant quelconque plutôt que le type du variant lui-même qui serait la cause d'une maladie génétique.

Cette hypothèse a pu être vérifiée avec le syndrome de Potocki-Lupski (PLS) induit par une duplication au même locus qu'une délétion induisant le syndrome de Smith-Magenis (SMS)[109].

1.2 Capture de l'information génétique et assemblage du génome humain

1.2.1 Technologies de séquençages

Les variations génétiques sont ainsi des modifications du génome capables d'induire de grandes modifications chez un individu. Suite à la découverte de l'ADN en 1953 par Maurice Wilkins, Rosalind Franklin, James Watson et Francis Crick, d'importants efforts scientifiques et technologiques ont été réalisés afin de pouvoir capturer l'information que pouvait contenir cette molécule.

Première génération de séquençage

L'accès au matériel génétique a été possible grâce au séquençage d'ADN, dont deux méthodes ont été développées en 1977, par deux équipes indépendantes. La première dirigée par Frederick Sanger, se base sur un séquençage par synthèse de brin d'ADN complémentaire. L'objectif est similaire au mécanisme de réparation d'ADN d'une cassure simple brin. Une ADN polymerase est utilisée pour ajouter des nucléotides modifiés au brin à compléter. Ces nucléotides modifiés appelés didésoxyribonucléotides (ddNTP) sont quatre versions altérées des nucléotides d'adénine (A), guanine (G), cytosine (C) et thymidine (T). Leur particularité est que l'incorporation d'un nucléotide modifié bloque la synthèse du reste du brin complémentaire. Le séquençage complet d'une séquence nécessite quatre réactions chimiques en parallèle, chacune comportant en concentration faible un des quatre types de nucléotides modifiés et les quatre nucléotides non modifiés en concentration forte. Les fragments ainsi séquencés sont d'une taille variable et subissent par la suite une migration sur gel, où plus le fragment est grand, moins le fragment migrera dans le gel. Quatre lignes indépendantes sont présentes sur ce gel, chacune associée à une réaction. Les positions des différents nucléotides dans la séquence sont repérables par une bande, indiquant une incorporation de ddNTP, sur la ligne correspondante de la plaque de gel. Cette visualisation des nucléotides sous forme de bandes est réalisée à l'aide d'un système d'imagerie tel que les rayons X ou la lumière ultra-violette. Elle est rendue possible grâce à un traceur radioactif, préalablement incorporé dans l'ADN séquencé[113]. L'autre équipe dirigée par Allan Maxam et Walter Gilbert, base leur méthode sur le clivage chimique

de brin d'ADN. Chaque fragment à séquencer est préalablement marqué en 5' par radioactivité, puis une unique coupure d'un brin marqué est réalisé chimiquement. Quatre réactions de coupure sont réalisées, chacune coupant uniquement un type de nucléotide. A la fin de ce processus le brin d'ADN à séquencer, présent en de multiples copies dans l'échantillon, sera coupé à différents endroits du brin. L'ensemble de ses fragments sont ensuite déposés sur gel et une migration par éléctrophorèse a lieu. La révélation par marqueur radioactif ou fluorescence, de ses fragments migrés, permet la lecture de la séquence en nucléotides du brin séquencé[88].

Ces deux techniques de séquençage permettent ainsi de produire des lectures, ou *reads*, de longueur moyenne de 500 à 600 paires de bases, affichant un très faible taux d'erreurs, de l'ordre de 0,001%. Cependant, l'utilisation plus limitée d'éléments radioactifs de la technique de Sanger a conduit à sa démocratisation et est toujours utilisée de nos jours. Cette technologie a permis d'obtenir la première version de la référence du génome humain. Elle est encore utilisée pour valider de manière plus fiable la présence de variations génétiques observées par l'analyse de données issues des technologies de séquençage de seconde génération. Cette première génération de séquençage possède néanmoins des inconvénients : le besoin d'une forte concentration d'ADN, la préparation de *librairies* correspondant à des fragments d'ADN à séquencer stockés dans des chromosomes bactériens artificiels (BAC), ou encore le besoin de réaliser 4 séquençages en parallèle. L'ensemble de ces facteurs induisent un coût économique, humain et temporel important.

Seconde génération de séquençage

L'année 2005 a vu l'arrivée d'une nouvelle génération de séquençage surpassant la première génération en parallélisant massivement le séquençage, réduisant son coût et le temps du séquençage. Cette nouvelle génération permet de séquencer à moindre coût des millions de *reads* courts en parallèle. La lecture de la séquence a été optimisée, ne nécessitant plus de migration sur gel d'éléctrophorèse. Quatre technologies ont été mises au point entre 2005 et 2010 : Roche 454 (2005), Illumina (2006), SOLiD (2007) et Ion Torrent (2010). L'ensemble de ces technologies se base sur une première étape de préparation de l'ADN où celui-ci est fragmenté et où des adaptateurs sont ensuite fixés à chaque extremité des fragments d'ADN. A partir de cela, les protocoles divergent.

La méthode de séquençage Illumina est réalisée par ligation des fragments d'ADN sur une plaque portant des adaptateurs complémentaires à ceux présents sur les fragments d'ADN. Une première amplification par PCR a lieu, via une réaction par pont (*bridge*), où l'adaptateur libre du fragment va se lier avec un autre adaptateur fixé sur la plaque. Une synthèse de fragment

d'ADN a lieu à partir de ces adaptateurs, en utilisant le fragment d'ADN comme modèle par l'ADN polymérase. Ces étapes d'hybridations et d'amplifications continuent jusqu'à obtenir des milliers de fragments (Figure 1.7). Du fait que ces amplifications ont lieu de proche en proche, des clusters se forment, où chaque cluster contient des milliers de copies d'un fragment d'ADN.

Des amorces, des ADN polymérases ainsi que des nucléotides modifiés sont ensuite ajoutés à la plaque. Ces nucléotides ont deux particularités, ils émettent une couleur particulière lorsqu'ils sont utilisés par l'ADN polymérase et ils possèdent un terminateur réversible pour qu'un seul nucléotide soit intégré à la fois. Les amorces se lient aux fragments d'ADN, qui sont étendus par incorporation de nucléotides, à l'aide de l'ADN polymérase. Lorsque le nucléotide est intégré, une lecture de la longueur d'onde est réalisée afin d'identifier le nucléotide ajouté. La plaque est ensuite lavée afin de retirer les nucléotides non incorporés, ainsi que les terminateurs des nucléotides ayant été incorporés. Le processus peut ainsi se poursuivre, avec une nouvelle incorporation de nucléotides, jusqu'à ce que le fragment d'ADN soit totalement séquencé (Figure 1.8). Les *reads* séquencés via cette technologie possèdent une longueur moyenne de 100 à 150 paires de bases et un taux d'erreur inférieur à 0.1% . A l'image de la technologie développée par Sanger, la technologie Illumina est celle qui s'est le plus démocratisée au sein des technologies de seconde génération.

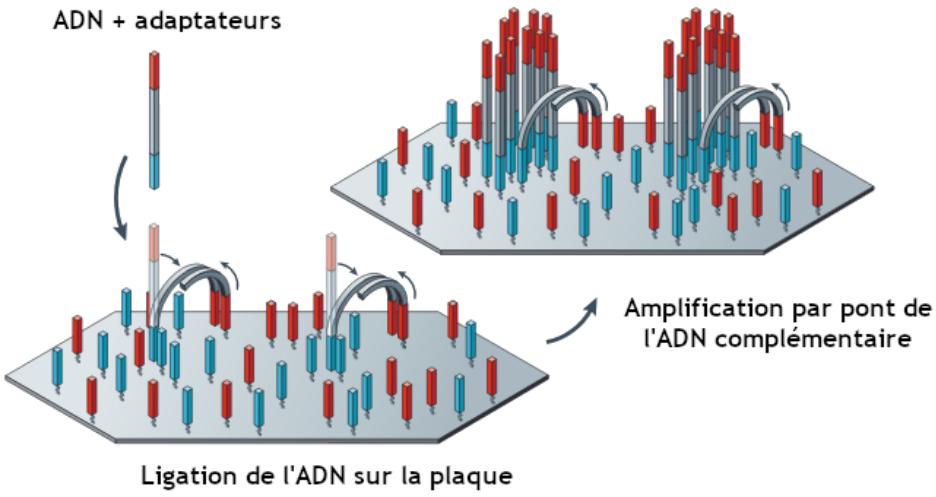

FIGURE 1.7 – Immobilisation et amplification de l'ADN en amont de l'étape de séquençage *Illumina*. Figure adaptée de [91]

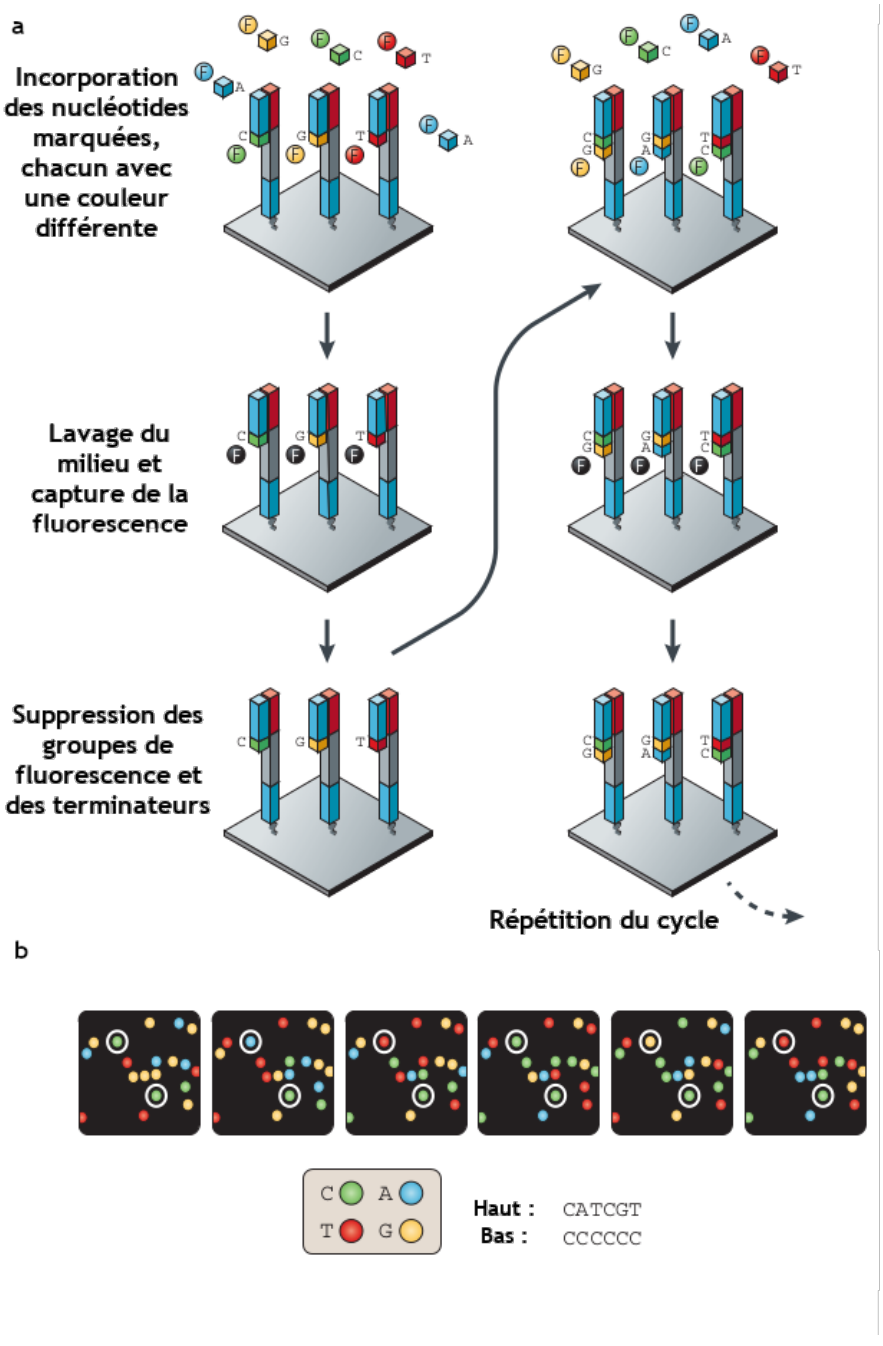

FIGURE 1.8 – Technologies de séquençage *Illumina* : séquençage de l'ADN. Figure adaptée de [91]. (a) Des nucléotides sont marqués par fluorescence et un terminateur y a été attaché pour qu'un seul nucléotide puisse se lier sur chaque fragment. Les nucléotides marqués sont déposés dans le milieu afin de pouvoir se lier avec les nucléotides complémentaires des fragments d'ADN à séquencer. Le milieu est ensuite lavé afin de pouvoir capturer les nucléotides marqués liés aux fragments d'ADN. Les groupes permettant la fluorescence et empêchant l'incorporation d'autres nucléotides sont retirés. Les différentes étapes sont reproduites jusqu'à ce que les fragments d'ADN soient entièrement séquencés. (b) Exemple de capture des nucléotides liés aux fragments d'ADN.

Une amélioration des séquençeurs Illumina a été développée permettant de séquencer ensemble les deux extrémités des fragments d'ADN, appelée séquençage *paired-end* (Figure 1.9). Une première étape de fragmentation de l'ADN est réalisée afin d'obtenir des fragments double brins d'une taille comprise entre 200 à 500 paires de bases. Des adaptateurs sont liés aux fragments afin de permettre leur séquençage. Cette technologie à l'avantage d'apporter des informations supplémentaires par rapport au séquençage *single-end*. La distance entre les *reads* pairés d'un même fragment, appelée taille d'insert, est connue et les deux *reads* sont orientés dans des sens opposés.

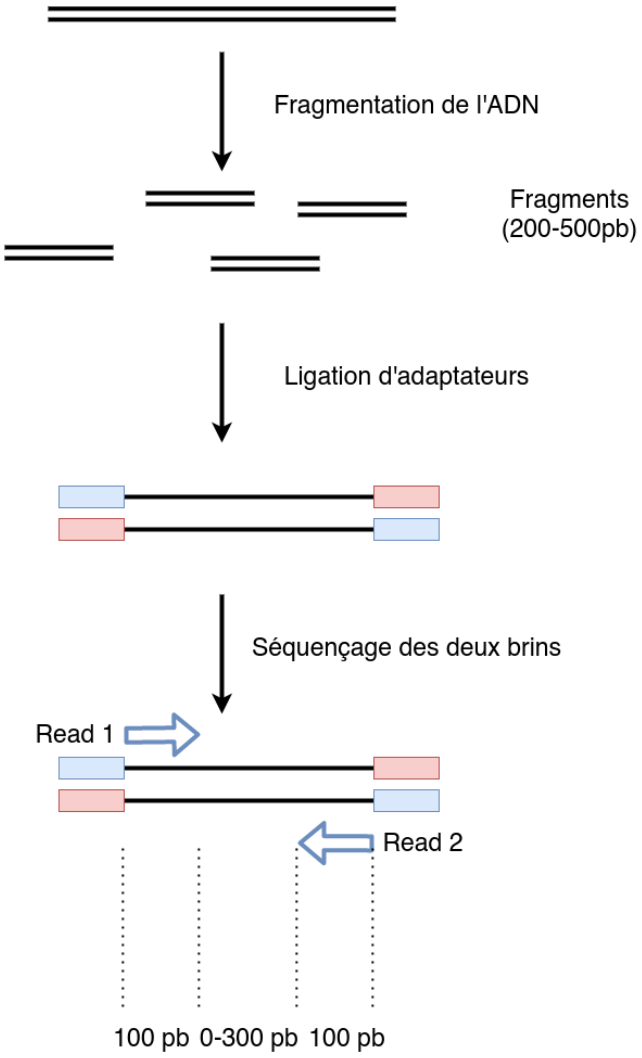

FIGURE 1.9 – Technologies de séquençage *reads courts paired-end*. Figure adaptée de [91]. Cette technologie permet le séquençage des deux extrémités du fragment double brin.

Cette seconde génération de séquençage a ainsi permi la démocratisation du séquençage

conduisant à la génération de nombreux génomes de référence pour de multiples espèces. Les principes de la seconde génération de séquençage ont été étendus pour être appliqués à d'autres domaines tels que l'étude du transcriptome via le séquençage d'ARN, de l'épigénétique, des génomes de population ou encore des génomes de l'ensemble des organismes contenus dans un environnement, appelé métagénomique.

Cette génération de séquençage possède cependant deux limites majeures : elle peut nécessiter une étape d'amplification des fragments d'ADN avant qu'ils ne soient séquencés et la longueur des *reads* est courte (<150 paires de bases). L'amplification des fragments d'ADN par PCR est une étape pouvant induire une hétérogénéité de couverture de séquençage le long du génome. Des études ont mis en évidence qu'un biais de séquençage existait dans des régions pauvres ou enrichies en GC avec la technologie Illumina. L'étape d'amplification des fragments d'ADN serait l'une des responsables majeurs de ce biais[1]. La taille des *reads* quant à elle rend complexe l'analyse, l'assemblage et l'alignement des *reads* contre un génome de référence localisés dans des régions répétées et dont la longueur est très souvent supérieure à la taille des *reads*.

Troisième génération de séquençage

En 2011, une nouvelle génération de séquençage a été développée et avait pour but, à l'image de la seconde génération, de surpasser les limites de la génération précédente. Cette génération vise donc à produire de très grands fragments d'ADN, d'une taille médiane atteignant 10 kilobases mais pouvant atteindre plusieurs mégabases. Ces fragments sont obtenus directement depuis l'ADN extrait et ne nécessitent pas une amplification systématique de l'ADN contrairement aux technologies de seconde génération. Deux technologies ont émergé avec deux approches distinctes : Pacific Biosciences (PacBio) et Oxford Nanopores Technologies (ONT).

La technologie de séquençage PacBio repose sur une approche de séquençage d'une seule molécule d'ADN en temps réel (*Single Molecule Real-Time*, SMRT) (Figure 1.10 a). Le processus repose sur un séquençage par synthèse similaire à l'approche utilisée par Illumina, mais plutôt que d'exécuter des cycles d'amplifications, le signal émis lors de l'incorporation d'un nucléotide est détecté en temps réel. PacBio utilise une matrice moléculaire d'ADN topologiquement circulaire, appelée SMRT-bell, qui est composée d'un fragment d'ADN double brins avec des adaptateurs d'épingle à cheveux simple brin à chaque extrémité. Une fois que la SMRTbell est assemblée, elle est liée par une ADN polymérase et chargée sur une cellule SMRT pour être séquencée. Au cours de la réaction de séquençage, la polymérase se développe autour de la matrice de la SMRT-bell et incorpore des désoxynucléotides triphosphates marqués par fluo-

rescence dans le brin d'ADN. Après chaque incorporation, un laser excite le fluorophore et une caméra enregistre l'émission. Le fluorophore est ensuite séparé du nucléotide avant que le désoxynucléotide triphosphate suivant ne soit incorporé. Ce processus est répété des milliers de fois pour révéler l'identité et la séquence de chaque base dans le modèle de la SMRT-bell.

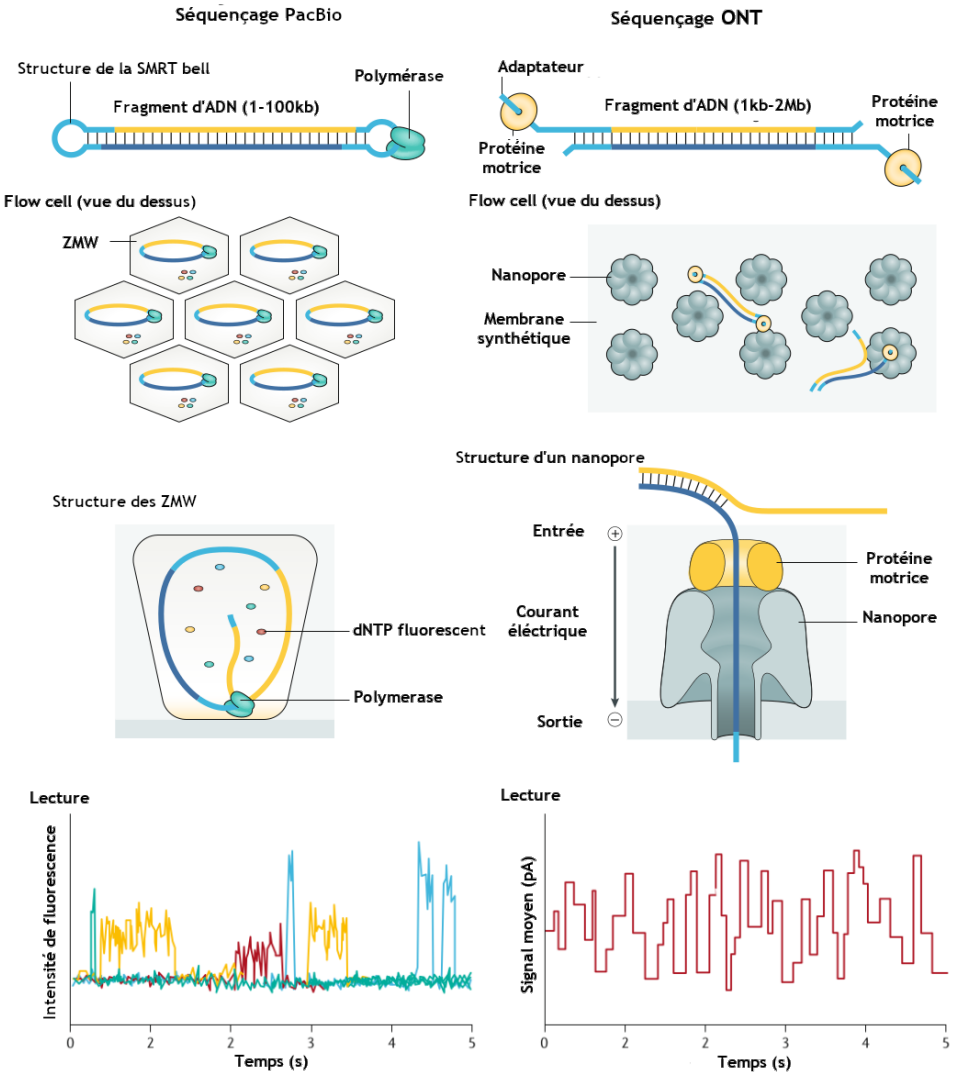

FIGURE 1.10 – Technologies de séquençage long *reads* Pacific Bioscience (PacBio) à gauche et Oxford Nanopore Technologies (ONT) à droite. Figure adaptée de [83]. La technologie PacBio est caractérisée par l'utilisation d'une *SMRT-bell* et d'une lecture par fluorescence des fragments séquencés. La technologie ONT est caractérisé par l'utilisation de nanopore et d'une lecture de l'ADN séquencé par la capture des fluctuations du courant induites par le passage des nucléotides à travers les pores.

Contrairement à la technologie PacBio, ONT utilise des molécules d'ADN linéaires plutôt que circulaires et base son séquençage sur l'utilisation de nanopores (Figure 1.10 b). Le séquençage ONT commence par l'ajout d'un adaptateur de séquençage, possédant une protéine motrice, aux molécules d'ADN double brins à séquencer. Le mélange d'ADN+adaptateur est chargé sur une cellule à flux ou flowcell, qui contient des centaines à des milliers de nanopores incorporés dans une membrane synthétique. La protéine motrice présente au niveau de l'adaptateur va dérouler l'ADN double brins qui, avec un courant électrique, va entraîner l'ADN chargé négativement à travers le pore à une vitesse contrôlée. Lorsque l'ADN se déplace à travers le pore, il provoque des réactions d'hypersensibilité. Les perturbations du courant sont différentes selon chaque nucléotide, ce qui permet d'analyser en temps réel la séquence d'ADN.

Technologie de séquençage	Plateforme	Type de données	Taille des reads	Précision (%)	Prix par Gb ($)
Pacific Biosciences	RS II	CLR	5-15 kb	87-92	333-933
	Sequel	CLR	25-50 kb		98-195
	Sequel II	CLR	30-60 kb		13-26
	Sequel II	HiFi	10-20 kb	>99	43-86
Oxford Nanopore Technologies	MinION/GridION	Long	10-60 kb	87-98	50-500
	MinION/GridION	Ultra Long	100-200 kb		500-2000
	PromethION	Long	10-60 kb		21-42
Illumina	NextSeq 550	Single-end	75-150 pb	>99.9	50-63
		Paired-end	75-150 pb (x2)		40-60
	NovaSeq 6000	Single-end	50-250 pb		10-35
		Paired-end	50-250 pb (x2)		

TABLE 1.2 – Comparaison des technologies de seconde et de troisième génération. Table adaptée de [83]. Les coûts affichés ne prennent pas en compte le coût des machines, de la main d'oeuvre, de la maintenance et des ressources informatiques nécessaire pour l'utilisation de ces technologies. CLR : *longs reads* continus (*continuous long reads*), HiFi : *reads* de haute fidélité (*High Fidelity*).

La première limite notable des technologies de séquençage de troisième génération concerne la qualité des fragments séquençés. Alors que les technologies de seconde génération affichent un taux d'erreur en dessous de 0.1%, les premiers protocoles de cette troisième génération présentaient des taux d'erreur supérieurs à 20-30%. La nature des erreurs est également différente. Alors que la seconde génération induit des erreurs de substitution, la troisième génération de séquençage induit des erreurs d'insertions/délétions.

Il est néanmoins important de noter que le type d'erreur de séquençage diffère entre les deux technologies de troisième génération. La technologie de séquençage de PacBio induit plus d'erreurs de type insertions que de délétions et qui sont distribuées aléatoirement le long du *read*. Inversement, la technologie de séquençage ONT induit plus d'erreurs de type délétions que d'insertions. Elle réalise aussi des erreurs de séquençage systématiques dans les homopolymères. L'amélioration ces dernières années, de la chimie de ces protocoles, a permis de réduire de manière significative le coût de séquençage, se rapprochant peu à peu du coût de séquençage de seconde génération (Table 1.2)[83]. Très récémment, de nouveaux protocoles ont vu le jour permettant de réduire le taux d'erreur à moins de 5-6% pour la technologie ONT et à moins de 1% grâce au protocole HiFi de PacBio. Cette baisse du taux d'erreur de PacBio est notamment obtenue grâce à une auto-correction des fragments séquencés par alignement des copies séquencées d'un même fragment. Le prix de ces séquenceurs et de ces nouveaux protocoles restent difficilement abordables en vue d'une utilisation en routine dans un usage médical.

1.2.2 Assemblage du génome humain de référence

Assemblage du premier génome humain

De part la taille des *reads* n'excédant pas quelques centaines de paires de bases, le séquençage de première génération ne permettait d'accéder qu'à des fragments du génome humain (3.2Gb). Cette contrainte nécessita donc une étape d'assemblage des *reads* afin d'obtenir un génome complet reconstitué. Ceci n'a pas empêché la communauté scientifique de relever le défi en 1993 dans l'objectif de produire une référence complète du génome humain. C'est ainsi que deux projets ont été développés : le *Human Genome Project* et la compagnie privée de Craig Venter, Celera Genomics.

Le *Human Genome Project* était un projet mobilisant la communauté scientifique internationale dont le but visait à produire une première version complète du génome humain. Ce projet a été officiellement lancé en 1990 et s'est achevé 13 ans plus tard, le 14 avril 2003, bien que les premières analyses ont été publiées en 2001[52, 132, 68]. La tâche était divisée par chromosome, où chaque équipe de recherche était en charge de séquencer et d'assembler un ou plusieurs chromosomes particuliers. Il aura nécessité plus de 3 milliards de dollars pour produire une première version du génome. Du fait de la technologie, seule les régions euchromatiques ont été séquencées, représentant 92.1% du génome humain. La publication du génome de référence a permi de créer une base commune pour la communauté scientifique. Ce génome permet une uniformisation et standardisation des études du génome humain, où chaque étude se compare

à une même référence. Les informations et les découvertes de ces études peuvent donc être réutilisées et comparées sans avoir à prendre en compte l'utilisation de génomes différents.

Amélioration du génome de référence

La première version du génome humain ne comportait que les régions euchromatiques, laissant de côté les régions centromériques et télomériques. Les objectifs suivants visaient à corriger les erreurs de séquençage potentielles, de résoudre les centaines de régions dont les séquences étaient non résolues dans l'euchromatine, aussi appelées *gaps* et de proposer des génomes alternatifs contenant de la variabilité génétique. Il a fallu 6 ans suite à la publication du premier draft, soit en 2009, pour voir une mise à jour publiée du génome de référence. Ce génome s'est vu ajouté des chromosomes avec des assemblages alternatifs, la correction de 150 régions contenant des problèmes d'alignements, ainsi que la résolution de 25 *gaps*[31]. La mise à jour du génome de référence s'est poursuivie avec l'utilisation des nouvelles techniques de séquençage afin de résoudre les *gaps* restants. C'est ainsi que l'utilisation des *longs reads* PacBio appliquée dans un contexte de séquençage de cellule unique (*single cell*) a permis la résolution de 50 des 164 *gaps* restant de la version GRCh37 et de la réduction de la taille de 40 autres *gaps*[18]. La version la plus récente est GRCh38 publiée en 2013, bien que la version 37 soit toujours utilisée dans plusieurs études et dans la majeure partie des laboratoires de diagnostics. Cette dernière version a également profité de l'arrivée des technologies *longs reads* pour améliorer la résolution du génome[53]. A l'image du *Human Genome Project*, le consortium *Telomere-to-Telomere* (T2T) a pour objectif d'assembler complètement le génome humain. En 2020 et pour la première fois, un chromosome humain a été entièrement assemblé, télomères et centromère inclus. L'assemblage de ce chromosome X a notamment été possible grâce à l'utilisation de la technologie *ultra long reads* de nanopore combinée à d'autres technologies pour valider l'assemblage[93]. Cette réussite démontre qu'il est maintenant envisageable d'obtenir une version complète du génome humain.

Il est important de noter que deux versions du génome humain peuvent être trouvées, une proposée par *Ensembl* et la seconde par l'Université Californienne de Santa Cruz (UCSC). Bien que la séquence génomique soit la même, les annotations des chromosomes, des gènes et autres régions d'intérêt diffèrent. Les génomes humains de l'UCSC sont identifiables à travers la dénomination *hgX*, X étant la version du génome, tandis que l'annotation *Ensembl* utilise l'appelation *GRchX*. Depuis la production de la version GRCh38, *The Genome Reference Consortium* propose et travaille sur une unification de ces deux versions proposées par *Ensembl* et l'UCSC.

Limites du génome de référence

La première limite du génome de référence concerne la représentation simplifiée sous forme d'un génome haploïde. Les analyses actuelles sont principalement basées sur l'alignement de *reads* sur un génome de référence. Ces analyses peuvent être biaisées par cette linéarité : des alignements peuvent manquer ou des *reads*, provenant des allèles non-référencés, peuvent s'aligner de manière incorrecte sur la référence. Cela peut *in fine* conduire à une mauvaise interprétation des résultats scientifiques, en particulier pour les analyses portant sur des régions hypervariables[30, 112]. Par exemple, les génomes provenant du continent africain contiennent plus d'allèles alternatifs, et peuvent donc être plus gravement affectés par un biais de référence[46]. La seconde limite est induite par l'état de résolution du génome. Comme nous l'avons vu, certaines régions ne sont toujours par résolues, comme les régions centromériques et télomériques, qui les rendent difficiles à étudier. Il est également important de noter que le génome n'est qu'un support d'informations n'expliquant qu'en partie le fonctionnement de l'organisme[26].

Annotation du génome de référence

Le séquençage et l'assemblage d'un génome de référence a permis de donner accès à de l'information génétique à grande échelle. Cela a permis de répondre à de nombreuses questions scientifiques telles que la structure du génome, sa composition en gènes et en répétitions. Le projet international ENCODE lancé en 2003, avait pour objectif d'identifier l'ensemble des éléments fonctionnels du génome humain. Le processus d'annotation impliqua des annotations automatiques par des outils de prédiction, des curations automatiques et manuelles, ainsi que des vérifications par approches expérimentales. D'autres projets et études sont venus compléter l'annotation des gènes, tels que *RefSeq* et *AceView*[104, 129]. Les gènes peuvent être détectés en utilisant les ORF (*Open Reading Frame*) comme marqueurs de la présence de gènes. Ces ORF correspondent à des séquences bornées par des codons. Ces séquences commencent par un codon *start* (ATG) et se terminent par un codon *stop* (TAA, TAG, TGA). Du fait de la lecture par codon, trois cadres de lecture par brin d'ADN est possible. L'étape d'identification de ces codons à la main est cependante contraignante et le développement d'outils de prédiction, dits *ab initio*, a permis de faciliter cette détection. L'alignement de régions potentiellement codantes avec des cDNA, des EST (*expressed sequence tag*) ou de gènes d'autres espèces via *BlastX* ont permis d'appuyer l'existence de gènes dans le génome. Les gènes, composés d'exons et d'introns, sont principalement confirmés via le séquençage d'ARN. En effet, même si des marqueurs existent suggérant la présence de sites exoniques et introniques, le séquençage d'ARN reste la méthode

permettant de prouver l'existence d'un gène.

Les ARN non codants ainsi que les régions régulatrices peuvent être annotés via l'utilisation d'outils dédiés tels que *tRNA-ScanSE* ou l'identification par similarité de séquences avec des bases de données dédiées. Des stratégies de comparaison et de similarité de séquences entre espèces sont également utilisées afin de prédire des régions régulatrices potentielles n'ayant pas été identifiées expérimentalement[124]. Les éléments répétés ont principalement été identifiés grâce à RepeatMasker (`http://www.repeatmasker.org/`), outil qui combine plusieurs outils de détection de répétitions. Parmi ces outils, *Tandem Repeat Finder* (TRF) est utilisé pour détecter la présence de séquences (graines) répétées en tandem, appelées microsatellites, minisatellites ou encore répétitions simples[10]. *Dfam* est un outil qui permet de détecter la présence d'éléments mobiles[51]. Pour cela, l'outil s'appuie sur une base de données contenant la séquence des éléments mobiles déjà connus. Grâce à ces différents outils, RepeatMasker met en évidence que les éléments répétés occupent plus de 50% du génome de référence.

L'ensemble de ces annotations génèrent une quantité d'informations importante difficilement exploitables à l'oeil nu. Des *Genome Browser* ont été développés pour faciliter cette exploitation de données. Ces applications permettent de rechercher et visualiser rapidement l'ensemble des annotations contenues dans une base de données pour un locus donné[58].

1.3 Méthodologies d'analyses de données de séquençage

La réduction du coût de séquençage a ouvert l'étude des variations génétiques à l'échelle de génomes individuels et de populations entières. L'alignement de *reads* sur le génome de référence permet de détecter des alignements incohérents. Ces incohérences traduisent d'une variabilité génétique entre l'individu séquencé et le génome de référence. L'analyse de variants est généralement composée de trois étapes : un alignement, une détection de variants (*variant calling*) et une annotation des variants détectés (Figure 1.11).

1.3.1 Alignement de séquences

Le problème de l'alignement de deux séquences peut se présenter comme l'identification du nombre minimal d'opérations d'édition (subsitution, insertion, délétion) permettant de transformer un mot en un autre. Ce problème algorithmique est au coeur de la bioinformatique et permet la comparaison de séquences nucléiques ou protéiques dont les domaines d'utilisation sont vastes. Par exemple, il rend possible la comparaison de séquences entre espèces dans le but

d'inférer l'histoire phylogénique des espèces ou encore l'identification de variations génétiques.

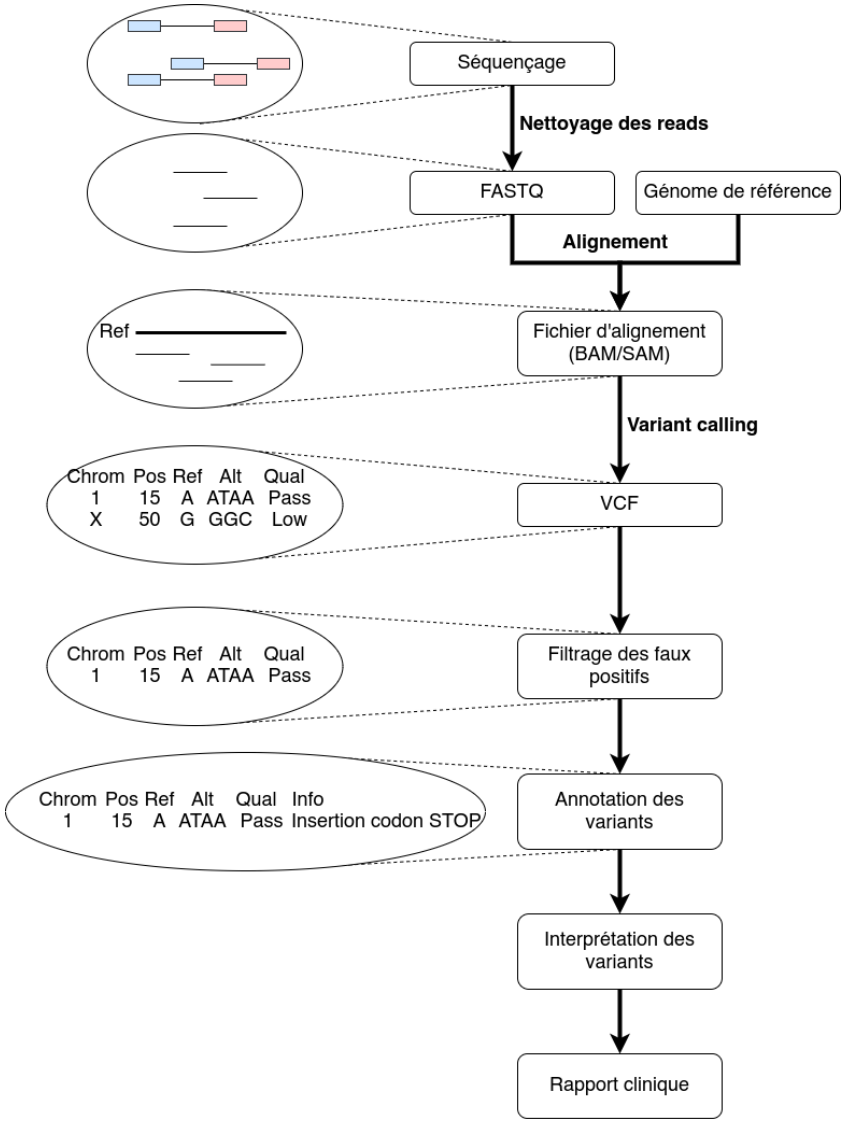

FIGURE 1.11 – Pipeline de détection de variations génétiques dans un contexte de diagnostic médical. Suite au séquençage, les *reads* sont nettoyés de leurs adaptateurs. L'alignement des *reads* sur un génome de référence sont rapportés dans des fichiers BAM/SAM. Des outils de détection de variants sont utilisés en se basant sur les informations fournies dans le fichier BAM. Les variants détectés sont rapportés dans des fichiers au format vcf qui sont filtrés afin de retirer les faux positifs. Les variants conservés sont annotés afin de permettre une interprétation de la part du diagnosticien et de réaliser un rapport clinique.

Développé par Needleman et Wunsch en 1970, l'algorithme de programmation dynamique portant le même nom résoud le problème de l'alignement de séquences nucléotidiques et pro-

téiques de manière exacte[99]. Cet alignement qualifié de global permet d'identifier la similarité de deux séquences sur l'intégralité de leur longueur. Smith et Waterman ont proposé en 1981 un second algorithme d'alignement de séquences, en programmation dynamique également, mais qui répond à une autre problèmatique qui est celle de l'alignement local[118]. Cet alignement permet de mettre en avant des régions similaires au sein de deux séquences. Ces deux types d'alignements sont réalisés grâce à une matrice à deux dimensions, où chaque dimension représente une des deux séquences, et qui stocke les scores d'alignements de toutes les paires de préfixes des deux séquences. Une relation de récurrence permet de remplir une case donnée de la matrice en fonction de certaines de ses voisines. Le résultat est obtenu en un temps quadratique O(M,N), où M et N représentent la longueur des séquences.

La complexité quadratique de ces algorithmes les rend pratiquement inutilisables dans le cas de recherche d'alignement de séquence ou de *reads* sur un génome. En effet, un séquençage produit plusieurs millions de *reads* et chaque *read* doit être aligné sur tout le génome. Pour résoudre cette complexité, des approches heuristiques basées sur une indexation des données et une réduction de l'espace de recherche ont été proposées.

L'indexation de données permet de compresser et d'accéder rapidement à la séquence sur laquelle les *reads* sont alignés. Une des approches est le stockage du génome de référence sous forme de k-mers, correspondant à l'ensemble des mots de taille k contenu dans le génome. La stratégie réduisant l'espace de recherche, appelé *seed and extend*, identifie l'ensemble des k-mers partagés, appelés graines (*seeds*), entre les *reads*, appelés ici *query*, et le génome de référence, appelé ici *target*. Ces graines sont ensuite utilisées pour étendre (*extend*) l'alignement, généralement via des méthodes d'alignements reposant sur de la programmation dynamique. Cette stratégie est implémentée dans l'outil d'alignement BLAST[4]. Si elle a beaucoup été utilisée pour la recherche de similarité dans des bases de données, elle reste peu utilisée pour l'alignement de millions de *reads* sur un génome.

Une seconde approche basée sur une *Burrows-Wheeler Transform* (BWT) et un FM-index permet un alignement rapide des *reads* sur une génome de référence, tout en limitant l'espace mémoire requis pour l'indexation du génome de référence[13, 39]. La BWT est une transformation du génome de référence qui permet sa compression. Associée au *FM-index*, elle permet le requêtage de mots afin de trouver le nombre d'occurrences et les positions d'un mot dans un texte. Cette stratégie s'est démocratisée et est maintenant utilisée dans la plupart des outils d'alignements de *reads* tels que BWA ou encore Bowtie2[76, 69]. Les alignements obtenus par ces outils sont principalement rapportés dans deux formats standardisés. Le format SAM (*Sequence Alignment Map*) représente les alignements sous forme de texte, alors que le format

BAM (*Binary Aligment Map*), les représente sous forme binaire[78]. Le format SAM est composé de onze champs obligatoires décrits Table 1.3. Les informations rapportées concernent les identifiants des séquences, la position d'alignement sur le génome de référence ou encore la longueur de l'alignement.

Colonne	**Nom**	**Description**	**Type**
1	QNAME	*Header* du *read*	Chaîne de caractères
2	FLAG	Drapeau décrivant l'alignement	Entier
3	RNAME	*Header* de la référence	Chaîne de caractères
4	POS	Position de début de l'alignement sur la référence	Entier
5	MAPQ	Qualité de l'alignement	Entier
6	CIGAR	Code *CIGAR*	Chaîne de caractères
7	MRNM	*Header* du second *reads* pairés	Chaîne de caractères
8	MPOS	Position d'alignement du second *reads* pairés	Entier
9	ISIZE	Longueur inferrée de la distance entre les *reads* pairés	Entier
10	SEQ	Facteur du *read* aligné	Chaîne de caractères
11	QUAL	Score de qualité Phred	Chaîne de caractères

TABLE 1.3 – Description des champs d'informations du format SAM. Table adaptée de [78].

L'alignement de *reads* sur un génome humain rencontre des difficultés dans des régions répétées. De part la taille des *reads* inférieure à 200 paires de bases, la confiance d'un résultat d'alignement dans de telles régions est plus faible que dans des régions non répétées. Le problème ne se situe pas dans la difficulté à aligner la séquence mais dans la fiabilité d'avoir aligné le *read* sur la répétition dont il provient biologiquement. Un *read* peut être aligné à de multiples localisations avec le même score de similarité. Une étude du génome de référence permet de réaliser des cartes dites de faible mappabilité. Ces cartes référencient les régions où les *reads* d'une certaine taille peuvent être alignés à de multiples localisations avec le même meilleur score d'alignement[57]. Ces cartes peuvent être utilisées pour ignorer les alignements dans ces régions lors des analyses en aval.

1.3.2 Méthodes de détection des variations de structure

Comme nous l'avons vu précédemment, l'alignement est un outil puissant permettant d'identifier des régions communes entre deux séquences. La détection des variations de structure repose sur la recherche d'incohérences dans les alignements causées par de tels variants. Contrairement aux variants de structure, les SNP et les indel sont plus facilement détectables car ces événements sont entièrement contenus au sein d'un *read*. De ce fait, l'événement est borné par des séquences qui sont correctement alignées sur le génome de référence.

Identification des points de cassures

Les points de cassures peuvent être définis comme des adjacences de séquence différentes entre le génome d'un individu et celui de référence. Ces points de cassures sont des régions contenant potentiellement des variations génétiques. Ils peuvent être identifiés grâce à trois caractéristiques d'alignement de *reads* (Figure 1.12).

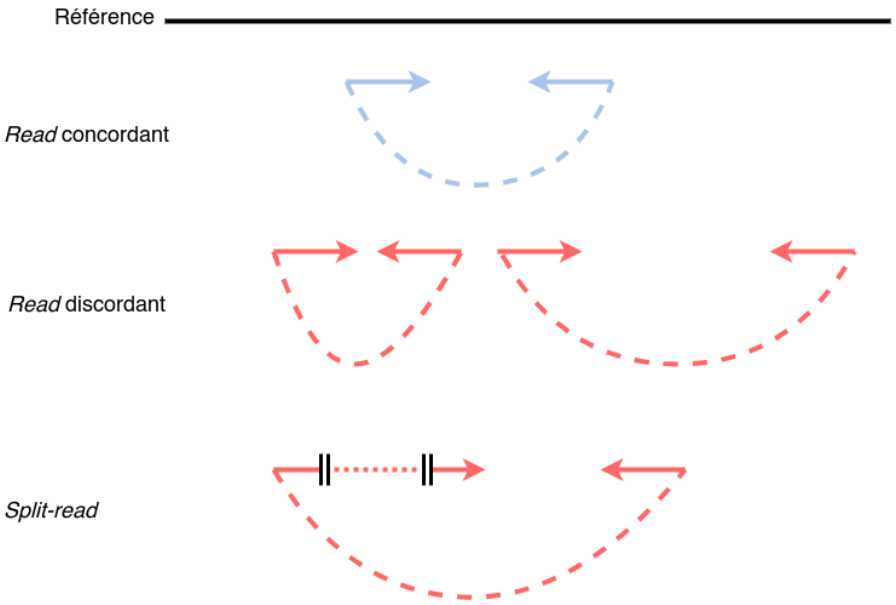

FIGURE 1.12 – Exemple des informations obtenues par l'alignement des *reads paired-end*. Figure adaptée de [24]. Les *reads* concordants représentent des paires dont la distance entre les deux alignements respecte la distance attendue, connue à partir de la taille des fragments séquencés. Les *reads* discordants représentent des paires dont la distance ou l'orientation entre les deux alignements ou leurs orientations est différente de celles attendues. Les *split-reads* représentent des *reads* dont l'alignement a conduit à une ou plusieurs coupures des *reads*.

La présence de trous (*gaps*) dans l'alignement est caractéristique d'une insertion ou d'une

délétion de petite taille. L'alignement de différentes portions d'un *read* à plusieurs endroits, appelé *clipped* ou *split reads*, est caractéristique d'événements de plus grandes tailles. Enfin l'utilisation de la technologie *paired-end* des *reads courts*, permet d'identifier des événements potentiels lorsque la distance ou l'orientation entre deux *reads* d'une même paire est différente de celle attendue (Figure 1.12).

Pour les variants de type délétions, les informations utilisées peuvent être l'absence de couverture de *read*, la présence de *split reads* ou d'une distance entre les deux *reads* d'une paire supérieure à celle attendue (Figure 1.13). Les inversions induisent un motif très particulier où deux points de cassures sont observés au début et à la fin de l'inversion. Un *read* d'une paire est également mal orienté dans le cas d'une inversion (Figure 1.13). La couverture de séquençage n'est pas un élément informatif dans ce cas, puisqu'il n'y a pas eu insertion ou délétion de matériel génétique. La transposition induit deux motifs particuliers qui sont une délétion et une insertion du segment transposé à des positions différentes. Cet événement, comme à l'image de l'inversion, n'induit pas de perte ou de gain de matériel génétique. Les insertions sont l'un des types les plus difficiles à caractériser (Figure 1.13). Les insertions sont différentes selon la nature de la séquence insérée, comme nous l'avons vu précédemment. Cette hétérogénéité de nature induit des alignements différents pour chaque type d'insertion. Par exemple, les *reads* associés à une insertion *de novo* ne s'alignent pas sur le génome de référence, tandis que les *reads* associés à une duplication vont induire une hausse de couverture au niveau de la séquence dupliquée.

Les régions répétées rendent difficiles la localisation précise des points de cassure car les *reads* associés à ces régions peuvent s'aligner sur l'ensemble des régions y compris la région qui contient le variant. De plus, une variation située dans une région dupliquée peut conduire à sa détection de multiple fois au sein des autre copies. Avec un génome humain qui contient plus de 50% d'éléments répétés et les variants qui ont tendance à être associés à ces éléments, la détection de points de cassure se révèle limitée avec des *reads courts*. La technologie de séquençage de troisième génération permet de résoudre cette limitation. La taille des *reads* produits par cette technologie permet à ces *reads* de contenir le variant en entier. Ils contiennent également plus d'informations sur le contexte génomique que les *reads courts*, ce qui permet de localiser les variants de manière non ambiguë (Figure 1.13).

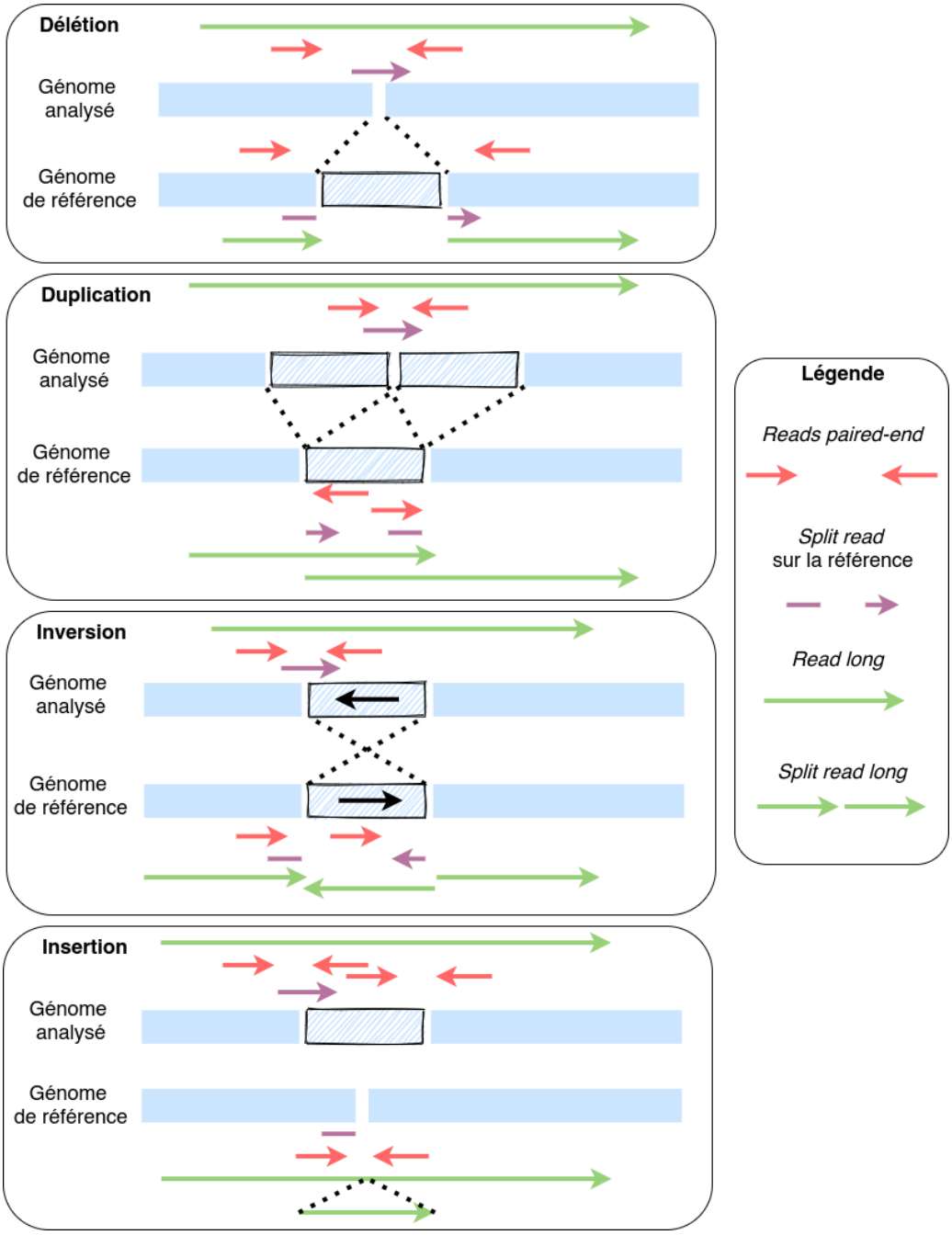

FIGURE 1.13 – Approches utilisées pour détecter des variants de structure à partir de l'alignement de *reads*. Figure adaptée de [85].

Résolution fine des variants de structure

Suite à la détection des points de cassure, l'objectif est de fournir la séquence du variant, d'identifier le génotype, ainsi que de calculer des métriques de qualité. Ces métriques sont utilisées pour mesurer la probabilité que le variant identifié soit un vrai variant et non pas un faux positif. Dans cette thèse nous définissons le terme résolution de séquence comme la capacité à assembler la séquence du variant.

La caractérisation des SNP et des indel est faite en analysant les *mismatch* et les *gaps* dans les alignements. De ce fait, il est plus simple de récupérer la séquence alternative, présente nativement dans les *reads* et dans le génome de référence. Pour les variants de structures, dont la taille peut être supérieure à la taille d'un *read*, la méthodologie varie selon les outils de détection et le type de variant. Contrairement à la délétion, dont la séquence supprimée peut être identifiée dans le génome de référence, la résolution de la séquence des insertions nécessite une étape d'assemblage plus fastidieuse. La majorité des méthodes assemble l'insertion en partant de la séquence présente à gauche du point de cassure pour arriver à retrouver la séquence présente à droite du point de cassure, en utilisant l'ensemble ou un échantillon de *reads*. La principale limitation de l'assemblage se situe dans la capacité à recruter efficacement les *reads* qui sont associés à l'insertion. De plus, le contenu et la taille de l'insertion ne sont pas connus à l'avance et ne sont pas identifiable avec l'alignement. Il est donc impossible de savoir si l'ensemble des *reads* associés à l'insertion ont bien été recrutés pour l'assemblage.

La technologie des *longs reads* a révolutionné la détection des variants de structure en surpassant les limites de la seconde génération de séquençage. De part une taille de *reads* de plusieurs kilobases, il est possible de s'émanciper de l'étape d'assemblage car le variant est contenu dans le *read*(Figure 1.13). Les limitations observées concernant la taille des *reads* de seconde génération restent valables pour cette troisième génération. Ainsi, la détection d'événements supérieurs à 5 kilobases se montre plus difficiles à détecter[85]. Enfin, le taux d'erreur dans les séquences limite la qualité et la précision de la séquence exacte du variant et la position reste approximtive.

Représentation des variations génétiques dans les bases de données

Les bases de données sont des structures capables de stocker des connaissances et avec le moins de redondance possible. Avec la démocratisation du séquençage, la quantité de données produites n'a cessé d'augmenter et le besoin de les stocker de manière efficace est devenu un champ d'étude à part entière. L'atout des bases de données, autre que le stockage, est

la possibilité de réaliser des requêtes dans le but de rechercher des données spécifiques. Par exemple, BLAST permet l'alignement de séquences avec des séquences contenues dans des bases de données, telles que RefSeq. Cette base de données collecte et stocke l'ensemble des données de séquences, incluant ADN, ARN et protéines, ainsi que leurs annotations fonctionnelles et strucurelles[101].

Des bases de données ont été développées pour stocker l'ensemble des variations génétiques. dbSNP est une base de données stockant les petits variants tels que les SNP, les indel ou encore les marqueurs de microsatellites (petites répétitions en tandem) de différentes espèces[116]. Une estimation en 2019 révèle que dbSNP contient plus de 675 millions de petits variants uniquement décrits chez *Homo sapiens*. A l'image de dbSNP, dbVar a été développée pour référencer les variants de structure de différentes espèces, mais depuis 2017, dbSNP et dbVar ne supportent plus que les données humains[70]. En 2020, ce sont plus de 6 millions de variants de structure qui sont stockés dans cette base de données provenant de 196 études. Une sous représentation des insertions au sein de dbVar est observable en comparaison au nombre de délétions rapportées. Seulement 28% des variants contenus dans dbVar correspondent à des insertions. Au sein de ces 28 % d'insertions, seulement 1.5% possèdent une séquence nucléique associée, principalement obtenue avec des technologies *longs reads*.

Les bases de données s'agrandissent grâce à la soumission de variants de la part de chercheurs ou de personnes associées à des laboratoires de recherches privés ou publics. Chaque variant soumis nécessite un ensemble d'informations afin d'attester de son existence. Ces informations concernent l'identification de la séquence ou du type de variant, de la méthode de détection, de l'étude associée, de la population concernée ou encore du génotype. De ce fait, les bases de données sont susceptibles de contenir des faux positifs, dont les dernières estimations rapportent 2.2% de faux positifs de SNP dans dbSNP. Bien que ce chiffre soit faible, ces faux positifs peuvent conduire à des conclusions erronées lors d'études de génétique humaine[5]. Les bases de données sont donc uniquement représentatives de ce que nous avons découvert et des biais peuvent exister.

1.3.3 La détection de variations génétiques pour un usage médical

Analyses standards

Identification des symptômes et du contexte génétique

La première étape dans le diagnostic d'une maladie d'origine potentiellement génétique est le regroupement de caractéristiques phénotypiques de la maladie. Des déficits dans le déve-

loppement physique ou mental peuvent être des signes suggérant une maladie génétique. La recherche de prédispositions à une maladie est réalisée par la recherche d'antécédents dans la généalogie du patient. Le contexte génétique tel que l'origine du patient, la présence de maladie génétique chez des ancêtres de la famille ou d'une union consanguine sont des facteurs qui peuvent guider le diagnostic. La seconde étape vise à s'appuyer sur la littérature à partir des caractéristiques observées chez le patient pour aiguiller vers des maladies potentielles. La base de donnée HPO, *Human Phenotype Ontology*, permet à travers un vocabulaire standardisé de phénotypes, d'associer des phénotypes à des maladies et à des variations génétiques. La troisième étape consiste en le séquençage du génome du patient dont les techniques varient du séquençage de gène unique au séquençage de génome complet. La décision de la technique à utiliser relève de l'expertise du clinicien qui se base sur l'ensemble des informations recueillies lors des étapes précédentes.

Séquençage de gène unique

Cette méthode est conseillée lorsque le résultat des premières étapes du diagnostic conduit, avec une grande confiance, à une maladie bien caractérisée. Le gène *FGFR3*, par exemple, est le seul connu pour être associé à l'achondroplasie. Le test monogénique du *FGFR3* permet de détecter des variations génétiques chez 99% des patients atteints d'achondroplasie et constitue donc l'approche la plus efficace en termes de coût et de temps[142]. La technique de séquençage Sanger est utilisée dans ce type d'approche.

Séquençage par groupe de gènes

Le séquençage par groupe de gènes est préféré dans le contexte de maladies associées à plusieurs variations génétiques. Cette technique permet de choisir un ensemble de régions d'intérêt pour être séquencées à moindre coût pour identifier avec une bonne probabilité une variation d'intérêt diagnostic. Plus la profondeur est grande, plus il est facile de retirer les erreurs de séquençage, les variants faux positifs, ou d'identifier le génotype. Cette réduction du matériel à séquencer permet une baisse du coût temporel et financier par rapport aux techniques de séquençage d'exomes ou de génomes entiers.

Cette technique permet de s'intéresser à plusieurs gènes, dont les mutations sont connues pour induire des phénotypes particuliers mais qui peuvent être associés à différentes maladies. Le séquençage par groupe de gènes s'est notamment montré très efficace dans le diagnostic de la dystrophie musculaire, de la cardiomyopathie ou de l'épilepsie dont les causes sont multigéniques[25]. La limite de cette technique réside cependant dans le choix et le nombre des gènes à séquencer.

Hybridation génomique comparative

L'hybridation génomique comparative ou *CGH array*, est une méthode cytogénétique permettant la détection d'un changement du nombre de copies (CNV) à l'échelle du génome[103]. Cette technique compare le génome du patient à un génome de référence et identifie des différences quantitatives d'ADN entre les deux génomes. L'ADN de chaque génome est marqué avec un fluorophores différent puis, les deux échantillons marqués sont déposés sur une puce à ADN sur laquelle est fixée des fragments d'ADN pouvant s'hybrider avec les séquences marquées. Un puit sur la puce présentant une sur-représentation d'une couleur particulière révéle une sur-représentation ou une sous-représentation de ce fragment chez l'un des deux génomes. Il est ainsi possible avec la *CGH array*, d'identifier des différences du nombre de copies de tailles supérieure à 50-100 kilobases avec une résolution limitée[123].

Séquençage d'exomes et de génomes

Le séquençage d'exomes est devenu la technique la plus utilisée pour la découverte de maladies génétiques mendéliennes[44, 43]. Cette dernière permet de séquencer uniquement les exons, soit 2% du génome humain. Pour cela, une étape de capture est réalisée suite à la fragmentation de l'ADN, grâce à l'utilisation d'amorces (*probes sequences*) qui s'hybrident et retiennent uniquement les fragments correspondants aux régions exoniques. Puis seul les fragments retenus sont séquencés grâce à des séquenceurs de seconde génération. Ce type de séquençage s'est montré particulièrement efficace dans la détection de variations génétiques responsables de maladies génétiques, permettant de poser un diagnostic dans 25 à 50% quelque soit le type de pathologie monogénique suspectée[143, 98]. L'avantage de cette technique est qu'elle s'affranchit d'un *a priori* sur les gènes potentiellement impliqués dans une maladie, nécessaire pour le séquençage de gènes ou par groupe de gènes.

Le séquençage de génomes est encore peu utilisé dans le diagnostic médical, notamment car la complexité d'analyse en dehors des exons est d'une toute autre mesure. Il reste néanmoins une bonne option lorsque les précédentes techniques n'ont pas donné de résultats concluants[84]. Il tend néanmoins à se démocratiser et son utilisation en première intention dans le diagnostic de maladies, a permis d'améliorer le rendement diagnostic, la prise en charge et de réduire le coût financier du diagnostic[12].

Séquençage de trio

Le séquençage de trio consiste à séquencer l'enfant et ses parents. Cela permet une identification plus sensible des variants *de novo* et des hétérozygotes composites, qui sont des variants dont les deux allèles possèdent deux variants à un même locus. Cette stratégie permet de prendre en considération les variants hétérozygotes rares observées dans chaque parent dont

l'homozygotie chez l'enfant peut être la cause de la maladie. Elle permet également d'identifier les variations *de novo* spécifiques à l'enfant malade qui peuvent être plus susceptible d'être impliquées dans la maladie[72].

Annotation des variations

La détection de variations génétiques nécessite des moyens bioinformatiques et l'utilisation d'outils spécifiques de *variant calling*, vu à la section précédente. Ces derniers identifient un ensemble des variations dont le nombre peut être de l'ordre de plusieurs millions. Cette quantité de variants rend difficile une analyse à l'oeil nu. Des méthodes d'annotation ont été développées afin de caractériser l'ensemble des variations détectées et de pouvoir les classer selon leur probabilité d'implication dans des maladies génétiques. Le principe derrière les outils d'annotations est similaire et consiste au recoupement d'informations de différentes bases de données à partir des coordonnées génomiques des variations identifiées par les *variant callers*[11, 42]. Cette annotation infère les impacts fonctionnels de chaque variant et calcule la fréquence allélique des variants dans la population. Ces informations sont ensuite utilisées pour prioriser les variations responsables d'une maladie génétique. Par exemple, les variations qui sont identifiées et retrouvées fréquentes dans des bases de données génomiques telles que *1000 Genome Project*, sont peu susceptibles d'expliquer la maladie. De ce fait, les variants identifiés avec une fréquence allélique très faible dans la population sont plus propices à être pathogènes. L'identification des caractéristiques phénotypiques associées à la maladie peut permettre d'associer un ensemble de gènes à un phénotype. Ainsi, les variations localisées dans ces gènes représentent des meilleurs candidats pour expliquer la maladie.

Protocoles standardisés de détection de variants

Du séquençage à la détection de variations génétiques, de multiples outils bioinformatiques sont nécessaires. Dans l'objectif d'automatiser la détection de variants, de nombreux *pipelines* ont été développés. Cependant face à un besoin de reproductibilité et de standardisation des pratiques dans le domaine médical, des guides de bonnes pratiques d'analyse de donnnées de séquençage ont été mis au point. Dix sept recommandations décrites Table 1.4 et 1.5, ont ainsi été proposées en 2018 par l'association de pathologistes moléculaires et le collège américain des pathologies[111]. Ces préconisations n'ont été proposées que sur un nombre limité de variants qui sont les SNP, ainsi que les indel de petites tailles (<21 paires de bases) dont l'outil de détection GATK fait consensus[89]. De ce fait, les grands indel (>20 paires de base), les variants de structure, les fusions de gènes, les translocations, les variations d'expression génique et

les variants épigénétiques ne sont pas concernés par ces recommendations. Les méthodes de détection de tels variants sont plus hétérogènes que celles pour les indel et le recul sur ces variants n'est pas aussi étoffé que pour les petits variants dans la littérature.

Les limites de la détection de variants appliquées au domaine du diagnostic médical

Comme nous l'avons vu précédemment, le choix de la technique de séquençage impacte la possibilité de détection des variations génétiques. La technologie basé sur des *reads courts* permet de détecter de manière précise des petites variations génétiques inférieures à 20 paires de bases, ne nécessitant pas d'étape d'assemblage supplémentaire pour être finement caractérisées. Les variants supérieurs à 50 kilobases peuvent être identifiés via des hybridations de génomiques comparatives mais peuvent essentiellement détecter des variations du nombre de copies, sans pour autant détecter finement leur localisation. L'utilisation de la technologie long *reads* n'est pas encore utilisée en diagnostic médical en raison de sa faible performance à détecter précisement les petites variations. De ce fait, les technologies de première ou de seconde génération restent les plus utilisées. A l'image du choix de la technique de séquençage, le choix de l'outil de *variant calling* est également une tâche compliquée. Plus de 70 *variant callers* ont été développés à l'heure actuelle pour détecter des variants de structure à partir de *reads courts*. Chacun posséde sa particularité et aucun consensus concernant la sélection des outils n'existe à ce jour. Ce constat montre la complexité à détecter des variants de structure.

Le Plan France Médecine Génomique souhaite proposer des analyses de données de séquençage standardisées pour améliorer la détection de variants et rendre le diagnostic par séquençage accessible à tous. Bien que les analyses pour des petits variants semblent pouvoir être standardisées avec plus de facilité, celles pour détecter des variants plus grands posent plus de problèmes. Des variants structurellement différents, un grand nombre de *variant callers* disponibles ainsi que des capacités de détection très hétérogènes (voir Chapitre 2) sont autant de composantes importantes à prendre en compte pour proposer des analyses performantes.

1.4 Objectifs de la thèse

La détection de variations de structure est au coeur de nombreux domaines de recherches et d'applications en biologie. Bien que les concepts informatiques menant à l'identification de variants à partir de données de séquençage de seconde génération soient définis, la détection de variants de structures reste imparfaite. Cette imperfection a conduit à une identification plus faible des variants de structure que des petits variants (SNP, indel). La détection de variants de

structure tend à être résolue par l'utilisation de données de séquençage de troisième génération. Actuellement, les défauts de cette technologie ne permettent pas un usage dans le cadre du diagnostic de maladies génétiques, qui reste sur l'utilisation de données de séquençage de seconde génération. L'amélioration de la détection de variants de structure, et plus particulièrement des insertions, est donc un enjeu crucial pour permettre d'élaborer des stratégies de détection d'insertions fiables dans le contexte médical. En effet, les insertions représentent un type de variation complexe à détecter. Plusieurs types d'insertion ont pu être identifiés, chacun avec des caractéristiques propres. L'assemblage des insertions est également plus difficile que celui des délétions. L'assemblage nécessite des approches d'extraction des *reads* associés à l'insertion, dans un contexte où plus de 50% du génome est constitué de répétitions. Par conséquent, la caractérisation des insertions est beaucoup plus faible dans les bases de données.

L'objectif de ce travail est triple : (1) comprendre les méthodologies des *variant callers* actuels pouvant être responsable d'une plus faible détection des insertions, (2) identifier les verrous biologiques et méthodologiques qui conduisent à la mise en échec des *variant callers* à détecter des insertions et enfin (3) proposer des pistes d'améliorations visant à surpasser les limites actuelles des *variant callers*.

Ce premier chapitre a permis de comprendre les mécanismes conduisant à la génération de variations génétiques, les concepts méthodologies, ainsi que les techniques pour les détecter. Nous avons pu également identifier le contexte et les limites actuelles de la détection de variations génétiques dans le diagnostic médical. Le chapitre 2 présente l'état de l'art concernant les méthodes de détection de variants de structure avec un accent sur les insertions. Ce chapitre présente également les méthodes et le matériel utilisés pour la validation du bon fonctionnement de ces outils. Le chapitre 3 présente une caractérisation fine des insertions par l'analyse de plusieurs ensembles de variants identifiés dans des récentes études. Ce chapitre présente également les propriétés des insertions qui semblent impacter la capacité de détection des *variant callers* basés sur des *courts reads*. Le chapitre 4 introduit une validation, par l'évaluation d'outils, des caractéristiques des insertions identifiées au chapitre 2 qui peuvent expliquer la mise en échec des *variant callers*. Le chapitre 5 propose un exemple d'amélioration d'un *variant caller* suite aux résultats obtenus dans les chapitre 3 et 4. Le chapitre 6 propose une conclusion à cette thèse, ainsi que les perspectives qui en découlent.

N°	Caractéristiques
1	Les laboratoires cliniques proposant des tests d'analyse NGS devraient procéder à leur propre validation des pipelines
2	Un professionnel médical qualifié ayant reçu une formation appropriée en matière d'interprétation et de certification des NGS doit superviser et participer au processus de validation des pipelines
3	La validation ne doit être effectuée qu'après l'achèvement de la conception, du développement, de l'optimisation et de la familiarisation du pipeline bioinformatique et de ses composants
4	La validation du pipeline doit reproduire fidèlement l'environnement réel du laboratoire dans lequel le test est effectué
5	La validation doit porter sur tous les composants individuels du pipeline utilisés dans l'analyse, et chaque composant doit être examiné et approuvé par un professionnel de la médecine moléculaire qualifié et par le directeur du laboratoire
6	La conception et la mise en oeuvre du pipeline doivent garantir la sécurité des informations identifiables sur les patients et être conformes à toutes les lois applicables aux niveaux local, national et de l'État
7	La validation du pipeline doit être appropriée et applicable à l'utilisation clinique prévue, aux spécimens et aux variants détectées
8	Les laboratoires doivent veiller à ce que la conception, la mise en oeuvre et la validation du pipeline soient conformes aux normes et réglementations applicables en matière d'accréditation des laboratoires
9	Le pipeline fait partie de la procédure d'essai, et ses composants et processus doivent être documentés conformément aux normes et règlements d'accréditation des laboratoires
10	L'identité de l'échantillon doit être préservée à chaque étape du pipeline, avec un minimum de quatre identificateurs uniques, y compris un identificateur de localisation unique dans le contenu de chaque fichier de données lu et/ou généré par le pipeline
11	Des paramètres spécifiques de contrôle et d'assurance de la qualité doivent être évalués pendant la validation et utilisés pour déterminer les performances satisfaisantes du pipeline

TABLE 1.4 – Recommandations de validation de pipeline de détection de variants. Table adaptée de [111].

N°	Caractéristiques
12	Les méthodes utilisées pour modifier ou filtrer les séquences utilisées à tout moment dans le pipeline avant l'interprétation doivent être validées pour garantir que les données présentées pour l'interprétation représentent avec précision et de manière reproductible ces séquences dans l'échantillon, et une documentation complète de ces méthodes doit être conservée dans le cadre de la documentation des tests conformément aux normes et réglementations d'accréditation des laboratoires
13	Les laboratoires doivent prévoir des mesures spécifiques pour garantir que chaque fichier de données généré dans le pipeline conserve son intégrité et signale ou empêche l'utilisation de fichiers de données qui ont été modifiés de manière non autorisée ou involontaire
14	La validation *in silico* peut être utilisée pour compléter la validation du pipeline bioinformatique mais ne doit pas remplacer la validation de bout en bout des pipelines bioinformatiques à l'aide d'échantillons humains
15	La validation du pipeline doit inclure la confirmation d'un ensemble représentatif de variants avec des données indépendantes de haute qualité ; des mesures de validation appropriées par type de variant doivent être indiquées
16	Les laboratoires cliniques doivent garantir l'exactitude de la nomenclature et des annotations des variants générées par le logiciel et disposer d'une alerte pour indiquer quand la nomenclature et les annotations générées par le logiciel doivent être révisées et/ou corrigées manuellement, et la documentation de toute correction doit être conservée
17	Une validation supplémentaire est requise chaque fois qu'un changement significatif est apporté à l'une des composantes de la bioinformatique

TABLE 1.5 – Suite des recommandations de validation de pipeline de détection de variants. Table adaptée de [111].

Chapitre 2

Etat de l'art : Détection de variations de structure

Le chapitre précédent a permis de mettre en évidence l'importance de la détection de variants dans le diagnostic médical. Ce diagnostic se focalise actuellement principalement sur la recherche de SNP, d'indel ou de CNV supérieurs à 50-100 kilobases à partir de données de séquençage de seconde génération. Les variations de structure (SV) et plus particulièrement les insertions restent moins et plus difficilement détectées. Cela conduit naturellement à une sous-représentation des insertions dans les bases de données référençant les variations de structure. Dans cet état de l'art, nous allons nous consacrer à la compréhension des méthodes utilisées par les *variant callers* pour identifier des variants de structure et plus particulièrement les variants de structure de type insertion. Nous étudierons les méthodes et le matériel utilisés afin de valider le bon fonctionnement des *variant callers*. Nous essaierons également de comprendre la complexité particulière associée à la découverte des grandes insertions.

2.1 Algorithmes des *variant callers*

2.1.1 Informations utilisées pour le variant calling

Les *variant callers* puisent leurs informations dans les fichiers d'alignements. Une première étape en amont de l'utilisation des *variant callers* est donc l'alignement des *reads* sur un génome de référence. Des *reads* peuvent ne pas être alignés et sont qualifiés de *reads* non alignés. Ces *reads* informent que l'individu possède des éléments nouveaux, absents dans le génome de référence. Ces éléments peuvent correspondre à des contaminations, à des symbiotes ou à de nouvelles séquences. Les *reads* peuvent être également coupés en plusieurs parties où chacune possède un alignement optimal à différentes localisations du génome. Ces *reads* appelés *split reads*, permettent notamment de détecter la position précise du point de cassure du variant, qui correspondent à la position où l'alignement a sectionné le *read*. L'orientation des *reads* est

également fournie dans les fichiers d'alignements. Des orientations anormales dans l'alignement des *reads* sont des informations qui peuvent corroborer la présence d'inversion.

La technologie *paired-end* permet d'accéder à deux nouvelles informations qui sont la distance connue entre deux *reads* d'une même paire et l'orientation de ces *reads* sur le génome de référence qui est opposé. Après avoir aligné les *reads*, il est possible vérifier si ces deux informations sont respectées. Lorsque la distance observée ou l'orientation des *reads* est différente de ce qui est attendu par le séquençage, les *reads* sont qualifiés de *reads discordants*. Cette information est l'une des premières utilisée par les variant callers puisqu'elle permet d'identifier des variants de structure selon la taille de la distance et l'orientation observées entre les paires de *reads*.

La profondeur de séquençage correspond au nombre de *reads* qui s'alignent sur une région du génome. Cette information permet d'identifier la présence de gain ou de perte de régions génomiques en analysant la perte ou le gain soudain de profondeur de séquençage dans une région donnée.

L'alignement procure également une qualité d'alignement ou *mapping quality* qui informe sur la probabilité qu'un *read* soit correctement aligné à la bonne position[77]. Cette qualité d'alignement est défini par :

$$Mapping\_quality(Q_s) = -10 \log_{10} Pr(read\ soit\ incorrectement\ aligné) \tag{2.1}$$

Un score de qualité d'alignement égal à 30 se traduit par une probabilité de 1 sur 1000 qu'un *reads* soit mal aligné sur le génome. Cette métrique est utilisée par les *variant callers* pour écarter les *reads* avec une faible qualité d'alignement.

Néanmoins l'ensemble de ces informations présente plusieurs limites qui nécessite des précautions dans leur utilisation. Lorsque des variants de structure sont d'origine duplicative ou localisés dans des régions répétées, plusieurs alignements optimaux peuvent être identifiés à différentes localisations. On parle alors de *reads* multi-alignés, dont la qualité d'alignement est faible.

Les outils de *variant calling* peuvent être divisés en deux groupes, ceux dont l'objectif est de détecter tous types de variants de structure, et ceux dont le but est de détecter un type particulier de variant. Plus de 70 *variant callers* génériques ont été développés à ce jour. Afin de permettre une plus grande clarté nous présenterons dans ce manuscrit les principales approches de détection plutôt que la description exhaustive de l'ensemble de ces outils.

2.1.2 Les *variant callers* génériques

Méthodes basées sur une seule information d'alignement

Les reads discordants

Les premières méthodes sont focalisées sur l'utilisation d'un seul type d'information d'alignement pour identifier des SV. La discordance des paires de *reads* est l'une des premières informations utilisée pour détecter des variants de structure. En effet, la distance entre les deux *reads* d'une même paire permet de déduire la présence d'une insertion ou d'une délétion. Une insertion conduit à des distances plus petites que la distance théorique, tandis que les délétions induisent des distances entre les deux *reads* d'une même paire plus grande. Cette caractéristique permet notamment à *PEMer*[64], d'identifier des insertions, des délétions, mais également des inversions qui induisent des orientations discordantes des *reads* alignés. L'ensemble des paires de *reads discordants* pour une région génomique donnée est analysé dans le but de réaliser des groupes ou *clusters* de distances entre les différents *reads* pairés. L'objectif de cette étape est de regrouper l'ensemble des *reads* qui correspondent à un même variant. Des *reads discordants* sont regroupés en clusters s'ils possèdent une distance entre les paires similaires. *BreakDancer*[37] complète cette approche en proposant en outre la détection de translocations inter et intra chromosomiques lorsque les paires des *reads discordants* sont localisées sur des chromosomes différents.Le *clustering* des *reads discordants* permet d'identifier finement la localisation des variants et de les valider par plusieurs *reads*. Dans *CLEVER*[87] ce concept de clusterisation est formalisé en un graphe d'alignement et une recherche de clique caractérisant une variation. Ce graphe est composé de sommets correspondant aux *reads* alignés dans une région donnée. Les arêtes correspondent à une distance entre deux *reads* d'une même paire similaire pour deux paires de*reads* localisés dans une même région génomique. Une clique est un sous-ensemble des sommets d'un graphe où deux sommets quelconques de la clique sont toujours adjacents. Cette stratégie, utilisée par *CLEVER* permet de regrouper ensemble, de manière efficace, l'ensemble des *reads* corroborant une même variation. Ce dernier énumère l'ensemble des cliques et identifie celles qui supportent des insertions et des délétions.

Les *reads discordants* sont donc intéressant pour identifier des régions dans lesquelles des variants de structure sont présents. La localisation des points de cassure est imprécise, puisqu'il faut que le *read* soit aligné sur toute sa longueur jusqu'au point de cassure.

Les split reads

Des statégies basées sur les *split reads* ont été développées pour détecter des grandes varia-

tions. *Pindel*[144] se base sur le principe qu'il est possible de trouver des *split reads* dits parfaits, composés de deux fragments qui s'alignent sur un génome de référence. Ces *reads* permettent de détecter précisemment le point de cassure et d'utiliser ces fragments comme ancre pour résoudre le variant avec une stratégie d'assemblage. *Splitread*[55], poursuit l'approche initiée par *Pindel* et ajoute l'utilisation des *split reads* non parfaits pour détecter des indel et SV. Ces *reads* correspondent à des *reads* dont les fragments ne sont pas sectionnés de manière égale. Cette approche permet de détecter des insertions *de novo* dont un des fragments sectionnés des *split reads* ne s'aligne pas sur le génome. *Gustaf*[130] intègre la stratégie de *CLEVER* pour l'appliquer aux *split reads* et créer un graphe de *split reads*. Dans ce graphe, les alignements des *split reads* sont les sommets et les sommets dont les alignements se chevauchent sur le génome sont liés par une arête. Ainsi les délétions sont détectées par la présence d'une liaison entre deux sommets qui sont séparés sur le génome de référence par une séquence absente dans les *split reads*. Au contraire, les insertions de type duplications dispersées sont identifiées par la présence de liaisons entre sommets dont l'un est localisé à une autre position dans le génome.

Les *split reads* sont plus précis que les *reads discordants* concernant la localisation du SV. Le point de cassure est contenu dans les *reads* conduisant à leur fragmentation lors de l'alignement. Cette localisation reste limitée lorsque le variant est localisé dans une grande répétition ou si ce dernier contient à ces extrémités des microhomologies avec le point de cassure. Ces *reads* sont également plus propices à la détection de délétions que d'insertions. La séquence et la taille de la délétion correspondent à la séquence de la référence séparant les deux fragments du *split read*. Cette approche restait néanmoins limitée avec les premiers *reads* dont la taille ne mesurait que 25-50 pb, ce qui explique une utilisation plus tardive des *split reads* que des *reads discordants*. Les *split reads* sont plus sujets à être mal alignés car les fragments, plus petit, ont une plus grande probabilité de s'aligner ailleurs sur le génome qu'un *read* entier, plus grand.

La profondeur de séquençage

La profondeur de séquençage peut être utilisée pour détecter la perte et le gain de copies. Appelées CNV, les variations du nombre de copies se réfèrent à la perte ou au gain de duplications dans le génome, généralement supérieures à 1 kilobase. Cette taille a été définie par les limites technologiques de détection de CNV induites par la *CGH array*. Le panel d'outil de détection de CNV est tout aussi vaste que celui des *variant callers* génériques, comptant plus de 80 outils à l'heure actuelle. La principale information utilisée pour détecter les CNV est la profondeur de séquençage calculée sur des fenêtres glissantes le long du génome. Cette profondeur est combinée avec des normalisations, des tests statistiques ou des modèles probabilistes

afin de prédire le gain ou la perte de copies. Il est important de noter que l'information où a lieu l'événement peut manquer, l'outil ne rapportant uniquement des différences du nombre copies.

Le génotype peut être estimé en comparant la profondeur de séquençage en un point avec la profondeur de séquençage des régions voisines. Une baisse de profondeur de séquençage dans une région peut suggérer la présence d'une délétion homozygote ou de la présence d'un variant à l'état hétérozygote.

L'assemblage

Une stratégie non basée sur une première étape d'alignement mais sur une étape d'assemblage a notamment été proposée par Li et al.[79]. L'objectif est de réaliser un assemblage *de novo* des données de séquençage puis d'aligner le résultat de cet assemblage sur un génome de référence. Cette stratégie propose plusieurs avantages, les *contigs* assemblés sont théoriquement plus grands que les *reads*. De ce fait, les alignements confèrent des informations sur de plus grandes distances. Il est également possible d'accéder à la séquence des variants s'ils ont été correctement assemblés.

Le *variant calling* par assemblage n'est pas sans défaut, il porte les défauts associés à l'assemblage. Les régions répétées sont difficiles à assembler et conduisent souvent à la production de petits *contigs*. Les duplications ne sont représentées que par un seul *contig*. Cette méthode permet uniquement d'identifier des variants homozygotes car le génome assemblé est haploïde. *Fermikit*[74] et *Assemblytics*[97] proposent une application de cette méthode.

Les stratégies utilisant une seule information restent limitées dans leur approche à détecter des SV. Puisque chaque information peut être utilisée indépendamment pour détecter des variants, des approches ont été proposées où ces informations sont combinés pour améliorer la détection de variants et réduire la détection de faux positifs.

Méthodes basées sur une combinaison de signatures

Reads discordants et profondeur de séquençage

Chen et al. dans la publication de *BreakDancer* suggérait l'utilisation de la profondeur de séquençage comme nouvelle information, autre que les *reads discordants*, pour identifier certains SV. L'augmentation de la couverture de séquençage révèle, quant à elle, un gain de duplication qui peut être dupliquée ou en tandem. *inGAP-sv* implémente cette approche, après avoir détecté des points de cassure grâce à des *reads discordants*[105].

Michaelson et Sebat proposent toutefois une approche alternative à la détection standard de variations. Ces derniers mettent en avant la complexité de détecter l'ensemble des variants

de structure tant les signaux produits par ces derniers sont nombreux, pouvant conduire à la découverte de faux positifs. Ils proposent ainsi l'utilisation du machine learning via un *Random Forest classifier*, qui va apprendre à classer différents variants à partir de données réelles du 1000 Genome Project. Sept types de variants de structure sont définis dans le *classifier* : délétion, duplication, délétion en tandem, duplication en tandem, délétion faux positif, duplication faux positif et invariant. Une matrice d'information est donnée au classifier pour lui permettre d'associer un variant potentiel à un type. Chaque ligne de cette matrice correspond à une position dans le génome. Les colonnes sont séparées en trois grandes catégories : les informations locales d'une fenêtre de 100 paires de base et deux catégories d'informations flanquantes à cette position. Chaque catégorie contient 15 données informatives, qui vont être utilisées pour identifier les SV. Cependant, le poids de chaque information diffère selon le type de variants de structure traduisant les différents signaux produits par chaque type. Cette approche a été implémentée dans l'outil *forestSV*[92].

Reads discordants et split reads

Les *reads discordants* et les *split reads* permettent d'obtenir des informations complémentaires pour un même variant. Les *reads discordants* informent sur les régions qui peuvent contenir un SV, où les *reads* s'alignent sur l'ensemble de leur longueur. Les *split reads*, indiquent précisémment les séquences flanquantes au niveau des points de cassure. *Hydra*[107] introduit cette utilisation des *split reads* pour détecter précisemment les points de cassure de variants identifiés à l'aide des *reads discordants*. *DELLY*[108] est l'outil qui a démocratisé la recherche de variants de structure basée sur ces deux informations.

Reads discordants, split reads et profondeur de séquençage

La combinaison des trois informations (*reads discordants*, *split reads*, profondeur de séquençage) a pour but d'améliorer la détection de variant, de réduire l'identification de faux positifs et de génotyper les variants. L'ensemble de ces informations permettent de confirmer ou d'infirmer la présence de variant. *Lumpy*[71] ou encore *TIDDIT*[35] sont des *variant callers* ayant implémenté cette approche.

Reads discordants, split reads et assemblage local

L'utilisation des *reads discordants* et des *split reads*, combinée à une stratégie d'assemblage pour accéder à la séquence des variants, fait partie des dernières stratégies proposées. Les points de cassure sont identifiés grâce aux *reads discordants* et aux *split reads*. L'assemblage est réalisé uniquement pour accéder à la séquence du variant et pour valider le point de cassure détecté.

Dans un but d'efficacité, un assemblage local est réalisé en utilisant seulement les *reads* proches des points de cassure. Cette approche est implémentée dans les plus récents *variant callers* *Manta*[20], *GRIDSS*[16] ou encore *svABA*[136].

Méthodes d'assemblage local

L'assemblage local consiste en un assemblage de *reads* dans une région précise. En effet, il n'est pas nécessaire d'assembler l'ensemble du génome pour obtenir les séquences des variants, seule les régions concernées suffisent. Ces méthodes vont se focaliser sur le recrutement de *reads* aux alentours des régions d'intérêt ainsi que des *reads* pouvant être associés aux variants identifiés. Deux approches sont principalement utilisées et reprennent les méthodes d'assemblage *de novo* global.

Overlap Layout Consensus et string graph

L'overlap layout consensus (OLC) est une méthode d'assemblage où les *reads* sont tous alignés les uns contre les autres. Un *overlap graph* est construit où les sommets sont les *reads* et les arêtes représentent l'existence d'un chevauchement entre deux *reads* supérieur à un seuil fixé. Une séquence consensus, appelée *contig*, est ensuite rapportée par l'exploration de ce graphe. Cette approche possède l'avantage d'utiliser toute l'information portée sur l'ensemble de la longueur des *reads*. En contrepartie, cet approche nécessite d'effectuer de grandes quantité d'alignements entre *reads*. Son utilisation pour assembler un génome entier en utilisant l'ensemble des *reads* n'est pas idéale car le graphe contient autant de sommets que de *reads* et autant d'arêtes possible qu'il peut exister entre sommets. La présence de répétitions supérieures à la taille des *reads* dans le génome à assembler conduit à des fragmentations de la séquence consensus rapportées. Les *reads* associées à ces répétitions sont reliés à des *reads* de régions génomiques différentes. Les outils qui implémentent cette approche préfèrent couper la séquence consensus avant et après le région répétée afin de ne pas rapporter une fausse grande séquence consensus. De ce fait, les outils tendent à rapporter de multiples *contigs* et non pas un seul *contig* par chromosome (voir Figure 2.1)[7]. L'OLC reste une approche envisageable dans le cas de l'assemblage de variant puisque le nombre de *reads* utilisé peut être réduit.

Le terme *string graph* est utilisé lorsque la complexité du graphe de chevauchement est réduite. Cette simplification est obtenue à travers une réduction transitive et la compression des chaînes de sommets qui possède un degré d'entrée et un degré de sortie en un unique *contig* (*unitig*, voir Figure 2.1). Ces changements permettent une exploration du graphe et une identification des régions répétées simplifiées. Cette stratégie est utilisée par *svABA* pour

assembler la séquence des variants[136].

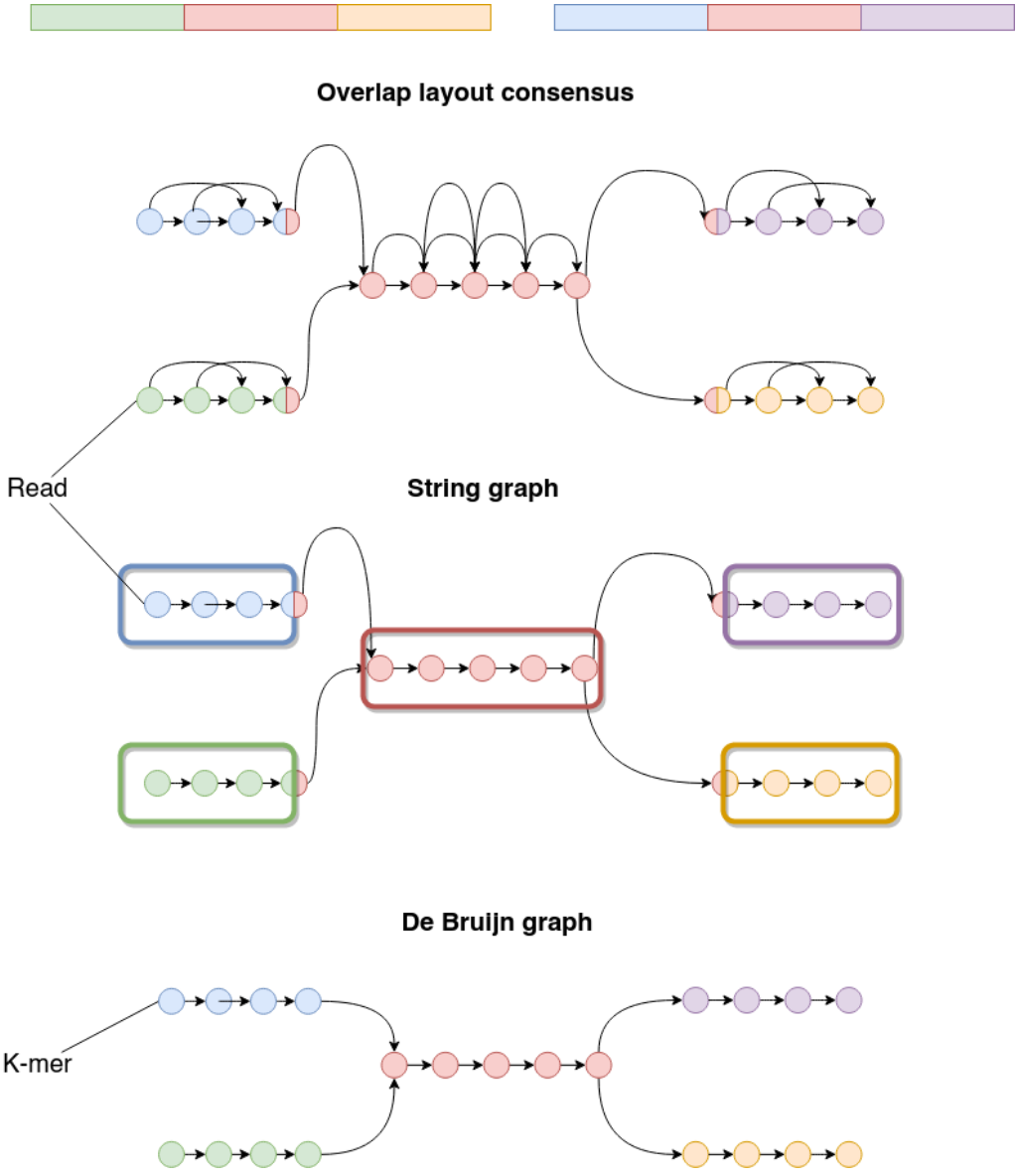

FIGURE 2.1 – Construction de l'OLC, string graph et graphe de De Bruijn et représentation des répétitions dans ces graphes, figure adaptée de [80]. Les deux segments représentent deux régions génomiques différentes. Chaque segment est composé de trois blocs dont le bloc rouge correspond à une séquence partagée entre les deux segments. Les sommets de l'OLC et du string graph correspondent à des *reads*, tandis que les sommets du graphe de De Bruijn correspondent à des k-mers. Les arêtes de l'OLC et du string graph représentent une relation entre deux reads par alignement, alors que les arêtes du graphe de De Bruijn correspond à un chevauchement exact de taille k-1 entre deux sommets. Les rectangles colorés présents dans la représentation en string graph correspond à des *unitigs*s.

Graphe de De Bruijn

Le graphe de De Bruijn est un graphe dont les sommets sont des mots d'une taille fixée k, appelés k-mers, et les arêtes entre k-mers correspondent à l'existence d'un chevauchement exact d'une taille de k-1 entre ces sommets (voir Figure 2.1). Dans le cas des données de séquençage, les k-mers sont l'ensemble des sous séquences de taille k composant tous les *reads*. Une des caractéristiques de ce graphe est l'absence de redondance dans les sommets, réduisant la taille du graphe en comparaison à l'OLC. Un k-mer même présent dans plusieurs *reads* ne sera représenté qu'une seule fois dans le graphe. L'inconvénient de cette approche est que l'information portée par le *read* entier est perdue puisqu'il est découpé en k-mers. Par conséquent, des k-mers partagés dans deux régions conduisent à la présence d'un noeud avec un degré d'entrée (sortie) de deux ou plus, indiquant l'entrée (sortie) dans un k-mer répété. Il devient ainsi nécessaire d'identifier les k-mers d'entrées et de sorties qui sont liés à la même région. Pour cela, les méthodes les plus récentes reposent sur l'alignement des *reads* sur le graphe pour identifier les liens entre k-mers sur de plus grandes distances. Cette approche reste néamoins limitée à la taille des *reads*. Le graphe de De Bruijn reste l'approche la plus démocratisée pour l'assemblage *de novo* local car elle reste la plus efficace et moins coûteuse en mémoire et temps de calcul. *GRIDSS*[16] implémente cette approche pour assembler les variants.

Les meta variant callers

Suite à la publication d'un grand nombre de *variant callers*, des *meta variant callers* ont été proposés afin d'améliorer la détection de variant. Le concept des *meta variant callers* est d'utiliser plusieurs *variant callers* sur un même jeu de données, puis d'analyser les fichiers contenant les variants pour en synthétiser un seul. Ces pipelines ne proposent donc pas de méthodes pour identifier de nouvelles variations mais des méthodes pour identifier les variants les plus vraisemblables et réduire ainsi les fausses découvertes.

La première étape de ces pipelines consiste à identifier les variants communs entre les différents vcf. *iSVP*[94] se base sur les résultats de *Pindel* et *BreakDancer*, tandis que *MetaSV*[95] utilise les résultats de *BreakSeq* et *CNVnator* en plus des deux *variant callers* cités précédemment. *SVMerge*[139] utilise pour sa part une combinaison de *BreakDancer*, *Pindel*, *SE cluster*, *RDXplorer* et *RetroSeq*. L'utilisation de plusieurs outils basés sur des méthodes et des implémentations différentes présentent l'avantage de pouvoir identifier le plus de variants possibles.

Deux approches sont possibles pour identifier des variants identiques, soit par leur proximité spatiale, soit par leur similarité de type, de taille et de localisation. Les variants détectés uniquement par un seul outil sont soit écartés, soit conservés, mais identifiés comme de faible

confiance ou de faible qualité. Néanmoins cette étape comporte de nombreux biais, présupposant que des variants identifiés par plusieurs *variant callers* suggèrent une plus grande possibilité d'être de vrais variants. Or ces *variant callers* sont basés sur l'exploration et l'intérprétation de mêmes informations d'alignements. Il est donc probable que la détection d'un faux variant soit réalisée par plusieurs *variant callers*. L'hypothèse inverse est possible, du fait d'une utilisation différente de ces informations, des vrais variants peuvent être identifiés par un seul *variant caller*.

La seconde étape consiste en un recrutement de *reads* pour l'assemblage et d'alignement local pour caractériser finement les points de cassure, le génotype et la séquence si possible. Des outils comme SVMerge propose une troisième étape d'annotation de chaque variant pour rapporter l'information du contexte génomique de chaque variant dans le vcf final[139].

2.1.3 Fichier de variations génétiques : le format *vcf*

Les variations génétiques sont rapportées dans un fichier text au format standardisé appelé format vcf (*variant call format*). Ce format permet une analyse sans ambiguïté et une utilisation facilité pour les outils de bioinformatique. Les fichiers au format vcf appelés *callsets* sont composés d'entêtes, signifié par un #, informant des spécifictés du contenu du fichier et d'*a minima* 10 colonnes. Ces colonnes sont :

1. CHROM : le nom du chromosome du génome de référence où a été identifié la variation génétique ;
2. POS : la position de la variation sur le génome de référence ;
3. ID : identifiant de la variation génétique (spécifique à chaque *variant caller*) ;
4. REF : séquence de référence associée à la position et au chromosome du génome de référence. Peut être une séquence ou le type de la variation génétique ;
5. ALT : liste des allèles alternatifs identifiés par le *variant caller*, qui peuvent être une séquence nucléique ou le type de la variation génétique ;
6. QUAL : score de qualité calculé par chaque *variant caller*, permet d'évaluer la confiance de la découverte d'une vraie variation génétique ;
7. FILTER : filtre appliqué par le *variant caller*, basé sur le score de qualité d'alignement des *reads*. La qualité "PASS" indique que le *variant caller* estime que la variation détectée est de haute qualité et est potentiellement un vrai positif. Ce champ permet de filtrer les variations génétiques pouvant être des faux positifs ;

8. INFO : informations libres délivrées par le *variant caller* et des informations concernant la séquence de référence ou alternative si elle n'est pas donné dans les colonnes REF et ALT ;

9. FORMAT : indique le format des informations concernant des informations de génotypage et de qualité des *reads* délivrées dans la colonne INDIV ;

10. INDIV : informations de l'individu analysé. Les informations sont associées à la colonne FORMAT ;

Ce format bien que standardisé autorise la détection d'un même variant sous des formes différentes. Cette flexibilité complexifie l'automatisation d'une comparaison entre plusieurs *call-sets*. Un exemple est décrit Table 2.1.

CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT	INDIV
1	15000	id_1	A	ATTCGTTT	80	PASS	Type=DUP	GT	1/1
1	15000	id_1	A	INS	80	PASS	SEQ= ATTCGTTT	GT	1/1
1	15000	id_1	A	DUP	80	PASS	SEQ= [1 :25000] ; length=9	GT	0/1
1	15000	id_1	A	A[1 :25000]	80	PASS	Type=DUP	GT	./.

TABLE 2.1 – Différents formes rapportant un même événement dans un fichier au format vcf. GT : génotype

En particulier, la présence d'homologie entre le point de cassure et une extrémité de la séquence du variants de structure peut conduire à des représentations différentes des variants (Figure 2.2). Ces représentations conduisent à des localisations et des séquences différentes pour un même variant. En 2015, Tan et al. ont proposé une uniformisation pour représenter les variants dans les fichiers vcf[127]. Cette normalisation se base sur un premier principe qu'est le principe de parismonie. Un variant est parsimonieux si et seulement si le variant et la référence sont représentés avec le plus petit nombre de nucléotide.

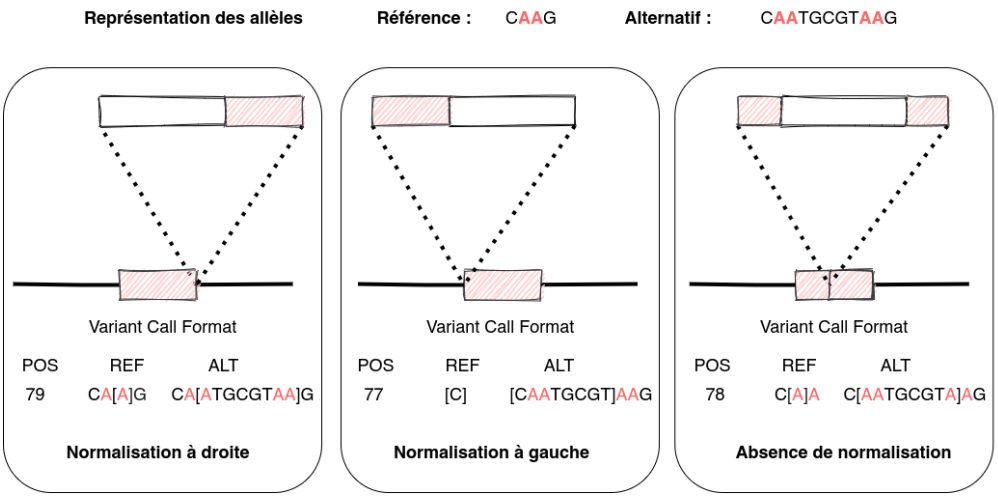

FIGURE 2.2 – Différentes représentation d'une insertion contenant une homologie avec son point de cassure. Bien que la séquence insérées soit la même, les positions d'insertions et les séquences diffèrent entre les différentes représentations. Les nucléotides entre corchets représentent la forme parsimonieuse.

Le second principe est celui de rapporter les variants en utilisant la normalisation à gauche. Un variant est normalisé à gauche si et seulement s'il n'est plus possible de déplacer sa position vers la gauche tout en gardant la longueur de tous ces allèles constantes. L'exemple au centre de la figure 2.2 présente la normalisation à gauche, où l'homologie est placée en début de la séquence et celle de la référence est placée après l'insertion. Un variant qui est normalisé est alors défini par sa représentation parismonieuse et normalisée à gauche. Cette normalisation s'est démocratisée et a permis une comparaison entre différents *callsets* plus simple à réaliser. Des outils comme vt[127] proposent une normalisation des *callsets*.

Cette normalisation à gauche permet l'identification de microhomologies entre les séquences aux points de cassure et le variant inséré. L'inconvénient de cette méthode est qu'elle est sensible aux erreurs de séquençage et aux SNP proches des variants. Ces derniers rompent l'homologie des séquences réduisant la normalisation à gauche.

2.1.4 Problèmes induits par les insertions

L'ensemble des *variant callers* génériques tendent à découvrir préférentiellement les délétions et les indels. La détection devient presque inexistante lorsque les insertions sont d'une taille supérieure à 300-500 paires de bases[15]. Les délétions possèdent des signaux très spécifiques : une perte de couverture locale des *reads* sur le génome et des *split reads* qui s'ancrent avant et

après la délétion (Figure 2.3). La séquence est obtenue en récupérant la séquence sur le génome de référence qui sépare les deux ancres des *split reads*.

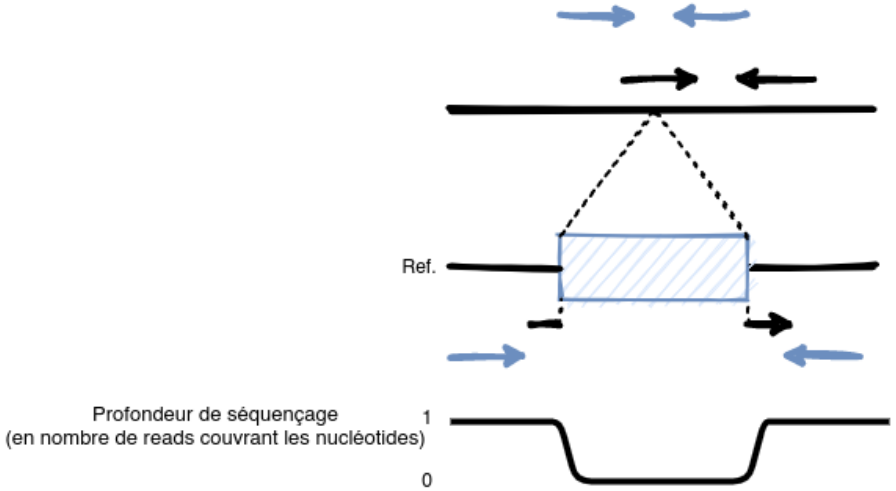

FIGURE 2.3 – Informations d'alignement produites par les variants de structure. La délétion (en bleu) est reconnaissable grâce à trois signaux forts, un distance entre les paires de *reads* plus grande qu'attendue, des *split reads* couvrant les régions avant et après l'insertion, ainsi qu'une perte de profondeur de séquençage sur la longueur de la délétion.

Les insertions n'ont pas de signaux aussi spécifiques, l'information des *reads discordants* permet de détecter une insertion potentielle si celle-ci est d'origine duplicative, car la paire de *reads paired-end* s'aligne sur le génome. Les insertions *de novo* ne possèdent pas de distance entre les paires de *reads* puisqu'une des paires ne s'aligne pas sur le génome de référence (Figure 2.4). Un gain de couverture locale permet de supposer uniquement l'existence de duplication sans informer sur la localisation de son insertion. L'identification des insertions (taille et séquence) est également plus complexe que celle des délétions. Contrairement aux délétions, la taille des insertions, supérieure à la taille des fragments séquencés, n'est pas identifiable à travers les informations délivrées par l'alignement. Il est donc nécessaire d'ajouter une étape d'assemblage pour obtenir la séquence et la taille de l'insertion.

La première étape réalisée par les *variant callers* générique pour cet assemblage est d'identifier et de regrouper l'ensemble des *reads* associés à l'insertion. L'absence d'un unique *read* peut conduire à une cassure dans l'assemblage qui est alors impossible à résoudre. L'approche standard est de recruter l'ensemble des *reads* au niveau des points de cassure, ainsi que leurs *reads* pairés. Les *reads* pairés qui sont alignés de multiple fois ou à aucune position sur le génome de référence peuvent être soit gardés soit écartés selon les *variant callers*. L'évincement de ces *reads* conduit à un échec lors de l'assemblage de duplications ou d'insertions *de novo*. De plus

lorsque l'insertion est grande, des parties de l'insertion ne sont plus rattachées à une ancre au niveau des points de cassure grâce à leur paires, complexifiant alors leur recrutement.

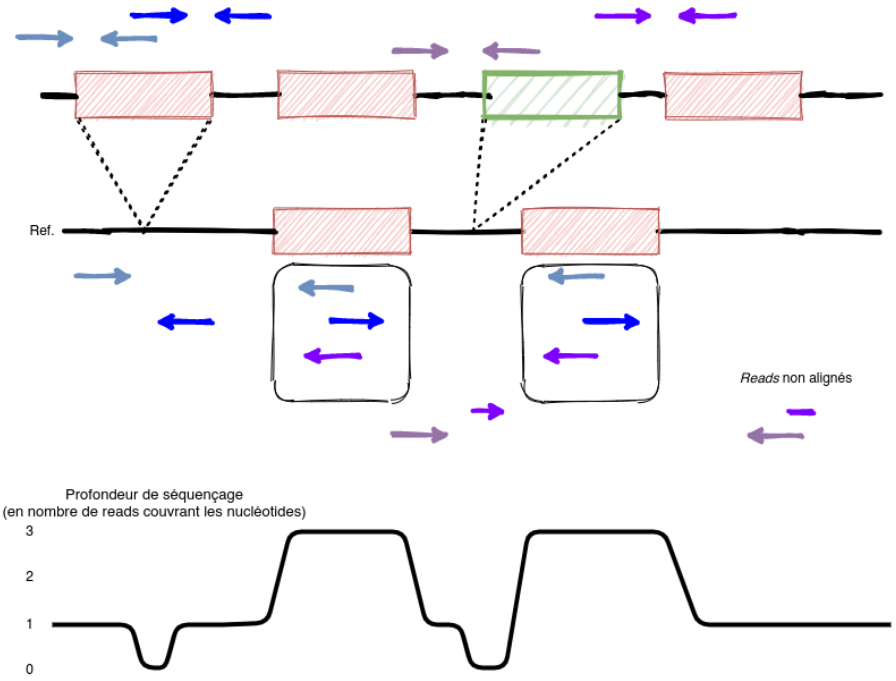

FIGURE 2.4 – Informations d'alignement produites par les variants de structure. Les signaux produits par les insertions sont plus contrastés que ceux des délétions. Les insertions de type duplicative (en rouge) induisent une distance entre paire de *reads* plus grande qu'attendue. Les *reads* associés à une duplication ont un risque d'être alignés de multiple fois sur le génome (*reads* encadrés). Ces informations ne sont pas observables avec des insertions *de novo* (en vert) car une des paires ou une partie d'un *read* ne s'aligne pas sur le génome de référence. L'ensemble des insertions produisent une perte de profondeur de séquençage locale au niveau des points de cassure. Les insertions de type duplicative induisent une augmentation de la profondeur de séquençage au niveau des copies de l'insertion.

Des stratégies itératives, comme celle proposée par *CREST*[137], tentent de résoudre ce problème en réalisant des étapes d'assemblage, puis de réalignement des *reads* sur ces séquences assemblées pour compléter entièrement l'assemblage. Cette stratégie reste coûteuse en temps et en mémoire puisqu'elle nécessite de réaligner les *reads* sur le *contig* assemblé. Une autre stratégie est de localiser les différentes positions des *reads discordants* associés aux points de cassure. Les *reads* aux abords de ces différentes positions sont recrutés et ce même si ces derniers ne sont pas directement liés aux points de cassure. L'objectif est de pouvoir recruter un maximum de *reads* susceptibles de faire partie d'une grande duplication. Pour l'assemblage des insertions *de novo*, les *reads* qui ne s'alignent pas sur le génome de référence sont recrutés. Cette stratégie

est notamment utilisée par *svABA* et *GRIDSS*[136, 16].

La seconde étape est d'identifier la séquence insérée par l'exploration du graphe d'assemblage. Cette exploration n'est pas triviale puisqu'elle est rendue complexe par la présence de répétitions, d'allèles alternatifs ou d'erreurs de séquençage. Ces derniers peuvent générer des chemins alternatifs ou bien des boucles de répétitions, qui sont difficilement résolvables lorsque les répétitions sont plus grandes que la taille des k-mers (Figure 5.5). *GRIDSS* résoud cette situation en marquant les sommets qui ont été explorés dans le graphe et n'autorise pas un nouveau passage par ceux-ci. Cette approche permet de ne pas être bloquée dans une boucle lors de l'exploration, mais elle ne permet pas de rapporter l'intégralité de la séquence si plusieurs copies sont présentes dans la séquence (solution bleue, Figure 5.5).

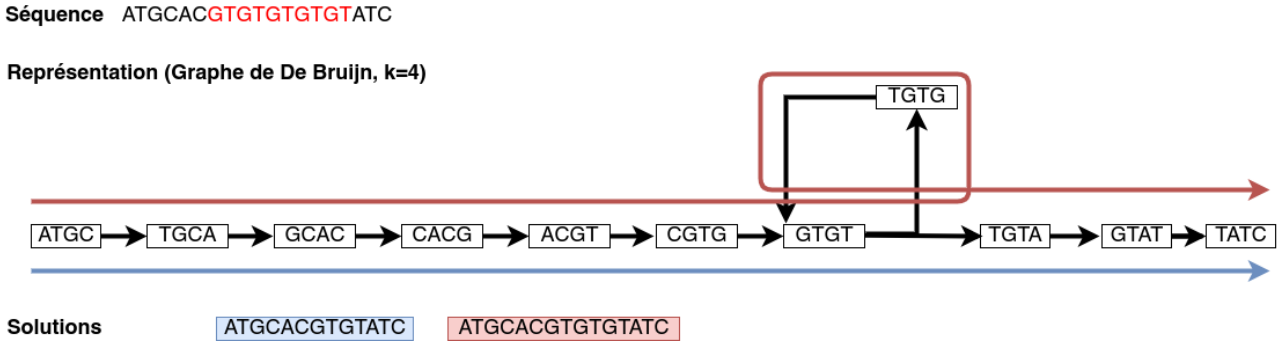

FIGURE 2.5 – Motif produit par des répétitions plus grande que la taille des k-mer dans un graphe de De Bruijn. Le chemin bleu n'autorise pas la visite de plusieurs sommets lors de l'exploration, le chemin rouge autorise une seconde visite des k-mers branchant (ici GTGT). Dans les deux cas la séquence rapportée est plus courte que la séquence originale.

2.1.5 Les variant callers dédiés aux insertions

Des méthodes dites dédiées ont été développées afin d'identifier un type d'insertion particulier. L'objectif est de rechercher des motifs caractéristiques d'un variant, limitant par conséquent la détection de faux positifs induite par des signatures partagées par différents types.

Insertions de novo

Les insertions *de novo* possèdent une caractéristique qui leur est propre : leurs séquences ne se retrouvent pas dans le génome de référence. De ce fait, les *reads* provenant de ces séquences ne s'alignent pas sur le génome de référence, hormis les *split reads* situés aux points de cassure. Partant de cette caractéristique, l'objectif est simple : rechercher les *split reads* ou les *reads discordants* dont un *read* ou un fragment de celui-ci ne possède pas d'alignement sur la référence.

L'assemblage est effectué avec des *reads* qui ne s'alignent pas sur le génome. L'inconvénient de cette approche est qu'elle ne peut assembler que les insertions qui sont entièrement nouvelles. La présence d'une portion de séquence non nouvelle dans l'insertion *de novo* peut conduire à la mise en échec de l'assemblage de l'insertion.

NovelSeq[47] a été le premier outil à introduire l'identification d'insertions *de novo* en exploitant les *reads* non alignés. La première étape consiste à regrouper les *reads* (ancres) qui s'alignent sur le génome mais dont une des paires ne s'aligne pas sur le génome. L'objectif est d'identifier des *clusters* de *reads* qui supportent une même insertion. Dans un second temps, les *reads* non alignés associés à ces clusters sont utilisés pour réaliser un assemblage local. La dernière étape est de fusionner les ancres et les *contigs* qui y sont associés. Cette approche a été complétée par *Anise-Basil*[49]. En effet, des portions de l'insertion *de novo* ne sont plus soutenus par des ancres lorsque les insertions sont trop grandes. *Anise-Basil* propose de réaliser un assemblage itératif où les premiers *contigs* assemblés sont utilisés pour un nouvel alignement afin d'identifier des nouvelles ancres et d'assembler la séquence entière. *Pamir*[60] perfectionne la détection des insertions *de novo* en ajoutant une identification du génotype des insertions découvertes. *Popins*[61] propose la découverte d'insertion *de novo* à l'échelle de génomes de population. L'approche est de récupérer l'ensemble des *reads* non alignés sur le génome de référence de chaque individu. Ces *reads* sont ensuite assemblés en *contigs*, qui sont à leur tour récupérés et utilisés pour former des *super contigs*. Les *reads* de chaque individu sont alignés sur les *super contigs* pour identifier le génotype de l'individu.

Tous types d'insertions

MindTheGap[110] propose une approche originale basée sur l'analyse de signatures d'insertion dans des graphes de De Bruijn pour détecter des insertions. L'approche implémentée par *MindTheGap* vise à comparer les k-mers du génome de référence avec ceux d'un graphe contenant l'ensemble des k-mers issus des données de séquençage. Les points de cassure des insertions homozygotes sont révélés par la présence de k-1 mers absents de manière successifs dans le graphe mais qui sont présents dans le génome de référence. Les insertions hétérozygotes sont caractérisées non pas par une absence de k-1 k-mers successifs, mais par la présence de k-mers branchants dans le graphe à une distance de k-1. Les insertions sont assemblées en utilisant les k-mers aux niveaux des points de cassure comme ancre d'assemblage. L'exploration du graphe de De Bruijn est ensuite réalisée pour assembler l'ensemble des séquences entre ces deux ancres. L'avantage de *MindTheGap* est qu'il utilise un graphe qui est composé de l'ensemble des *reads* et non pas d'un sous échantillon des données de séquençage. De plus, la signature des

insertions dans le graphe de De Bruijn n'est pas spécifique à une taille ou à un type donné et permet donc la découverte d'insertions de toutes tailles et de tous types (voir Chapitre 5 pour plus de détail).

Elements mobiles

La détection d'éléments mobiles se base sur deux caractéristiques qui sont leur répétition dans le génome et l'utilisation de connaissances sur les familles d'éléments mobiles. La première va conduire à la production de *reads discordants* dont la distance entre *reads* est grande, où chaque *read* peut même s'aligner sur un chromosome différent. La seconde caractéristique va permettre d'utiliser une méthode semblable au génotypage, où est recherché si au moins une des paires des *reads discordants* s'aligne à un élément mobile déjà connu. *MELT*[40] et *ERVcaller*[21] sont des *variant callers* dédiés qui ont implémenté cette approche.

2.2 Evaluation des variant callers

2.2.1 Objectifs

Afin de prouver l'efficacité des algorithmes et de leur implémentation dans des outils, les *variant callers* sont évalués sur leur capacité à retrouver des variants contenus dans des jeux de données de séquençage simulés ou réels. Cette évaluation permet la comparaison d'outils, également appelée *benchmark*, qui vise à identifier les outils les plus performants. Chaque nouvel outil se compare avec un ou plusieurs autres *variant callers* à disposition dans la littérature pour montrer les performances supérieures de son outil. Face à un nombre croissant de *variant callers* à disposition, supposés de plus en plus performants, des études plus "indépendantes" ont entrepris de comparer les *variant callers*. Ces comparaisons visent également à caractériser les contextes d'utilisations et les limites de ces outils. Elles sont indispensables dans l'optique de guider les utilisateurs de *variant callers*, mais également d'aider les développeurs à améliorer les *variant callers*.

2.2.2 Métriques

Plusieurs métriques ont été définies pour comparer de manière plus ou moins objective les *variant callers*. Ces métriques se basent sur la découverte de variants qui sont attendus, appelés vrais positifs (VP), ceux qui ne sont pas identifiés, appelés faux négatifs (FN), et ceux qui sont

faussement trouvés, appelés faux positifs (FP).

La précision

La précision se définit comme la quantification du nombre de variants correctement découverts (vrais positifs) par rapport à l'ensemble des variants détectés par l'outil. Cette métrique permet notamment de savoir si l'outil a tendance à découvrir des faux variants ou non.

$$precision = \frac{variants\_correctement\_découverts}{\sum variants\_découverts} \tag{2.2}$$

Le rappel

Le rappel, ou sensibilité, se définit comme la quantification du nombre de variants correctement découverts par rapport à l'ensemble des variants que le variant caller devait découvrir dans le jeu de données.

$$rappel = \frac{vrais\_positifs\_découverts}{\sum vrais\_positifs\_existants} \tag{2.3}$$

La moyenne harmonique (F-measure)

La Fmeasure se définit comme une mesure "harmonieuse" combinant la précision et le rappel.

$$Fmeasure = \frac{precision * rappel}{precision + rappel} \tag{2.4}$$

Cette métrique peut être critiquée de part le fait que la précision et le rappel ont un poids équivalent dans l'équation et que l'information de ces deux métriques ne sont plus accessible à travers cette mesure. Ainsi, la présence unique de cette métrique ne permet pas de savoir si le rappel ou la précision fait défaut.

2.2.3 Méthodes d'évaluation des variant callers

Le propos des *variant callers* est d'identifier des variations dans des jeux de données réels afin de permettre de répondre à des questions d'évolution, de génétique des populations ou de diagnostic médical. Cependant, le développement d'un outil nécessite la vérification de son bon fonctionnement sur des échantillons représentatifs ou exhaustifs. Pour cela, des données artificielles dites simulées sont générées pour tester chaque fonctionnalité.

Simulation de données

La simulation de données arbore l'avantage d'un contrôle total sur les données qui sont générées mais a l'inconvénient d'être peu représentatif des données réelles. La méthode la plus commune de simulation de données consiste en une altération d'un génome de référence, dans lequel des variants ont été insérés. Ces génomes altérés sont ensuite séquencés artificiellement à partir d'un simulateur de séquençage. Des simulateurs de génomes altérés et de séquençage ont été développés pour permettre aux développeurs de *variant callers* de tester leurs outils. Les fonctionnalités de chaque simulateur diffèrent permettant la génération de variant, du génome altéré, des *reads* ou encore de l'alignement désiré (voir Table 2.2). Malgré cela, la stratégie couramment retrouvée est l'utilisation de scripts développés spécifiquement pour la validation de l'outil. Bien que cette approche permet un contrôle total des données de la part des développeurs, elle peut conduire à un biais dans les performances de l'outil qui ne fonctionnerait que sur des données spécifiques. Cette approche "maison" induit également une difficulté à reproduire les résultats obtenus car les scripts de validation ne sont pas systématiquement disponibles.

Jeux de données réels

Lorsque les outils présentent des résultats satisfaisants sur des données simulées, une seconde évaluation est réalisée sur des données réelles. Les résultats obtenus sont comparés à un ensemble de variants de référence, également retrouvés dans la littérature sous l'appelation de *gold standard callsets*. Ces *callsets* sont obtenus par l'utilisation d'une ou plusieurs technologies de séquençage ainsi que de multiple outils de variant calling. Il est donc important de noter que les *variant callers* testés sur données réelles résultent en réalité à une comparaison avec des résultats obtenus à travers d'autres *variant callers* dont les découvertes ont été vérifiées soit via des expérimentations, soit par l'utilisation de métriques informatiques. Ces *callsets* ne sont donc pas exhaustifs et représentent généralement les variants les plus faciles à détecter. Ils sont principalement composés de SNP et d'indel et ne possèdent pas ou peu de variants de structures et souvent sans séquences résolues.

Comparaison entre callsets

L'évaluation sur données simulées ou réelles résulte souvent d'une comparaison entre *callsets* où le premier correspond au *callset* de référence et le second au *callset* à tester. Comme nous l'avons vu dans l'introduction et dans la partie raffinage des variants, différentes formes de

représentation d'un même variant sont possibles avec le format vcf. Cette flexibilité du format vcf rend difficile la comparaison entre *callsets*. A cela s'ajoute la détection des différents *variant callers* qui ne sont pas toujours capables de détecter le variant à la base près. Des méthodes ont donc été proposées pour répondre à ce problème. Ces méthodes proposent en premier lieu une uniformisation des variants détectés. La comparaison entre variants peut se réaliser en se basant sur une fenêtre de positions dont la taille est modulable.

Fonctionnalités	SVEngine[141]	RSVsim[8]	SCNVsim[106]	VarSim[96]	BAMsurgeon
Délétions, duplications en tandem	✓	✓	✓	✓	✓
Inversions, insertions, translocations	✓	✓	✓	✓	✓
Insertion *de novo* : intégration d'ADN viral	✓	✓	x	✓	✓
Ploidie modulable	✓	x	✓	x	x
Génération des génomes (FASTA)	✓	✓	✓	✓	✓
Génération des *reads* (FASTA)	✓	x	x	✓	x
Balance allélique des variants	✓	x	x	x	x
Fréquence des variants somatiques	✓	x	x	x	✓
Caractérisation exacte du point de cassure	✓	✓	x	x	✓
Séquençage modulable (ex : tailles des *reads*, couverture)	✓	x	x	x	x

TABLE 2.2 – Fonctionnalités proposées par des simulateurs de variations de structures. Table adaptée de la publication de *SVEngine*[141]

Par exemple, *Truvari* (`https://github.com/spiralgenetics/truvari`), va par défaut regarder l'ensemble des variants dont la position du début et de la fin du variant chevauche un variant de référence localisé à plus ou moins 500 paires de bases de ces positions. Ainsi, un variant détecté à 499 paires de bases de la position réelle, mais dont la séquence ou le type correspond à celui attendu, sera identifié comme vrai positif. Il est également possible d'indiquer à l'outil de ne pas comparer les séquences, qui peuvent ne pas être rapportées dans le vcf. Les métriques usuelles de précision, de rappel et de génotype correctement détecté sont rapportés à la suite de cette comparaison. L'utilisation de tels outils n'est pas très populaire car ils sont très récents, *SVanalyzer* et *Truvari* date seulement de 2017-2018. La comparaison *faite maison* reste donc la plus retrouvée au sein des présentations des *variant callers*. Les contraintes pour valider un variant sont également très hétérogènes ou peu stringentes.

2.2.4 Etat de l'art de l'évaluation des variant callers

La première évaluation d'un outil est souvent réalisée au sein de la publication présentant l'outil dans lequel sont implémentées les nouvelles méthodes de détection.

Evaluation des outils dans leur publication

Dans une publication classique, suite à la présentation de la méthode et du contexte dans lequel l'outil peut être utilisé, une évaluation sur données simulées est réalisée. De manière générale, la stratégie la plus communément observée est la génération d'un ou plusieurs types de variations simulés à des positions aléatoires sur un fragment de chromosome ou sur un chromosome. *Delly*[108] propose par exemple une validation en simulant 100 variants de type duplications en tandem, délétions et inversions sur 10 mégabases du chromosome 16. La taille des variants est peu précise, indiquée comme comprise entre 500 et 5000 paires de bases. *GRIDSS*[16] qui fait partie des plus récents outils développés propose des simulations plus complètes. Six types de variants sont testés : délétions, insertions, duplications en tandem et deux types de translocation, l'un simple, l'autre contenant des répétitions. Pour les quatre premiers types, 35 tailles de variants sont testés avec 500 variants par taille. La publication ne spécifie pas si un jeu de données a été créé pour chaque taille ou si toutes les tailles ont été réalisées sur un seul jeu de données. Pour ces deux outils, la méthode de comparaison des performances entre l'outil publié et les autres outils testés n'est pas indiquée et/ou n'est pas présente sur le répertoire public de l'outil. Cette absence d'information rend difficile pour l'utilisateur d'estimer dans quelle mesure l'outil est fiable. Les publications n'informent pas toujours si la validation des variants

de structure repose sur la détection du type, la position précise ou encore sur l'identification de la séquence. *SvABA*[136] indique pour sa part que pour la validation de la détection de SV, les variants validés peuvent se situer à 500 paires de bases de la position de référence, mais n'indique rien concernant la vérification du type ou de la séquence.

Les outils sont ensuite évalués sur des données réelles en se comparant à des ensembles de variants de structure de référence. La précision de l'évaluation reste pour la majorité aussi floue que pour l'évaluation sur données simulées. Cependant, puisque les fichiers vcf de référence sont disponibles dans des répértoires publics, il est possible d'identifier les informations disponibles à une comparaison. *GRIDSS* est évalué sur des données de séquençage de l'individu NA18278 en se comparant aux fichiers vcf produits par Sandman et al en 2015 (nstd112). L'analyse de ce fichier révèle les informations disponibles : la position, l'allèle de référence, le type de variants de structure et la taille du variant. Cependant l'absence de séquence pour l'ensemble des variants de structure ne permet pas une identification précise et rend l'information de la taille du variant une caractéristique peu vérifiable sans la séquence. *Manta* propose une validation différente, où un fichier vcf de référence est construit avec les variants identifiés par *Manta* et deux autres *variant callers* pour des données de séquençage de NA12878. Les variants identifiés par l'ensemble de ces *variant callers* sont regroupés au sein d'un vcf utilisé comme vcf de référence. Les résultats de cette approche peuvent être remis en cause de par la conception du fichier vcf de référence. En effet, l'utilisation de l'outil lui même pour créer un ensemble de variants de référence sur lequel l'outil est ensuite évalué ne permet pas de prouver sa capacité à détecter des variants quelconques et donc d'estimer son rappel.

L'évaluation des outils retrouvée dans la publication les présentant n'est donc pas exempte de biais et peut conduire à une mauvaise estimation des capacités des outils. Des études indépendantes ont ainsi été proposées afin d'évaluer les outils sous un angle plus objectif, avec des protocoles standardisés.

Evaluation des outils par des études indépendantes

Les études évaluant les *variant callers* ont pour objectif à mettre en évidence les différences entre plusieurs outils face à des données simulées ou réelles. Les protocoles ne cherchent donc pas à mettre en avant un outil en particulier contrairement aux publications d'outils. L'objectif est d'identifier les limites de chaque outil et celles des *variant callers* en général. Ces expériences sont importantes puisqu'elles proposent des voies de développement et d'amélioration pour les *variant callers*.

Kosugi et al. en 2019 proposent une évaluation d'outils avec une exhaustivité sans précédente

avec pas moins de 69 outils. Les simulations sont réalisés par VarSim dont le génome entier GRCh37 est altéré de 8310 variants. 80% de ces variations sont dérivées de variations identifiées sur des données réelles, le reste est généré artificiellement par le simulateur. Les *variant callers* sont également testés sur données réelles en utilisant les données de NA12878 provenant de la base de données DDBJ dont le *callset* contient 1671 délétions, 979 insertions, 2611 duplications et 233 inversions.

Les variants rapportés par les *variant callers* sont validés lorsqu'il existe un chevauchement supérieur à 80% et 50% avec les variants de référence simulés et réels. Les insertions sont quant à elles validées si elles sont identifiées à moins de 200 paires de bases de la position de référence. Malgré ces paramètres assouplis, les rappels des *variant callers* sont faibles. Aucun outil utilisant les *reads courts* ne réussit à détecter l'ensemble des variants. Les insertions représentent le type de variant le plus difficile à identifier avec un rappel sur données simulées n'excèdant pas 60%. Les meilleurs outils affichent un rappel inférieur à 30% sur données réelles. Seuls les outils dédiés à la détection d'éléments mobiles réussissent à détecter plus de 72% des éléments mobiles sur données réelles.

Kosugi et al. évaluent une stratégie utilisée dans des études précédentes qui concerne l'utilisation de l'intersection de plusieurs *variant callers* pour améliorer la précision de la détection[65]. Le concept derrière cette stratégie est de ne conserver que les variants qui sont découverts par plusieurs outils sur un même jeu de données. Cette stratégie est utilisée de façon systématique dans la génération de l'ensemble des variants de structure de référence. Cette évaluation a permis de mettre en évidence que, (1) les outils basés sur les mêmes méthodes de détection tendent à améliorer modérement la précision en réduisant que faiblement le rappel, (2) la combinaison d'outils avec des méthodes différentes peut conduire à une meilleur détection si les bonnes paires d'outils sont choisies, (3) l'utilisation de mauvaises paires conduira à une faible augmentation de la précision et une forte réduction du rappel. Les auteurs rappellent que les *callsets* de référence contient l'ensemble des variants les plus faciles à détecter et qu'il n'est pas invraisemblable que des faux positifs y soient présents.

La même année, Cameron et al. proposent une évaluation d'outils afin d'identifier un ensemble de caractéristiques qui contribuent à l'échec dans la détection de *variant callers courts reads*[15]. Ces caractéristiques sont divisées en deux classes, celles associées aux donées de séquençage (profondeur, taille des fragments, tailles des *reads*), et celles associées aux caractéristiques biologiques des variants (localisation, proximité des variants, type de variants). Les performances des outils sur données simulées sont sujets à des biais. Les méthodes de simulation et d'évaluation sont semblables à celle de la publication de l'outil GRIDSS, qui sont

suceptbiles de le favoriser. Néanmoins, les simulations de différentes caractéristiques de séquençage permettent de montrer leurs impacts sur la capacité de détection des outils. Par exemple, une augmentation de la taille des *reads* ou la diminution de la longueur médiane du fragment permettent de détecter des variants de plus petites tailles. Une diminution trop importante de cette longueur conduit à un chevauchement des *reads* qui induit une incapacité à détecter n'importe quel événement.

L'évaluation sur données réelles est réalisée sur quatre *callsets* de référence NA12878, CHM1, CHM13, HG002, tous avec une résolution de séquences des variants obtenus grâce à la technologie *reads longs*. La détection par les *variant callers* est validée si un points de cassure est présent à moins de 200 paires de bases de la position réelle et si la taille de l'événement diffère de moins de 25% de la taille réelle. Les résultats présentés révèlent que les *variant callers* ont des difficultés à detecter des variants dans des régions répétées ou lorsque des événements tels que des SNP ou indels sont proches des variants. Cependant, les auteurs n'indiquent pas si cette baisse d'identification est due au fait que les variants ne sont plus identifiés ou si les *variant callers* aggrègent les événements ensemble.

Bien que ces études apportent d'importantes informations concernant les limites des outils actuels, aucune n'apporte de réponse concernant la faible détection des insertions. L'une des causes mises en avant est l'absence de *callsets* de référence de qualité pour les insertions[15]. Le manque d'exhaustivité d'un *callset* de référence rend difficile de savoir si les variants identifiés par les outils testés relèvent de faux positifs ou de vrais positifs, absents du *callset* de référence. Au cours de l'année suivante, cette donne change avec la publication de deux études proposant des nouveaux *callsets* de référence ont été publiées avec une exhaustivité sans précédent[17, 148].

2.2.5 Les nouveaux callsets de références

Chaisson et al., 2019

En 2019, Chaisson et al. proposent de nouveaux *callsets* de référence et concerne trois individus : NA19240 (nigérien), HG00514 (chinois) et HG00733 (porto ricain)[17]. Pour chacun, les deux parents ainsi que l'enfant ont été séquencés pour améliorer la détection de variants. L'ensemble des variants présents dans NA19240, HG00514 et HG00733 contient donc les variants des individus associés mais également de leurs parents. Les technologies de séquençage sont diverses : *reads courts*, *reads longs*, *linked reads*, *Hi-C* et *optical mapping*. La découverte de variants est basée sur une stratégie d'assemblage des haplotypes avec les *reads longs* et par

l'alignement des haplotypes sur le génome de référence. Les *reads courts* sont principalement utilisés pour la découverte de petits variants, difficile à détecter avec des *reads longs*. Ils sont également utilisés pour conforter la découverte des variants de structure par les *reads longs*. Les 14 *variant callers reads courts* utilisés diffèrent par leurs approches afin d'identifier un maximum de variants. Certains sont génériques comme *Delly* ou *Manta*, d'autres dédiés comme *NovoBreak* ou *MELT*.

La validation des variants identifiés avec les *reads longs* réside en une comparaison des variants trouvés à travers l'utilisation de deux méthodes d'assemblage. La première est *Phased-SV* qui utilise une méthode d'assemblage local *de novo*, la seconde est *MS-PAC* qui réalise une assemblage *de novo* "non guidé". Le premier filtre pour validé un variant est la présence d'au moins 4 *reads* s'alignant sur le variant identifié. Les deux *callsets* obtenus par les deux méthodes sont ensuite comparés et regroupés pour ne former qu'un *callset*. Un variant représentatif est généré si des variants se chevauchent à plus de 50% ou si ces derniers sont proches (<1 kilobase), autrement ils sont considérés comme unique. Les variants sont ensuite gardés s'ils sont soutenus par des fragments obtenus par *Optical mapping* (BNG). Cette nouvelle technique de cartographie permet d'obtenir un ensemble de taille de très longs fragments de 200 à 500 kilobases. Cette technologie réalise dans un premier temps une digestion du génome de l'individu avec différentes enzymes de restriction. Cette étape permet d'obtenir un ensemble de fragments de taille variable. Ces fragments sont ensuite comparés aux fragments du génome de référence dont la digestion par les enzymes de restriction a été réalisée *in silico*. Les différences de taille entre fragments de l'individu et du génome de référence permet d'inférer la présence de variants de structure[145]. Seuls les variants dont les tailles identifiées par *Phased-SV* ou *MS-PAC* et celle identifiée par BNG divergent de moins de 10% sont conservés.

La validation des variants identifiés avec les *reads courts* nécessite de remplir une condition parmi quatre : (1) un chevauchement de 50% existe entre le variant et un variant identifié avec des *reads longs*, (2) au moins trois *reads* sont retrouvés sur 70% du variant, (3) si le variant est une délétion et qu'il est situé dans une région peu couverte par les *reads longs*, (4) au moins trois *reads longs* supportent l'existence du variant.

Les résultats observés montrent une proportion équilibrée entre les variants de structure de types insertion et délétion. Les biais du génome de référence sont également retrouvés, avec une observation de 21% de délétions et 9% d'insertions supplémentaires retrouvés chez l'individu nigérien que l'individu chinois. Les *reads longs* ont montré une plus grande capacité à détecter les variants de structure que les *reads courts*. L'ensemble des *variant callers reads courts* présente également une plus faible capacité à détecter les insertions. Seulement 17% des

insertions décrites ont pu être détectées par les *reads courts* contre 52% pour la détection de délétions[17]. Cette observation révèle la limite actuelle dans la découverte de variants avec des *reads courts* dans un objectif de diagnostic médical.

Zook et al., 2020

En 2020 Zook et al., du *Genome in a Bottle Consortium* (GiaB), publient une mise à jour du *callset* de référence de l'individu de référence HG002[148]. L'objectif de ce consortium est de mettre à disposition de la communauté scientifique des méthodes, des données et des *callsets* de référence pour à terme améliorer les pratiques cliniques. Ce travail a pour but de proposer un nouveau *callset* de référence de haute qualité. Cette qualité mise en avant n'a pas pour but de fournir un ensemble de variants les plus exhaustifs possibles comme Chaisson et al.. Cette mise à jour se base sur une identification combinant de multiples méthodes de séquençage et de *variant callers*. Contrairement à l'étude menée par Chaisson et al., les méthodes pour le variant calling *reads longs* ne reposent pas sur une première stratégie d'assemblage. Les outils utilisés pour le variant calling avec les *reads courts* sont également différents, seul *Manta* est utilisé par les deux études.

L'approche de regroupement des variants ainsi que leur validation diffèrent de l'étude de Chaisson et al.. L'ensemble des variants, dont la séquence est résolue, sont gardés puis analysés par *SVanalyzer*. Cet outil, non publié, propose de (1) comparer des fichiers vcf afin d'estimer le rappel et la précision, (2) identifier le contexte génomique (répétition) des variants, (3) regrouper sous une entité unique un variant identifié par plusieurs *variant callers*, (4) réaliser un nouvel assemblage pour obtenir une séquence consensus pour chaque variant. Dans cette étude, seule la fonction de regroupement est utilisée avec une distance pour regrouper les variants égale à 20% la taille du variant. Une insertion de taille 1 kilobase à une position x identifiée par un variant caller est comparée à l'ensemble des variants trouvés à x +/- 200 paires de bases dans les autres *callsets*. L'ensemble des variants rapportés par *SVanalyzer* sont conservés si et seulement si ils sont supportés par au moins deux technologies ou cinq *variant callers* ou par *optical mapping*. Puis seuls les variants dont il a été possible d'identifier un génotype via *svviz2*[122] sont conservés. Enfin l'ensemble des variants distants de moins de 1 kilobase sont retirés du *callset* final. A partir de près 300 000 variants identifiés dans 68 *callsets*, seulement 12 745 ont été conservés. L'objectif n'est donc pas de produire un *callset* le plus exhaustif possible mais un *callset* avec le moins de faux positifs. Ce *callset* propose un ensemble de variants identifiés comme les plus simples et les plus probables d'être des vrais positifs. Les *variant callers* et la méthodologie utilisés dans cette étude montrent une meilleure détection de

la part des *variant callers reads courts* que Chaisson et al.

2.3 Synthèse

Dans ce chapitre, nous avons décrit l'état de l'art des méthodes développées pour la détection des variants avec la technologie *reads courts*. Les différentes approches se basent sur une utilisation de signatures d'alignement induites par les *reads* qui correspondent aux variants. Les difficultés des *variant callers* à détecter des insertions ont conduit à leur sous représentation dans les *callsets* de référence et dans les études de variant calling. De plus, la flexibilité du format vcf rend difficile la comparaison entre *variant callers*, l'identification précise des séquences et des points de cassure. De ce fait, la validation des outils est généralement réalisée sur des positions peu précises des points de cassure, sans comparaison de la séquence assemblée. Les récents *callsets*, d'une qualité sans précédente, ont permis de mettre en évidence les limites actuelles des *variant caller* basés sur des *reads courts* dans la détection des insertions. La caractérisation des variants jusqu'à la séquence rend enfin possible de caractériser ces variants et d'évaluer les *variant callers* avec une plus grande rigueur. Le chapitre suivant présente la caractérisation de ces variants.

Chapitre 3

Facteurs impactant la détection d'insertion

Comme nous l'avons vu au chapitre précédent, deux études récentes, de Chaisson et al. et du consortium GiaB, ont mis à disposition des *callsets* inédits de référence pour le génome humain. Ces callsets sont inédits en termes de quantité mais aussi de qualité des variants de structure, en particulier pour les insertions qui ont toutes une séquence résolue. Ce chapitre présente des résultats originaux basés ces récents *callsets* de référence. Nous essayerons, à travers ces callsets, d'identifier les caractéristiques propres à chaque type d'insertion dans le but de comprendre les limites des SV callers *reads courts* actuels. Pour cela nous focaliserons notre étude sur quatre facteurs susceptibles de compliquer la détection des insertions par les *variant callers* basés sur des *reads courts*. Ces quatre facteurs sont la taille des insertions, le type des insertions, leur localisation dans des régions particulières et enfin la présence d'homologie au niveau des points de cassure marqueurs d'une réparation de brins d'ADN cassés (voir Chapitre Introduction). Ce chapitre ainsi que le suivant ont fait l'objet d'une publication dans *BMC Genomics*.

3.1 Matériel et méthodes

3.1.1 Origine des données

Notre étude porte sur quatre individus dont un *callset* de référence par individu a été produit dans les publications de Chaisson et al. et de Zook et al.[17, 148]. Trois ont été produits par Chaisson et al. (NA19240, HG00514 et HG00733), et un par Zook et al. (HG002). Les *callsets* des individus suivants ont été téléchargés à partir des liens suivants :

NA19240, 15 693 insertions : `ftp://ftp.ncbi.nlm.nih.gov/pub/dbVar/data/Homo_sapiens/by_study/genotype/nstd152/NA19240.BIP-unified.vcf.gz`.

HG00514, 14 363 insertions : `ftp://ftp.ncbi.nlm.nih.gov/pub/dbVar/data/Homo_sapiens/by_study/genotype/nstd152/HG00514.BIP-unified.vcf.gz`.

HG00733, 15 476 insertions : `ftp://ftp.ncbi.nlm.nih.gov/pub/dbVar/data/Homo_sapiens/by_study/genotype/nstd152/HG00733.BIP-unified.vcf.gz`.

HG002, 13 179 insertions : `ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTri analysis/NIST_SVs_Integration_v0.6/HG002_SVs_Tier1_v0.6.vcf.gz`.

Génome de référence, GRCh38/hg38 : `http://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz`

Seules les insertions avec une séquence assemblée et identifiée chez au moins un des parents sont conservées. Aucun filtrage lié à la qualité ou à la couverture n'est appliqué. La version du génome humain de référence pour cette étude est la version la plus récente, appelée Hg38 (GRCh38). Pour comparer les *callsets* sur le même génome de référence, le *callset* de l'individu HG002 produit sur la version hs37d5 est converti en version Hg38 en utilisant *Picard*. La version hs37d5 du *callset* est converti en une version Hg19 puis de la version Hg19 vers une version Hg38, en utilisant les fichiers chains publics du git de *GATK*[89]. Il est important de noter que ce processus peut avoir des répercussions sur quelques variants de structure, car certaines régions génomiques peuvent différer entre les versions de référence. La conversion (*liftover*) réalisé sur le *callset* de l'individu HG002 a induit une perte de 60 variants de structure.

Les méthodes de séquençage et d'identification des insertions entre les deux études diffèrent, où seul un variant caller est commun (Manta), parmis une quinzaine utilisées pour chaque étude (Table 3.1). Bien que l'étude de Zook et al. présente un nombre plus restreint de technologie de séquençage, la couverture de séquençage est supérieure à celle de Chaisson et al. Cette observation est particulièrement marquée pour le séquençage *reads courts* qui est quatre fois supérieur dans l'étude de Zook et al..

Contrairement à l'étude de Chaisson et al., les variants identifiés par le GiaB sont caractérisés par deux qualités de détection. 7 244 insertions sont signalées avec un degré de confiance plus élevé (*PASS* dans le champ *FILTER*) et 5 935 autres insertions de plus faible confiance. Comme l'ont mentionné les auteurs, la première catégorie est susceptible d'être biaisée en faveur de variants plus faciles à découvrir. Nous ne voulions pas introduire ce biais potentiel au sein de notre étude, et après avoir vérifié que ces deux catégories présentaient des distributions de caractéristiques d'insertion similaires, nous avons décidé de mener nos analyses sur l'ensemble du callset (Figure 3.5).

Etude	Individu	Technologie de séquençage	Couverture	variant callers
Chaisson et al. 2019 [17]	NA19240 HG00514 HG00733	Illumina short insert Illumina liWGS Illumina 7kbp JMP	77 3 1	dCGH, Delly, GenomeStrip, NovoBreak,Pindel, retroCNV, SVelter, VH, Wham, Lumpy, ForestSV, Manta, MELT, Tardis, liWGS
		10X Chromium BioNanoGenomics Tru-Seq SLR Strand-Seq Hi-C PacBio Oxford Nanopore (HG00733)	245 113 4 7 17 38 19	Pas de variant caller dédié Stratégie "maison" basée sur une assemblage des haplotypes et leur alignement sur un génome de référence
Zook et al. 2019 [148]	HG002	Illumina HiSeq	300	Spirale Genetics tools, GATK-HC,Freebayes, Fermikits, MetaSV, TNscope, Scalpel, SvABA, Krunch, Cortex,Manta, Seven Graph Bridge Refinement
		10X Genomics Complete Genomics PacBio	86 100 44	LongRanger CGATools PbSv Hybrid : HySA, BreakScan

TABLE 3.1 – Technologies de séquençage, couverture de séquençage et les *variant callers* utilisés pour générer les callsets de référence utilisés dans cette étude.

3.1.2 Comparaison des callsets de référence

Afin d'estimer approximativement le nombre de variants d'insertion partagé entre les différents *callsets* de référence de Chaisson et Zook, les localisations des insertions sont comparées indépendamment du type ou de la séquence d'insertion. Les insertions situées à moins de 1 000 paires de bases les unes des autres sont considérées comme un variant partagé entre les indivdus.

3.1.3 Standardisation de l'annotation des insertions

Les *callsets* de référence produits par Chaisson et al. et Zook et al. proposent une caractérisation des insertions avec leur séquence résolue. Cependant chaque *callset* propose également sa propore caractérisation du type d'insertion, Chaisson et al. proposent une caractérisation en trois types : *tandem repeat*, *mobile element* et *complex*. Zook et al., proposent quant à eux une caractérisation en trois types, différentes de ceux de Chaisson et al. : *simple insertion* (pouvant être assimilé à des insertions *de novo*), *duplication* et *sub insertion* (correspondant à une nouvelle séquence pouvant contenir des fragments dupliqués). Ces définitions hétérogènes des types d'insertions ne correspondent pas une représentation exhaustive des types d'insertions existants comme décrit dans la littérature. Ainsi, dans le but d'identifier de manière exhaustive les caractéristiques de chaque type d'insertion, une standardisation de l'annotation des insertions est requise. Cependant, à notre connaissance aucune méthode n'existe permettant l'annotation d'insertions à partir de *callset*. Dans cette partie, nous proposons une méthode d'annotation automatique des insertions à partir de tels fichiers, en se basant sur leurs séquences et le contexte génomique dans lequel elles se situent.

Définition des types d'insertions

Nous avons défini cinq types d'insertions pour décrire une insertion, en nous basant sur l'annotation de DbVar et des recommandations du format vcf :

- ***insertion de novo*** : absence de la séquence inserée dans le génome de référence ;
- ***élément mobile*** : séquence homologue à des éléments mobiles connus ;
- ***répétition en tandem*** : séquence composée d'une graine répétée en tandem ;
- ***duplication en tandem*** : séquence dont une copie est présente dans le génome de référence aux abords du point de cassure ;
- ***duplication dispersée*** : séquence dont une copie est présente dans le génome de référence à une localisation différente du point de cassure.

Un type ***non assigné*** est utilisé pour les insertions dont les séquences ne rentrent dans aucun critère de type d'insertion précédemment décrite. Nous n'avons pas défini les duplications segmentaires et les CNV comme des sous-types supplémentaires des duplications dispersées, car elles sont définies dans la littérature par leur taille (supérieure à 1 kilobase), dont le seuil a été fixé dû aux limites de détection des technologies de puce *CGH*.

Il est cependant important de noter que certains types d'insertions possèdent des caractéristiques similaires. Ainsi un élément mobile est caractérisé par un nombre de copies important

au sein du génome, copie relevant de duplications pouvant être dispersées ou en tandem. De la même façon, une répétition en tandem correspond à de multiple duplications en tandem mises bout à bout.

Méthode d'annotation des insertions

Dans le but de pouvoir caractériser finement chaque type d'insertion, un arbre de décision est utilisé en se basant sur des caractéristiques uniques de chaque type (Figure 3.1). Chaque insertion est alignée contre le génome de référence humain Hg38 pour identifier des duplications dispersées et contre les régions aux abords du site d'insertion pour identifier des duplications en tandem. Pour les duplications dispersées, chaque séquence insérée est localement alignée sur le génome Hg38 en utilisant *Blat* avec les paramètres par défaut[62]. Pour qu'un alignement soit conservé, le seuil d'identité entre séquences doit être supérieur à 90%. Les séquences qui possèdent un nombre d'alignement supérieur à 50 sont écartées de l'annotation duplication dispersée. Ce plafond vise à éviter l'annotation d'éléments mobiles en tant que duplications dispersées. Pour l'annotation des duplications en tandem, les deux séquences de chaque côté du site d'insertion et de la même taille que l'insertion sont alignées par rapport à la séquence insérée en utilisant *Blat*. Seuls les alignements présentant au moins 90% d'identité de la séquence sont conservés.

Pour pouvoir caractériser un élément mobile nous nous référons, à l'image des variants callers dédiés aux éléments mobiles, à une base de données d'éléments mobiles humains. Chaque séquence insérée est scannée par *dfam* avec la base de données standard des profils HMM des éléments mobiles humains fournie par l'outil[51].

Les répétitions en tandem sont caractérisées en analysant l'existance de graines répétées aux sein des séquences insérées. *TandemRepeatFinder* (TRF) est utilisé pour annoter les répétitions en tandem dans chaque séquence insérée[10]. Les paramètres recommandés sont utilisés, à l'exception de la longueur maximale prévue du TR (-l) qui est fixée à 6 millions (valeur conseillée pour l'analyse de séquence d'origine humaine).

Les insertions *de novo* sont caractérisées par la présence d'une séquence unique, ne contenant pas de fragments dupliqués, de graines répétées ou d'éléments mobiles. Les insertions non assignées correspondent aux insertions qui ne remplissent pas les conditions pour être associées un type.

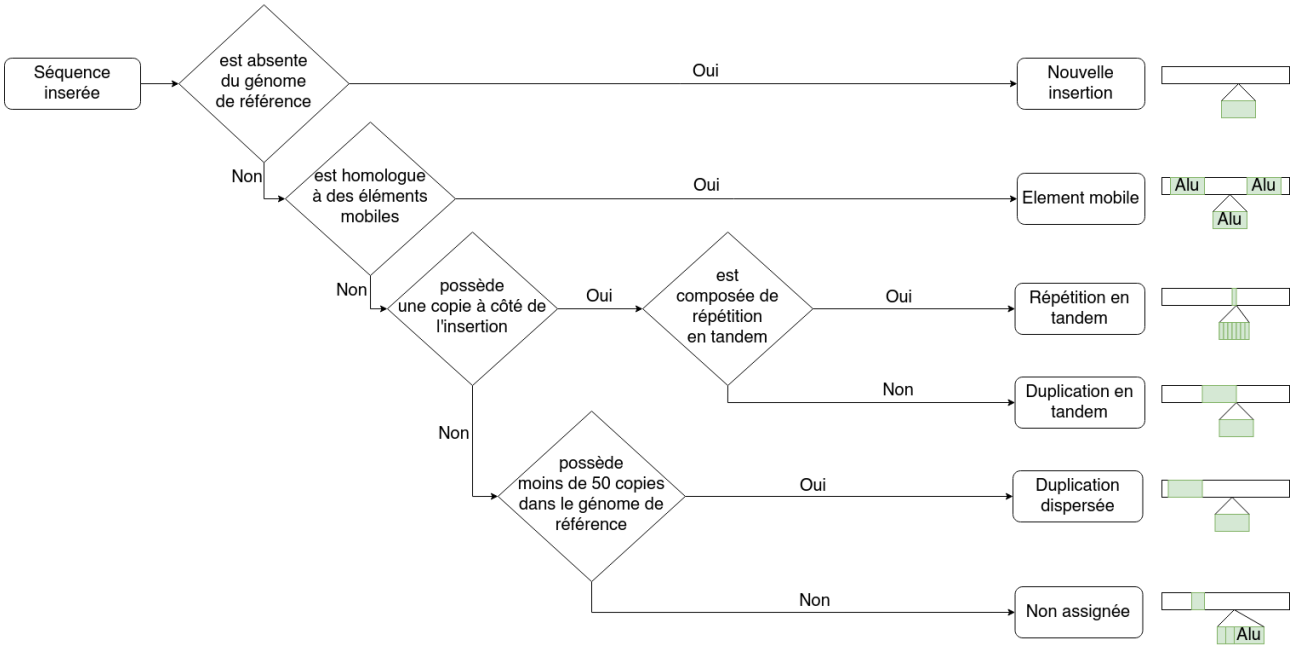

FIGURE 3.1 – Arbre de décision utilisé pour classer les types d'insertion. Cinq types d'insertion sont définis selon la nature de la séquence insérée : insertions *de novo*, répétitions en tandem, insertions d'éléments mobiles, duplications en tandem et dispersées. Les insertions non assignées contiennent des insertions qui ne répondent pas aux conditions pour être assignées à au moins un type.

Un seuil minimal de couverture de séquence, Min_{cov}, est utilisé pour annoter les insertions. Le rôle de ce seuil est d'associer le type le plus représentatif de la séquence insérée. Pour être affectée à un type d'insertion donné, la séquence insérée doit contenir au moins un segment contigu annoté avec le type correspondant et couvrant au moins $Min_{cov}\%$ de la séquence insérée. Les insertions *de novo* sont un cas particulier puisqu'elles nécessitent d'avoir Min_{cov} de leur séquence qui ne s'aligne pas sur le génome de référence. Ainsi chaque alignement est filtré selon ce seuil et les alignements en dessous de ce seuil ne sont pas utilisés. Puisque des types partagent certaines caractéristiques, un ordre de prioritisation, défini dans l'arbre décisionnel décrit Figure 3.1, est réalisé lors de l'assignation d'une insertion à un type. Ainsi une insertion, dont les alignements filtrés peuvent permettre une annotation en tant qu'élément mobile et un type de duplication, sera annotée élément mobile. L'ordre de prioritisation est le suivant :

1. élément mobile
2. répétition en tandem
3. duplication en tandem
4. duplication dispersée.

3.1.4 Localisation des insertions

Les régions répétées du génome sont connues comme étant les plus susceptibles d'être incorrectement alignées avec des *reads courts*, et donc de rendre difficile la détection de variants de structure. Nous nous sommes basés sur l'annotation RepeatMasker, distribué par l'UCSC, afin d'établir une cartographie des régions répétées. Six catégories de répétitions ont été définies :

— Répétitions simples, également appelées microsatellites ou *short tandem repeat* (STR), correspondent à de petites graines répétées en tandem. La taille des graines varie selon les auteurs pouvant aller de quelques paires de bases à quelques dizaines de paires de bases. Ces dernières représentent environ 3% du génome humain ;

— LINE (Long Interspersed Nuclear Elements), famille d'éléments transposables qui représente 21% du génome humain. Leur taille est d'environ 6 kilobases ;

— SINE (Short Interspersed Nuclear Elements), famille d'éléments transposables qui représente 15% du génome humain. Leur taille est comprise entre 100 et 700 paires de bases ;

— Autres éléments transposables, tels que les LTR (Long Terminal Repeat) ou les retroposons identifiés par RepeatMasker, correspondant à environ 15% du génome humain ;

— Duplications segmentaires, séquences d'au moins 1 kilobase qui sont présentent en de multiple copies le long du génome et qui partagent plus de 90% d'identité entre elles, représentant 4% du génome humain ;

— Régions non répétées correspondant à 50-55% du génome humain.

3.1.5 Homologies jonctionelles

L'homologie jonctionnelle, telle que mentionnée et définie par Ottaviani et al., est une séquence d'ADN qui possède deux répétitions fortement similaires aux jonctions des deux segments génomiques impliqués dans le réarrangement[102]. Dans le cas d'une insertion, une homologie jonctionnelle est un segment de séquence du côté gauche (respectivement droit) du site d'insertion qui est presque identique à la fin (resspectivement au début) de la séquence insérée. Nous étudions cette caractéristique pour plusieurs raisons. Comme nous l'avons vu dans le Chapitre 1, les mécanismes de réparation de l'ADN qui peuvent générer des variants de structure utilisent des homologies au niveau des points de cassure pour réparer l'ADN. Dans le chapitre précédent, nous avons constaté que ces homologies conduisent a une decouverte moins precise de la localisation des variants de structure (besoin de normalisation à gauche) et pourraient même empêcher leur découverte. Nous voulons donc quantifier la présence et les tailles des

homologies jonctionnelles dans un ensemble exhaustif d'insertions humaines.

Nous avons recherché les petites homologies jonctionnelles (< 10 paires de bases de chaque côté), sans autorisation de transformation, en balayant simultanément la séquence de 10 paires de bases du côté gauche (respectivement droit) du site d'insertion et la fin (respectivement le début) de 10 paires de bases de la séquence insérée. Le calcul est réalisé en comptant le nombre de nucléotides identiques successifs à partir du site d'insertion jusqu'à ce qu'une différence soit rencontrée (Figure 3.2 A).

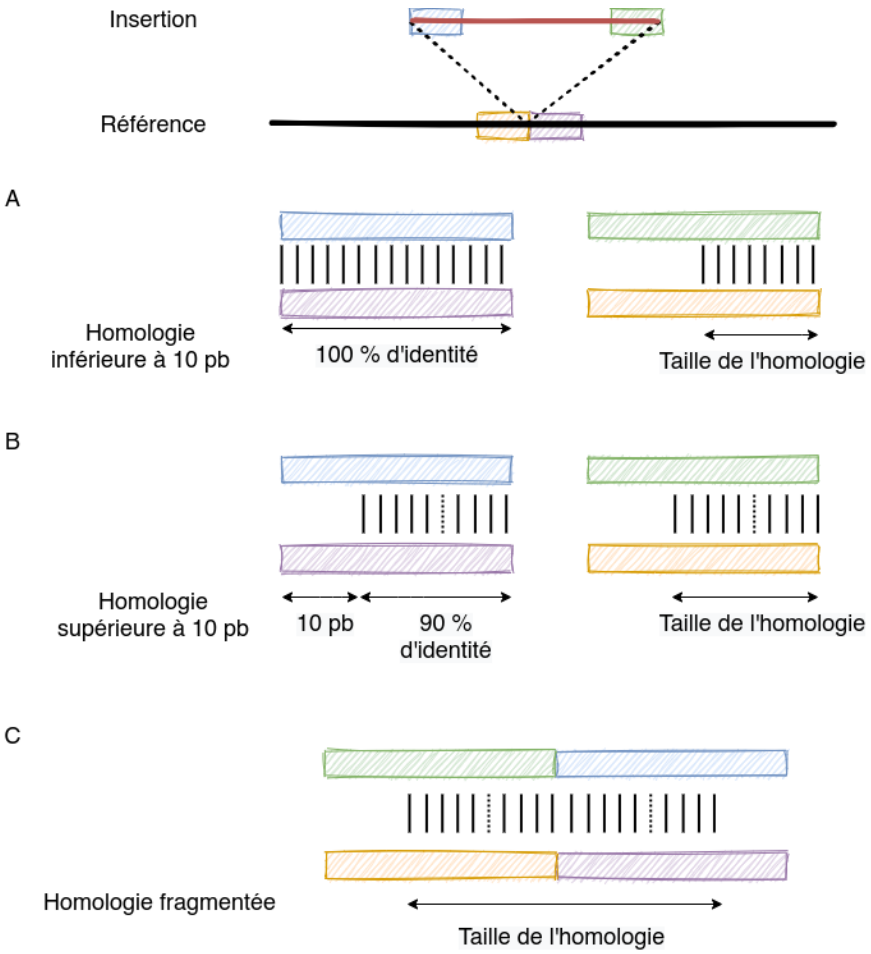

FIGURE 3.2 – Méthode de détection d'homologie jonctionnelle. A : les homologies d'une taille inférieure à 10 paires de base requière une identité de séquence de 100% entre les extremités de la séquence insérée et les points de cassure. B : les homologies d'une taille supérieure à 10 paires de base autorise une flexibilité du début de l'alignement et d'une identité de séquence de 90%. C : mesure de l'homologie jonctionnelle dans le cas ou celle-ci est fragmentée au niveau des points de cassure.

Pour les homologies plus importantes, les contraintes d'identité fixées à 100 % et de stricte

contiguïté du site d'insertion sont assouplies. Nous utilisons les alignements locaux, entre les jonctions des points de cassure et la séquence insérée, précédemment obtenus avec BLAT. Seuls les alignements ayant au moins 90 % d'identité et se produisant à un maximum de 10 paires de base avant (respectivement après) le site d'insertion et à un maximum de 10 paires de base à partir de la fin (respectivement du début) de la séquence insérée sont retenus (Figure 3.2 B). Dans le cas où plusieurs alignements existent d'un côté de la jonction, celui situé à la position la plus proche des extrémités est conservé. Si des homologies sont trouvées des deux côtés de la jonction ; la taille de l'homologie finale est obtenue en additionnant la taille des deux fragments après avoir éliminé le chevauchement potentiel sur la séquence insérée (Figure 3.2 C).

Afin de comparer nos résultats avec une distribution attendue des tailles d'homologie jonctionnelle qui peuvent être observées par hasard, nous avons généré 2 000 insertions aléatoires sur la séquence du chromosome 3 humain. Les séquences insérées ont été générées en concaténant 250 nucléotides échantillonnés uniformément sur l'alphabet A, C, G, T. Les sites d'insertion sont échantillonnés uniformément le long de la séquence du chromosome 3. Les tailles d'homologie jonctionnelle de ces insertions aléatoires sont identifiées en utilisant la même méthodologie décrite précédemment que pour les insertions réelles.

3.1.6 Rappel des variant callers basés sur les reads courts

Nous avons divisé l'ensemble des insertions des différents callsets en deux parties. La première partie est appelée *technologie reads courts* et contient les insertions portant le tag des reads courts *Illumina*. Pour les variants détectés par Chaisson et al. (NA19240, HG00514 et HG0733), la sélection est effectuée sur la section *INFO* et la variable *UNION*. La variable *UNION* peut prendre trois valeurs potentielles, *Pacbio*, *Bionano* ou *Illumina*, qui correspondent à la technologie de séquençage qui a permis de découvrir le variant. Pour le *callset* du GiaB (HG002), les insertions pouvant être découvertes par des reads courts sont identifiées par le tag *Ill* contenu dans l'information ExactMatchID de la section *INFO* du *callset*. Les insertions étiquetées avec le tag *Ill* uniquement avec des méthodologies de raffinage et non avec des méthodologies de découverte de variants sont écartées pour la partie *Technologie reads courts*.La deuxième partie, appelée *Autres technologies*, contient toutes les insertions restantes. Il convient de noter que toutes les insertions contenues dans la première partie portent également au moins une étiquette de technologie de reads longs et ne sont pas découvertes en utilisant uniquement des *reads courts*.

3.2 Résultats

3.2.1 Application de l'annotation standardisée

La méthodologie développée dans les sections précédentes est appliquée aux données produites par Chaisson et al. ainsi que par Zook et al. Plusieurs seuils de couverture de séquence minimale sont utilisés afin d'identifier l'impact du seuil dans l'annotation d'insertions. Les résultats de l'annotation des insertions de l'individu NA19240 sont décrits à la table 3.2, et montrent une constance dans la proportion des types associés malgré un taux d'éléments annotés variable selon le seuil de couverture utilisé.

% Couverture	**100**	**95**	**80**	**60**	**40**
Insertions *de novo*	677 (10%)	686 (6%)	869 (6%)	1 223 (8%)	1 639 (11%)
Elements mobiles	605 (9%)	2 047 (17%)	2 473 (18%)	2 828 (19%)	3 321 (22%)
Répétitions en tandem	4 399 (65%)	7 552 (62%)	8 735 (63%)	9 102 (62%)	9 235 (61%)
Duplications en tandem	444 (7%)	953 (8%)	1 000 (7%)	1 081 (7%)	1 082 (7%)
Duplications dispersées	486 (7%)	816 (7%)	774 (6%)	767 (5%)	713 (5%)
Non assignées	8 890	3 456	1 843	1 046	473
% annotées	43.4	78.0	88.3	93.3	97.0

TABLE 3.2 – Annotation des insertions du *callset* de référence de l'individu NA19240 selon le seuil minimum de couverture. Les valeurs entre crochet correspondent au pourcentage de représentation de la catégorie parmis les insertions annotées.

Un seuil à 100% ne permet d'annoter que 44% des insertions découvertes alors qu'un seuil à 40% permet d'en annoter 97%. Les deux types d'insertions les plus impactés par ce seuil sont les insertions *de novo* et les éléments mobiles. Un seuil à 100% ne permet d'annoter que 605 éléments mobiles, alors qu'un seuil plus bas à 80% permet d'en annoter 2 473, soit quatre fois plus. Cette différence s'explique par les différents type d'éléments mobiles existant et la présence de polymorphisme au sein des séquences d'insertions en comparaison aux séquences contenues dans la base de donnée utilisée par dfam. Dans un soucis de tenir compte du polymorphisme

des insertions décrites et de conserver une spécificité des insertions décrites comme *de novo*, un seuil de couverture de séquence minimale de 80% est fixé pour le reste de l'étude. Ce seuil permet d'annoter 88% des insertions associées à l'individu NA19240. Parmi les 12% d'insertions non attribuées, certaines peuvent correspondre à un mélange de plusieurs types d'insertion. Ce cas particulier n'ayant pas été pris en compte dans cette étude.

3.2.2 Caractérisation fine des insertions du callset de référence de NA19240

Répartition des types d'insertions

Au sein des 15 693 insertions identifiées chez NA19240, 8 735 (56%) sont annotées comme des répétition en tandem, 2 473 (16%) comme éléments mobiles, 1 000 (6%) comme des duplications en tandem, 869 (6%) comme des insertions *de novo* et 773 (5%) comme des duplications dispersées (Figure 3.3 B). En se comparant avec l'annotation réalisée par Chaisson et al., des proportions similaires sont observées sur les annotations des répétitions en tandem (57% contre 56% avec notre méthode) et des éléments mobiles (23% contre 16%). La différence observée au niveau de l'annotation des éléments mobiles est imputée au seuil de couverture de séquence minimale de 80%. Parmis les 1 843 (12%) insertions dites non assignées au seuil de 80% identifiées avec un seuil à 40% : 57% correspondent à l'annotation d'éléments mobiles, 22% à celle de répétitions en tandem, 15 % à celle de duplications en tandem et 5% à celle de duplications dispersées.

Taille des insertions

La taille des insertions est une caractéristique susceptible de limiter la détection des insertions. Comme nous l'avons vu dans le chapitre dédié à l'état de l'art, l'identification d'insertions plus grandes que la taille des *reads* représente une détection et une résolution plus complexe que pour des petites insertions.

Concernant la taille des insertions de l'individu NA19240, 67% des insertions présentent une taille inférieure à 250 paires de bases et seulement 8% possèdent une taille supérieure à 1 kilobase (Figure 3.3 A). La distribution des tailles est différente en fonction du type d'insertion (Figure 3.4). Les éléments mobiles présentent une forte sur-représentation d'insertion de taille entre 250 et 500 paires de bases (61%). Ce phénomène s'explique par la caractéristique d'éléments tels que les SINE, dont la taille avoisine les 300 paires de bases. Les insertions *de novo* présentent

une plus grande proportion d'insertion de grande taille que les autres types d'insertions, dont 164 (19%) qui ont une taille supérieure à 1 kilobase.

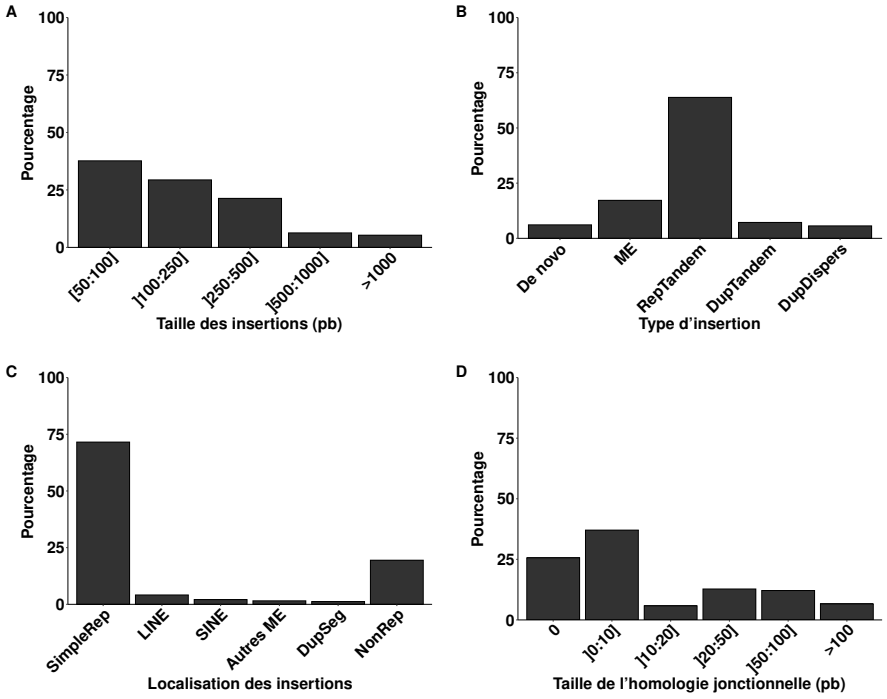

FIGURE 3.3 – Caractérisation fine des insertions du callset de NA19240 produit par Chaisson et al.. A : Distribution des insertions en fonction de leur taille. B : Distribution des types d'insertions annotés par notre méthode. C : Distribution des insertions selon leur localisation dans des régions répétées. D : Distribution de la taille des homologies jonctionnelles. ME : élément mobile ; RepTandem : répétition en tandem ; DupTandem : duplication en tandem ; DupDispers : duplication dispersée ; DupSeg : duplication segmentaire ; SimpleRep : répétition simple.

Localisation des insertions

Une forte sur-représentation est constatée dans les régions annotées comme des répétitions simples, avec 9 675 (70%) des insertions annotées de l'individu NA19240, qui sont situées dans ces régions qui ne représentent que 3% du génome (Figure 3.3 C). Le contexte génomique préféré des insertions varie selon le type d'insertion (figure 3.4). 8 047 (92%) répétitions en tandem, 723 (73%) duplications en tandem et 519 (63%) duplications dispersées ont été trouvées dans les régions de microsatellites. Inversement, 580 (67%) insertions *de novo* et 1 383 (56%) insertions d'éléments mobiles ont été localisées dans d'autres régions. Nous n'avons pas trouvé un taux plus élevé d'insertions dans les régions exoniques, introniques ou intergéniques par rapport à

une distribution uniforme le long du génome.

Homologies jonctionnelles

Parmis les 15 693 insertions de l'individu NA19240, 5 119 insertions (38%) ont montré des homologies jonctionnelles supérieures à 10 pb (figure 3.3 D). Cette proportion est supérieure à celle obtenue avec les insertions de séquences aléatoires, la plus grande homologie jonctionnelle observée étant de 7 paires de bases parmi 2 000 insertions simulées de façon aléatoire. Tous les types d'insertions présentent des homologies jonctionnelles plus importantes qu'attendu avec des insertions aléatoires. Les duplications en tandem et les répétitions en tandem sont les types présentant les plus grandes homologies jonctionnelles, avec 428 (43%) répétitions en tandem et 1 751 (20%) répétitions en tandem qui ont été identifiées avec une homologie jonctionnelle supérieure à 50 paires de bases (figure 3.4 C).

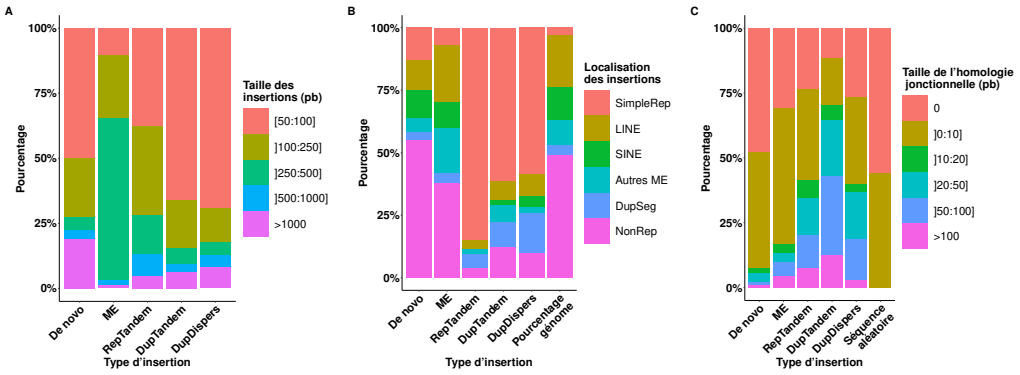

FIGURE 3.4 – Caractéristiques associées à chaque type d'insertion pour les insertions contenues dans le callset NA19240. A : Répartition de la taille des insertions. B : Répartition de la localisation des insertions. C : Répartition des homologies jonctionnelles par type d'insertion. ME : élément mobile ; RepTandem : répétition en tandem ; DupTandem : duplication en tandem ; DupDispers : duplication dispersée ; DupSeg : duplication segmentaire ; SimpleRep : répétition simple.

Comme prévu par leur nature en tandem, les répétitions en tandem et les duplications en tandem présentent des homologies de plus grande taille que les autres types d'insertion. Toutefois, pour beaucoup d'entre elles, la taille de l'homologie est inférieure à celle de l'insertion. L'explication des répétitions en tandem réside dans leur structure qui est une amplification d'une graine dans le génome de référence. Ainsi, la plus grande taille d'homologie correspond à la taille de la graine présente au point de cassure de droite (en cas de normalisation, de gauche). Comme pour les duplications en tandem, il est attendu que la taille de l'homologie jonctionnelle

correspond à la taille de l'insertion. Cependant cette conformation n'est pas systématiquement observée, en cause, la différence dans les méthodes utilisées pour définir l'homologie et l'annotation d'une duplication en tandem. L'annotation est plus flexible et ne nécessite la présenve d'un fragment couvrant 80% de la séquence insérée, où une correspondance plus stringente est requise pour identifier une homologie jonctionnelle. Ce résultat suggère que de nombreuses duplications en tandem présentent de petites variations avec leurs copies adjacentes qui rompent la correspondance exacte.

3.2.3 Comparaison des insertions entre individus

Les précédentes observations ont été effectuées sur l'individu NA19240 de Chaisson et al.. Nous nous sommes intéressés à savoir si ces dernières pouvaient être similaires chez des individus d'origines génétiques différentes. Nous avons tout d'abord considéré les deux autres individus de l'étude de Chaisson et al., à savoir : HG00514 (14 363 insertions) fils d'un trio chinois Han (CHS) et HG00733 (15 476 insertions) fils d'un trio portoricain (PUR). Ces *callsets* ont été obtenus avec les mêmes technologies de séquençage et les mêmes méthodologies de variant calling que pour l'individu NA19240.

Ensuite, nous avons analysé un *callset* obtenu par une étude différente, à savoir celui de l'individu HG002 (13 179 insertions) fourni par le Consortium Genome in a Bottle (GiaB)[148]. Dans cette étude, Zook et ses collègues ont également utilisé de multiples technologies de séquençage et de multiples *variant callers* pour obtenir un callset d'insertions et de délétions de haute confiance. Cependant, les méthodes de séquençage et de variant calling différent de celles utilisées par Chaisson et al. (voir Chapitre 2).

Avant de comparer les caractéristiques des insertions entre les différents individus, nous avons d'abord vérifié si elles contiennent des variants différents. En utilisant une estimation approximative des variants partagés, nous avons identifié que seulement 1 169 sites d'insertions sont communs aux quatre individus dans une fenêtre de 1 kilobase. En moyenne, 3 344 insertions sont partagées entre deux individus, et globalement, plus de 55% des insertions étudiées sont spécifiques à un individu.

La comparaison entre les différents individus montre que les distributions des types d'insertion, des tailles, des emplacements et des tailles d'homologie jonctionnelle sont similaires entre les trois individus de l'étude de Chaisson et al. et le callset du GiaB (Figure 3.5 et Figure 3.6).

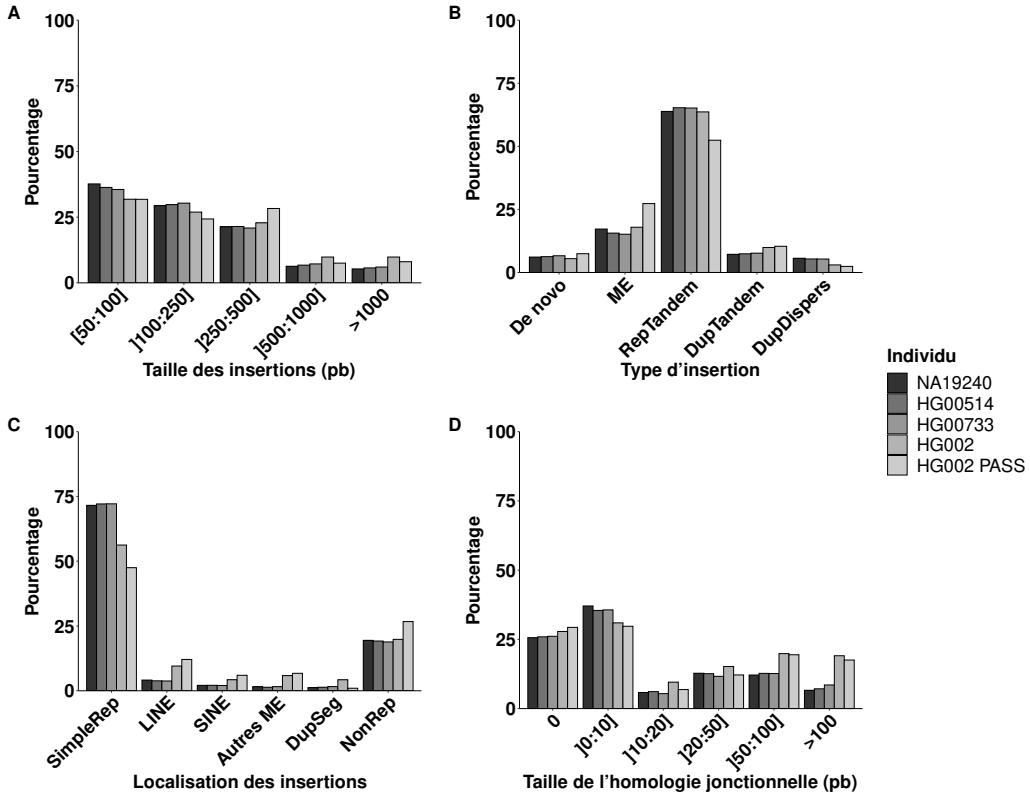

FIGURE 3.5 – Caractéristiques associées à chaque type d'insertion pour les insertions contenues dans les différents callsets. A : Distribution de la taille des insertions. B : Distribution de la localisation des insertions. C : Distribution des homologies jonctionnelles par type d'insertion. ME : élément mobile ; RepTandem : répétition en tandem ; DupTandem : duplication en tandem ; DupDispers : duplication dispersée ; DupSeg : duplication segmentaire ; SimpleRep : répétition simple.

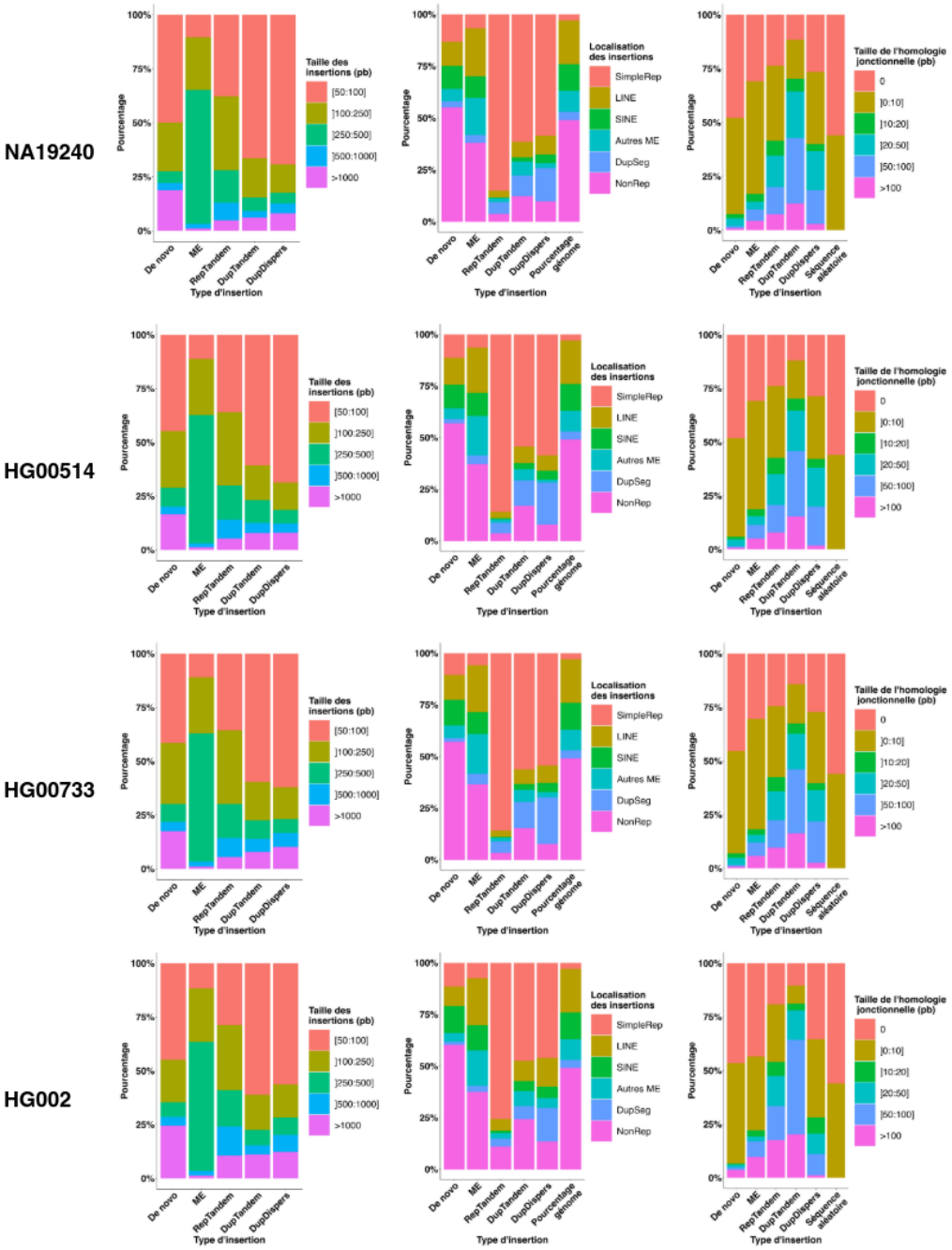

FIGURE 3.6 – Caractéristiques associées à chaque type d'insertion pour les insertions contenues dans les différents callsets. Première colonne : Proportion de la taille des insertions pour chaque type d'insertions. Seconde colonne : Proportion de la localisation des insertions pour chaque type d'insertion. Troisième colonne : Proportion des homologies jonctionnelles pour chaque type d'insertion.

3.2.4 Rappel des variants callers courts reads

Afin d'étudier si les caractéristiques d'insertion décrites précédemment ont un impact sur le rappel des variants callers de reads courts, nous avons reproduit l'analyse précédente en fonction de la technologie utilisée pour les identifier.

Pour l'individu NA19240, 17% insertions sont détectées par des *variant callers* avec des reads courts. Comme le montre la figure 3.7, la détection par les variants callers reads courts est très hétérogène en fonction des caractéristiques d'insertion décrites précédemment. Chaque caractéristique décrite dans ce travail (la nature et la taille de la séquence insérée, le contexte génomique du site d'insertion et les homologies jonctionnelles) a un impact sur la détection d'insertion avec des reads courts. Les insertions de plus de 500 paires de bases sont mal découvertes(< 3 %), même si une détection accrue est observée pour la classe de taille d'insertion de 250-500 paires de bases qui correspond aux insertions d'éléments mobiles (Figure 3.7 A). La plus grande différence dans la détection d'insertions avec des reads courts est observée parmi les types d'insertion : 57% des éléments mobiles et 40% des insertions *de novo* sont détectés avec des variants caller reads courts, contre seulement 9% des duplications en tandem, 6% des répétitions en tandem et 5% des duplications dispersées (Figure 3.7 B). Une baisse de détection est observée dans des régions répétées : les insertions dans des régions de microsatellites sont parmi les plus difficiles à découvrir (5%) (Figure 3.7 C). Concernant l'homologie jonctionnelle, plus l'homologie est importante, plus la détection de l'insertion est difficile (figure 3.7 D). Seules 5% des insertions situées dans des répétitions simples sont trouvées par des variants callers reads courts.

Les observations faites sur les deux autres individus HG00514 et HG00733 sont très similaires avec celles de NA19240. Alors que les observations générales sont similaires entre l'ensemble des individus de Chaisson et al., elles deviennent très différentes entre les deux études. Premièrement, dans l'ensemble, l'étude du Giab (Zook et al.) détecte 1,6 fois plus d'insertions avec des *reads courts* que l'étude de Chaisson et al. (24% et 17% d'insertions identifiées dans HG002 et NA19240 respectivement). Deuxièmement, la détection est plus homogène en ce qui concerne les caractéristiques d'insertion avec la méthodologie du GiaB (Figure 3.7). La caractéristique qui a le plus d'impact est la taille de l'insertion, avec une détection de seulement 1% des insertions d'une taille supérieures à 1 kilobase (figure 3.7 A). Comme pour NA19240, les répétitions en tandem semblent plus difficiles à découvrir avec les méthodes basées sur les reads courts, mais dans une moindre mesure dans le *callset* de HG002 (figure 3.7 B). Les insertions situées dans des répétitions simples sont également moins découvertes avec les *reads courts*. Néanmoins cette détection de 24% reste plus élevée que pour NA19240 où elle n'atteint que 5% dans ces régions (figure 3.7 C). Contrairement à l'étude de Chaisson et al., l'homologie jonctionnelle des

insertions de l'individu HG002 n'arbore pas la même tendance. Une diminution constante de détection avec des *reads courts* est associée à une augmentation de la taille des homologies pour les individus de Chaisson et al. L'impact d'homologie jonctionnelle sur la détection avec des *reads courts* est plus hétérogène avec le *callset* de HG002. Cela semble indiquer une absence de relation entre la taille de l'homologie et la capacité à détecter des insertions avec des *reads courts* (figure 3.7 D).

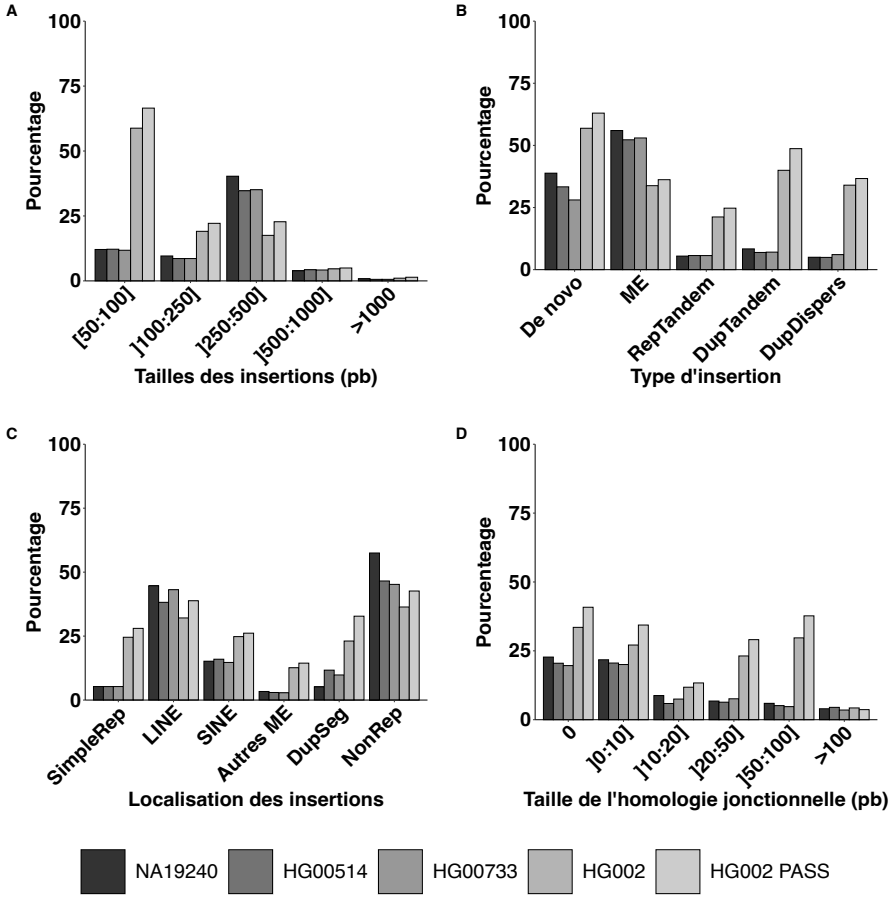

FIGURE 3.7 – Rappel de la découverte d'insertion selon le type de technologie de séquençage utilisé sur les différents callsets de référence. A : Rappel de la découverte d'insertions avec des reads courts en fonction de la taille des insertions. B : Rappel de la découverte d'insertions avec des reads courts en fonction des types d'insertion annotés par notre méthode d'annotation. C : Rappel de la découverte d'insertions avec des reads courts en fonction de leur localisation dans des régions répétées. D : Rappel de la découverte d'insertions avec des reads courts en fonction de la taille des homologies jonctionnelles présentent dans les insertions. Les pourcentages sont calculés à partir du nombre total d'insertion présent dans chaque callset. ME : élément mobile ; RepTandem : répétition en tandem ; DupTandem : duplication en tandem ; DupDispers : duplication dispersée ; DupSeg : duplication segmentaire ; SimpleRep : répétition simple.

3.3 Discussion

Nous avons présenté ici l'une des analyses les plus détaillées et les plus complètes des insertions réelles dans le génome humain, en cherchant les facteurs ayant un impact sur leur détection à l'aide de données de séquençage de seconde génération. Cela a été rendu possible grâce à la publication de quatre *callsets* exceptionnels par Chaisson et al.[17] et Zook et al. (GiaB)[148]. Ces catalogues d'insertions sont considérés comme les plus exhaustifs pour un individu humain donné grâce à leurs validations approfondies par des ensembles de données de séquençage exhaustifs et croisés. Non seulement, ces catalogues d'insertions sont considérés comme les plus exhaustifs pour un individu humain donné, mais ils sont également les premiers à comporter des insertions avec une résolution de la séquence pour toute taille et tout type d'insertion.

3.3.1 Annotation des insertions

La résolution fine des séquences insérées, présentes dans ces ensembles de données, nous a permis de proposer une classification affinée des variants d'insertion. Dans les deux ensembles de données, les types d'insertion ne sont pas formellement définis et les classifications différent entre les ensembles de données. Notre classification a permis de normaliser ces annotations hétérogènes et se présente comme une application directe des définitions de variants de la base de données dbVar[70]. Nous avons basé notre annotation de type d'insertion sur un seuil minimal de couverture de séquence, qui a été fixé à une valeur relativement élevée, 80%, afin d'assurer une bonne spécificité de notre annotation. L'augmentation de cette valeur conduit à un plus grand nombre d'insertions non assignées, car les annotations sont basées sur des alignements de séquences qui sont affectés par des erreurs de séquençage potentielles restantes dans les séquences insérées, le polymorphisme avec le génome de référence et l'utilisation d'heuristiques d'alignement. Si le nombre d'insertions non assignées diminue avec la valeur du seuil de couverture, les proportions des différents types d'insertion restent néanmoins stables.

Comme indiqué précédemment dans les études de Chaisson et al. et de GiaB, nous avons observé une distribution très hétérogène des types d'insertion et de leur emplacement le long du génome. La grande majorité des insertions consiste en des répétitions en tandem (63%) et la plupart des sites d'insertion sont situés dans des régions avec des répétitions simples (70%). Ces régions dites de faible complexité, bien que représentant une faible proportion du génome (1,2 %), sont donc une source majeure de variabilité inter-individuelle.

3.3.2 Caractérisation des insertions

La résolution de la séquence insérée fournie dans ces callsets nous a également permis d'analyser précisément les jonctions au niveau des points de cassure de chaque insertion. Il a été démontré que l'homologie jonctionnelle est une caractéristique fréquente des variants, qui peut être utilisée pour déduire le mécanisme moléculaire de réarrangement [28, 63]. Bien que des analyses d'homologies ont déjà été décrites pour des variants de structure humains (environ 2 000 variants, contenant moins de 400 insertions) [63], il s'agit, à notre connaissance, de la première quantification exhaustive de l'homologie jonctionnelle pour un ensemble aussi important et presque complet d'insertions chez un individu humain. Cependant, notre mesure de la taille de l'homologie dépend fortement de la précision de la localisation du site d'insertion et de la séquence insérée. Comme les variants de structures sont souvent difficiles à localiser avec précision, qu'elles sont soumises à des processus de normalisation à gauche et que leurs séquences insérées ont été obtenues pour la plupart à partir de longues lectures sujettes à des erreurs, nos mesures peuvent probablement aboutir à une sous-estimation des tailles réelles d'homologies.

Malgré ces biais potentiels, nos résultats montrent que les insertions réelles présentent des homologies jonctionnelles nettement plus importantes que les insertions qui sont tirées au hasard. Nos mesures nous ont permis de comparer cette caractéristique entre les types d'insertion et il a été constaté que tous les types d'insertion présentent une proportion substantielle avec de grandes homologies jonctionnelles (supérieures à 20 paires de bases). Les résultats montrent également que les insertions de grande taille ont tendance à comporter des homologies jonctionnelles plus importantes.

3.3.3 Impact sur le rappel des variant callers avec reads courts

Toutes les caractéristiques d'insertions identifiées dans notre étude (c'est-à-dire la nature et la taille de la séquence insérée, le contexte génomique du site d'insertion et les homologies jonctionnelles) montrent avoir un impact sur la détection avec des reads courts. Cependant, une différence importante a été observée entre les deux études, l'étude de Zook et al. étant capable de détecter 1,6 fois plus d'insertions que dans l'étude de Chaisson et al. avec des reads courts. La différence entre les deux études, en ce qui concerne la détection avec des reads courts, s'explique certainement par les différences dans les profondeurs de séquençage Illumina, les différents ensembles d'outils utilisés, ainsi que les différentes méthodologies de filtrage et de fusion des *callsets*. Les deux études ont utilisé à peu près le même nombre de

variant callers (13 et 15), mais avec une faible intersection : un seul variant caller (Manta) est commun aux deux études. En outre, l'annotation de la méthode de chaque variant dépend fortement de la méthodologie de l'étude pour filtrer et fusionner les nombreux *callsets* obtenus pour un même individu avec des technologies de séquençage et variants callers différents. Par exemple, il n'est pas clair si les tags associés à la détection avec des *reads courts* présents dans les *callsets* signifient nécessairement que le variant peut être découvert et résolu à la séquence près. Cependant, les deux études montrent des faiblesses similaires pour détecter les répétitions en tandem, les insertions de grande taille et les insertions situées à l'intérieur de répétitions simples.

Ces observations ont permis d'identifier des caractéristiques pour chaque type d'insertion dans un contexte de données réelles. Ces dernières semblent corrélées et rendent difficile la possibilité d'identifier le facteur qui impacte le plus les variants callers utilisant des reads courts. Pour illustrer ce propos, nous pouvons remarquer que les répétitions en tandem ont tendances à être localisées dans des répétitions simples. Il est ainsi difficile de savoir si ce sont les régions simples ou les répétitions en tandem qui posent problème pour les variants de structure basés sur des *reads courts*. Le chapitre suivant visera à identifier l'impact de chaque facteur indépendemment des autres en utilisant des simulations.

Chapitre 4

Evaluation des limitations des outils de détection courts reads

Dans le chapitre précédent, nous avons pu analyser divers types d'insertions à partir de callset de haute qualité. Les caractéristiques variées associées à ces insertions semblent, dans l'ensemble, avoir un impact sur la capacité des *variant callers* à détecter des insertions avec des *reads courts*. Cependant, dans ces jeux de données d'insertions réelles, la plupart de ces facteurs sont corrélés ce qui rend difficile l'identification du facteur qui impacte le plus les *variant callers*. Dans ce chapitre, nous verrons comment nous avons simulé ces différents facteurs indépendamment des autres, dans le but d'identifier l'impact des différents facteurs pouvant limiter la détection de *variant callers*. Pour cela nous testerons différents *variant callers*, génériques ou spécialisés dans la détection d'insertion, qui sont *Manta*[20], *svABA*[136], *GRIDSS*[16] et *MindTheGap*[110].

4.1 Matériel et méthodes

4.1.1 Simulations

Vingt-deux ensembles de données de séquençage ont été simulés pour caractériser l'impact des différentes caractéristiques d'insertion, sur la détection de *variant callers* utilisant des *reads courts*. Chaque ensemble de données est obtenu en modifiant le chromosome 3 humain avec 200 insertions. Les *reads* sont générés en utilisant *ART*[50] avec les paramètres suivants : 2x150 paires de bases pour simuler un séquençage paired-end, à une couverture de 40 X, avec une taille d'insert de 500 paires de bases et un écart type de 20 paires de bases. L'ensemble des insertions sont simulées avec un génotype homozygote, qui représente le génotype le plus facile à détecter.

Un premier scénario de référence est simulé qui représente le contexte le plus simple pour les *variant callers*. Puis cinq scénarios sont simulés où un seul facteur est modifié dans le but

d'identifier l'impact de cette modification sur le rappel des *variant callers*.

Simulation du scénario de référence

La simulation dite de référence représente *a priori* le contexte le plus facile à détecter, où les séquences insérées contiennent très peu de répétitions et sont nouvelles dans le génome. Le contexte génomique de l'insertion est également simple et sans répétition, et sans homologie jonctionnelles. Pour ce faire, nous avons simulé des insertions *de novo* de 250 paires de bases situées dans des exons sans aucune homologie aux jonctions des points de cassure (Figure 4.1). Les insertions *de novo* ont été extraites de régions exoniques choisies au hasard dans le génome de *Saccharomyces cerevisiae*. La taille de 250 paires de bases est choisie pour deux raisons. La première est qu'elle représente une taille supérieure à celle d'un *read* et donc permet de tester l'étape d'assemblage des *variant callers*. La seconde est que cette taille était dans la fenêtre de taille des insertions les mieux identifiées dans les *callsets* de référence du chapitre précédent.

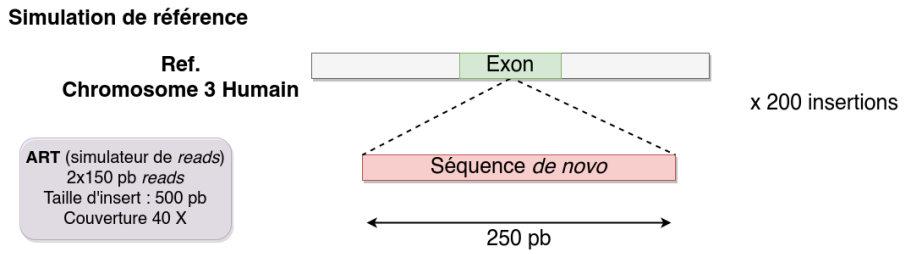

FIGURE 4.1 – Simulation de référence. 200 insertions *de novo* sont générées dans le chromosome 3 du génome humain Hg38.

Scénario 1 : variation de la taille de l'insertion

La localisation des insertions utilisées dans la simulation de base est conservée et les 200 séquences insérées sont remplacées par des séquences de 3 tailles différentes : 50, 500 et 1 000 paires de bases, extraites d'exons de *Saccharomyces cerevisiae* (Figure 4.2). Les différentes tailles sont testés de façon indépendante, un jeu de données est donc généré pour chaque taille.

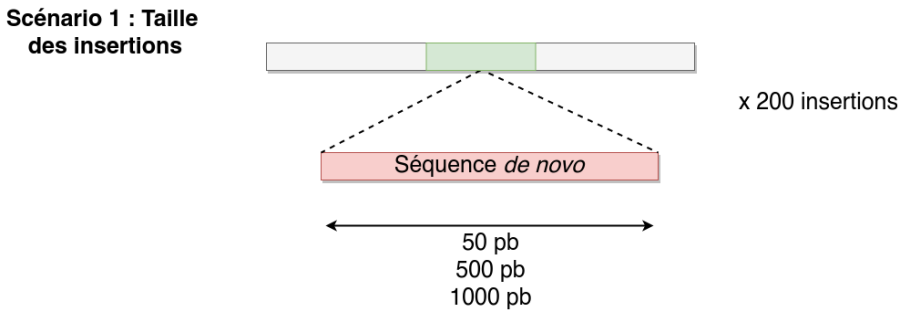

FIGURE 4.2 – Simulation de la variation de la taille des insertions. Chaque modalité de taille est réalisée à la même position que les insertions simulées dans le scénario de référence. Un jeu de données est réalisé pour chaque modalité de taille.

Scénario 2 : variation du type d'insertion

La localisation des insertions est identique à la simulation de référence, mais nous avons remplacé les séquences insérées de 250 paires de bases par des duplications dispersées, des répétitions en tandem, des duplications en tandem ou des éléments mobiles (Figure 4.3). Deux types de répétitions en tandem sont simulées, avec une taille du motif de 6 ou de 25 paires de bases, le motif provenant de la jonction du point de cassure gauche. Pour la simulation des éléments mobiles, 200 séquences d'éléments mobiles Alu d'une taille comprise entre 200 et 300 paires de bases sont extraites au hasard du génome humain sur la base de l'annotation RepeatMasker. Des duplications en tandem sont générées en dupliquant la séquence à droite du point de cassure de 250 paires de bases. Les séquences insérées des duplications dispersées simulées sont extraites à l'identique d'autres exons du chromosome 3.

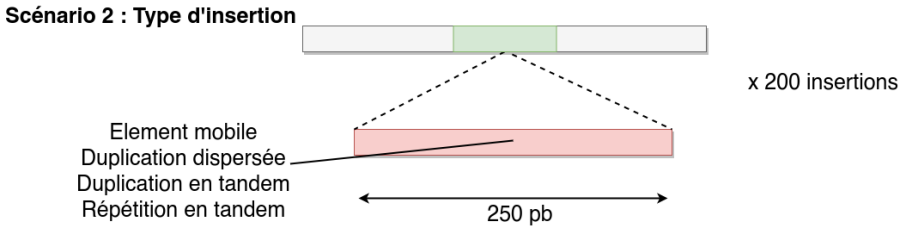

FIGURE 4.3 – Simulation de la variation du type d'insertion. Quatre types d'insertion sont simulés : éléments mobiles, duplications dispersées, duplications en tandem et deux variantes de répétition en tandem (taille de la graine 6 et 25 paires de bases).

Scénario 3 : variation de la taille de l'homologie jonctionnelle

Dans ce scénario, nous avons modifié les séquences d'insertion de 250 paires de bases de la simulation de référence pour obtenir une homologie jonctionnelle d'une taille donnée. Pour simuler les homologies jonctionnelles, nous avons remplacé les X premières bases de chaque insertion par une séquence de même taille provenant du point de cassure de droite. Nous avons simulé cinq tailles d'homologie jonctionnelle (valeur de X) : 10, 20, 50, 100 et 150 paires de bases (Figure 4.4).

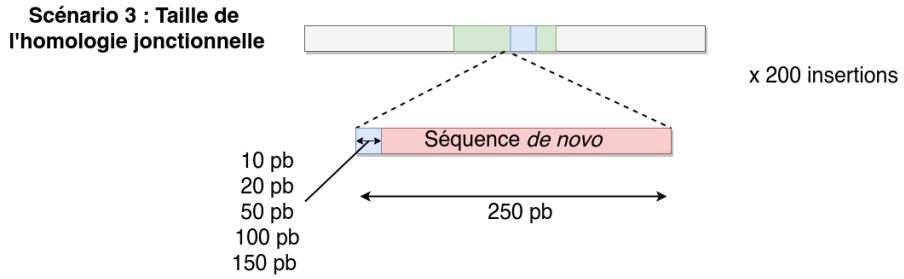

FIGURE 4.4 – Simulation de différentes tailles d'homologies jonctionnelles. Cinq tailles d'homologies jonctionnelles sont simulées : 10, 20, 50, 100 et 150 paires de bases.

Scénario 4 : variation du contexte génomique de l'insertion

Nous avons inséré les insertions de 250 paires de bases de la simulation de référence dans des contextes génomiques spécifiques : soit à l'intérieur de différents types d'éléments mobiles, à savoir les SINEs et les LINEs, dans de petites (<300 paires de bases) et de grandes (>300 paires de bases) répétitions simples ou dans d'autres régions non annotées par RepeatMasker (non répétées) (Figure 4.5). Un jeu de donnée additionnel avec des insertions proches est simulé en ajoutant des insertions proches des insertions simulées dans le scénario de référence. La distance entre les insertions est choisie uniformément entre 5 et 150 paires de bases.

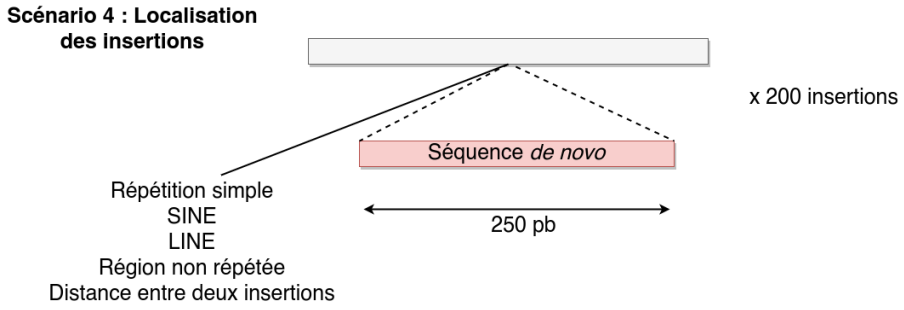

FIGURE 4.5 – Simulation d'insertion dans différents contextes génomiques.

Scénario 5 : Insertions réelles

Les 889 insertions situées sur le chromosome 3 du *callset* de NA19240 de Chaisson et al. ont été utilisées pour simuler trois ensembles de données supplémentaires[17]. De insertions *de novo* sont d'abord simulées aux localisations réels du chromosome 3, puis les insertions réelles sont simulées à l'intérieur des régions exoniques du chromosome 3. Enfin, les 889 insertions sont simulées comme décrites dans le fichier au format vcf.

4.1.2 Variant calling et méthodes d'évaluation

Pour chaque jeu de données simulées, nous avons aligné les *reads* avec *bwa* sur le chromosome 3 du génome de référence hg38. Les *reads* dupliqués sont marqués avec *samblaster* v.0.1.24 et convertis en fichier BAM avec *samtools* v1.6 [38, 78]. L'index BAM et le dictionnaire de référence sont obtenus via *Picard tools* v2.18.2.

Nous avons évalué quatre *variant callers* sur l'ensemble des données simulées. Nous les avons choisis en fonction de leurs bonnes performances dans des évaluations récentes et pour maximiser la diversité méthodologique[65]. L'ensemble de ces *variant callers* propose également un méthode d'assemblage permettant, en théorie, d'obtenir la séquence insérée. *GRIDSS*[16] , *Manta*[20] et *svABA*[136] sont des *variant callers* génériques basés sur une première étape d'alignement de *reads* sur un génome de référence. *MindTheGap*[110] est évalué car il correspond à un *variant caller* spécialisé dans la détection d'insertions et utilise une approche par assemblage sans étape d'alignement de *reads*.

GRIDSS v2.8.0, *Manta* v1.6.0, *MindTheGap* v2.2.1 et *svABA* v1.1.0 sont exécutés en utilisant les paramètres recommandés, ou autrement par défaut. Seules les insertions "PASS", qui sont supérieures à 50 paires de bases, sont conservées pour le calcul du rappel. Deux types de rappels sont calculés en fonction de la précision et des informations fournies pour chaque insertion identifée : un rappel du site d'insertion uniquement et un rappel "séquence résolue" où la séquence doit être correctement assemblée au site d'insertion simulée.

Le rappel sur le site d'insertion seulement est évalué uniquement sur la base de la prévision de la localisation du site d'insertion avec une marge de 10 paires de bases autour de la position simulée. Dans le cadre d'une évaluation plus rigoureuse, le rappel avec identification de la séquence nécessite d'identifier à la fois le site d'insertion à plus ou moins 10 paires de bases, mais également de rapporter la séquence de l'insertion. Lorsqu'elle est signalée, la séquence insérée doit partager au moins 90% de l'identité avec la séquence simulée et doit avoir une taille similaire de +/- 10%, pour être considérée comme un vrai positif. En cas d'absence de

séquence dans le fichier sous format vcf, si l'annotation fournie de l'événement permet d'extraire la séquence d'insertion du génome de référence (par exemple pour une duplication dispersée avec les coordonnées de la copie dupliquée), l'insertion est extraite du génome de référence et est évaluée. Le rappel est calculé comme le rapport entre le nombre de découvertes vrais positifs et le nombre d'insertions simulées (voir chapitre précédent).

4.1.3 Simulation de reads longs et variant calling

Pour chaque ensemble de données simulées avec des *reads* courts, un ensemble de données simulées avec des *reads longs* PacBio correspondant est généré. Cette simulation est réalisée en utilisant Simlord à une couverture de 40 X avec des probabilités de délétion, d'insertion et de substitution égales à 11%, 4% et 1% respectivement[125]. Les *reads* sont alignés avec Minimap2, les alignements sont triés avec samtools et les variants sont identifés avec *Sniffles*[75, 114]. *Sniffles* est selectionné car il représente le variant caller, utilisant des *reads longs*, le plus démocratisé à ce jour. L'évaluation du rappel du site d'insertion a suivi le même processus que pour l'évaluation des *variant callers* utilisant des reads courts. Pour le rappel sur la résolution de la séquence, deux seuils d'identité de séquence, 90 et 80%, sont utilisés pour valider les séquences insérées.

4.2 Résultats

4.2.1 Facteurs impactant la détection des insertions

Identification du site d'insertion

Les rappels sur les sites d'insertion pour les quatre méthodes sont présentés pour les différents jeux de données simulées dans la Table 4.1. Dans la simulation de référence, tous les outils réussissent à détecter 100% des insertions simulées, à l'exception de *GRIDSS* avec 81% de rappel. Ce plus faible rappel s'explique par l'annotation de 19% des insertions comme de faibles qualités par *GRIDSS*.

La taille de la séquence insérée montre un impact sur le rappel des sites d'insertion pour la plupart des outils, à l'exception de MindTheGap. *GRIDSS* présente des difficultés avec de petites insertions (50 paires de bases) alors que *Manta* et *svABA* ont plus de problèmes avec les grandes insertions. Le comportement le plus extrême est observé pour *svABA*, qui ne trouve aucun site d'insertion avec des insertions d'une taille supérieure à 500 paires de bases.

		Rappel sur le site d'insertion seulement (%)			
		GRIDSS	**Manta**	**SvABA**	**MindTheGap**
Simulation de référence : insertions *de novo* de 250 pb dans des exons		83	100	100	94
Scenario 1 Tailles de l'insertion	50 pb	56	100	100	95
	500 pb	100	86	0	93
	1,000 pb	100	88	0	94
Scenario 2 Type de l'insertion	Duplication dispersée	100	1	100	95
	Duplication en tandem	100	100	100	0
	Element mobile	100	2	100	56
	Répétition en tandem (graine : 6 pb)	100	90	0	0
	Répétition en tandem (graine : 25 pb)	99	66	1	2
Scenario 3 Homologie jonctionnelle	10 bp	100	100	96	0
	20 pb	100	100	85	0
	50 pb	77	68	12	0
	100 pb	100	22	49	0
	150 pb	100	0	100	0
Scenario 4 Localisation de l'insertion	Non répétées	83	100	94	97
	Répétitions simples (<300 pb)	82	100	100	63
	Répétitions simples (>300 bp)	87	94	95	55
	SINE	90	100	99	51
	LINE	80	100	97	85
	Insertions proches (<150 pb)	85	85	2	72
Scenario 5 Insertions réelles	Insertions *de novo*, positions réelles	84	80	71	35
	Insertions réelles, régions exoniques	84	74	57	9
	Insertions réelles, positions réelles	39	35	44	6

TABLE 4.1 – Rappel sur le site d'insertion de plusieurs *variant callers* avec des reads courts lors de différents scénarios simulés. Les cellules de la table sont colorées en fonction de la variation de la valeur de rappel de l'outil donné par rapport au rappel obtenu avec la simulation de référence (première ligne, colorée en bleu). Les cellules en rouge montrent une perte de rappel >10%, les cellules en vert montrent un gain de rappel >10% et les cellules en gris ne montrent aucune différence par rapport au rappel de base à +/- 10%.

Sur les jeux de données du scénario 2, concernant l'impact des types d'insertion sur le rappel, *GRIDSS* s'est montré comme le seul outil dont le rappel n'est pas affecté négativement. *Manta* n'a identifié aucun type de duplications dispersées et montre un rappel plus faible pour détecter les répétitions en tandem avec des graines de taille 25 paires de bases. MindTheGap s'est montré incapable de détecter des insertions de type duplications en tandem et n'a trouvé que 56% des insertions d'éléments mobiles. *svABA* n'a pas pu identifier des insertions avec des répétitions en tandem mais a pu détecter toutes les duplications dispersées, en tandem et les éléments mobiles.

En ce qui concerne l'homologie jonctionnelle, les outils montrent des comportements contrastés. *GRIDSS* est le seul outil non affecté par la présence et la taille de la séquence répétée aux jonctions d'insertion. Au contraire, MindTheGap est le plus touché par l'homologie jonction-

nelle, et se montre incapable de détecter les insertions avec des homologies de n'importe quelle taille testée. Le rappel de *Manta* diminue avec la taille des homologies jonctionnelles. Alors que *svABA* arrive à identifier des insertions avec des homologies jonctionnelles, de petite (<de 20 paires de bases) ou de très grande (150 paires de bases) taille, mais est affecté par des tailles moyennes.

En ce qui concerne l'impact du contexte génomique des insertions, aucune perte de rappel n'est observée dans les régions non répétées en comparaison avec le rappel de référence. Les outils génériques ne montrent aucune ou peu de différence du rappel dans les petites répétitions simples (<300 paires de bases), les SINE et les LINE. Les rappels de *Manta* et de *svABA* diminuent de 5 à 6% dans les zones de répétitions simples plus grandes que la taille de l'insert (>300 paires de bases). MindTheGap perd 39 et 43% de rappel lorsque les insertions se situent dans de grandes répétitions simples et dans des SINE. La simulation d'insertions proches les unes des autres sur le génome, à moins de 150 paires de bases, induit une réduction du rappel de *svABA* (-98%), MindTheGap (-22%) et *Manta* (-15%). Les insertions non identifiées par MindTheGap présentent notamment une proximité inférieure à 31 paires de bases. Cette taille correspond notamment à la taille du k-mer (mot de taille k), utilisé par MindTheGap pour construire le graphe de De Bruijn.

Enfin, lors de la simulation des insertions réelles (scénario 5), le rappel de tous les outils n'excède pas 44%, atteignant pour de nombreux outils leurs valeurs la plus faible parmi les différents ensembles de données simulées. Cela est particulièrement marqué pour *GRIDSS* dont le rappel est supérieur à 77% dans tous les scénarios simulés, mais n'atteint que 39% dans cette simulation. Lorsque nous relâchons un facteur de complexité tel que le type ou la localisation ; c'est-à-dire lorsque nous simulons soit des insertions *de novo* aux localisations réels, soit les types réels dans les régions exoniques, la baisse de rappel est beaucoup plus faible pour tous les outils. Cette observation révèle qu'il y a un effet synergique du type de l'insertion et de sa localisation réduisant fortement la capacité à détecter des insertions.

Qualité des insertions détectées

Les résultats précédents sont calculés en utilisant uniquement les variants détectés avec une qualité suffisante par chaque outil et annotés comme PASS dans le champ FILTER du fichier sous format vcf. La suppression de ce filtrage de la qualité permet d'augmenter le rappel principalement pour *GRIDSS* et *svABA* (Table 4.2). *GRIDSS* atteint un taux de rappel de 100% dans presque tous les scénarios, sauf dans le scénario simulant les insertions réelles, où une perte de rappel de 61% est observée. Ces différences indiquent qu'un nombre important de

sites d'insertions sont détectées mais identifiées comme de faible qualité.

		Rappel sur le site d'insertion seulement (%)			
		GRIDSS	**Manta**	**SvABA**	**MindTheGap**
Simulation de référence : insertions *de novo* de 250 pb dans des exons		100	100	100	94
Scenario 1 Tailles de l'insertion	50 pb	100	100	100	95
	500 pb	100	86	6	93
	1,000 pb	100	88	1	94
Scenario 2 Type de l'insertion	Duplication dispersée	100	49	100	95
	Duplication en tandem	100	100	100	0
	Element mobile	100	50	100	56
	Répétition en tandem (graine :6 pb)	100	92	22	0
	Répétition en tandem (graine : 25 pb)	100	66	100	1
Scenario 3 Homologie jonctionnelle	10 pb	100	100	98	0
	20 pb	100	100	89	0
	50 pb	100	51	65	0
	100 pb	100	12	100	0
	150 pb	100	0	100	0
Scenario 4 Localisation de l'insertion	Non répétées	100	100	98	97
	Répétition simple (<300 bp)	100	100	100	63
	Répétition simple (>300 bp)	99	94	100	55
	SINE	100	100	100	51
	LINE	100	100	100	85
	Distance entre insertions <150 bp	75	73	2	72
Scenario 5 Insertions réelles	Insertions *de novo*, positions réelles	84	80	71	35
	Insertions réelles, régions exoniques	84	74	57	9
	Insertions réelles, positions réelles	39	35	44	6

TABLE 4.2 – Rappel sur le site d'insertion de plusieurs *variant callers* lors de différents scénarios simulés sans filtre sur la qualité des insertions. Les cellules de la table sont colorées en fonction de la variation de la valeur de rappel de l'outil donné par rapport au rappel obtenu avec la simulation de référence (première ligne, colorée en bleu). Les cellules en rouge montrent une perte de rappel >10%, les cellules en vert montrent un gain de rappel >10% et les cellules en gris ne montrent aucune différence par rapport au rappel de base à +/- 10%.

Identification de la séquence des insertions

Les résultats précédents portent sur la capacité des outils a détecter le site des insertions simulées. Nous avons ensuite cherché à savoir si les *variant callers* testés étaient également capables de récupérer les séquences complètes insérées dans les différents scénarios de simulation (voir Table 4.3). Sur la simulation de référence, avec des insertions *de novo* de 250 paires de bases, tous les outils sont capables d'identifier la séquence pour l'ensemble des insertions identifiées.

		Rappel sur la séquence résolue (%)			
		GRIDSS	**Manta**	**SvABA**	**MindTheGap**
Simulation de référence : insertions *de novo* de 250 pb dans des exons		81	100	96	94
Scenario 1 Taille de l'insertion	50 pb	56	100	100	95
	500 pb	0	0	0	93
	1,000 pb	0	0	0	94
Scenario 2 Type de l'insertion	Duplication dispersée	0	0	16	95
	Duplication en tandem	0	0	0	0
	Element mobile	0	0	61	56
	Répétition en tandem (graine : 6 pb)	0	0	1	0
	Répétition en tandem (graine : 25 pb)	0	0	0	1
Scenario 3 Homologie jonctionnelle	10 pb	99	100	92	0
	20 pb	100	100	78	0
	50 pb	6	46	10	0
	100 pb	0	11	0	0
	150 pb	0	0	0	0
Scenario 4 Localisation de l'insertion	Non répétées	77	97	93	97
	Répétitions simples (<300 pb)	77	98	97	63
	Répétitions simples (>300 pb)	77	93	90	55
	SINE	77	99	94	51
	LINE	76	97	95	85
	Insertions proches (<150 pb)	75	73	2	72
Scenario 5 Insertions réelles	Insertions *de novo*, positions réelles	64	73	67	35
	Insertions réelles, régions exoniques	11	14	14	9
	Insertions réelles, positions réelles	6	23	30	6

TABLE 4.3 – Rappel sur la résolution de séquences des insertions détectées par les *variant callers* testés selon différents scénarios de simulation. Les cellules de la table sont colorées en fonction de la variation de la valeur de rappel de l'outil donné par rapport au rappel obtenu avec la simulation de référence (première ligne, colorée en bleu). Les cellules en rouge montrent une perte de rappel >10%, les cellules en verte montrent un gain de rappel >10% et les cellules en gris ne montrent aucune différence par rapport au rappel de référence à +/- 10%.

Cependant, ce rappel chute lorsque le scénario se complexifie. Bien que la découverte des sites d'insertion n'ait pas été très influencée par la taille de l'insertion, tous les outils sauf *MindTheGap* présentent une incapacité à identifier les séquences insérées lorsque la taille est supérieure à 500 paires de bases. *MindTheGap* assemble correctement presque toutes les séquences simulées, même celles de 1 kilobase. En ce qui concerne les autres types d'insertion, les *variant callers* échouent à identifier la séquence correctement, à l'exception de *MindTheGap* et de *svABA* pour certaines duplications dispersées et des insertions d'éléments mobiles. L'augmentation de la taille de l'homologie jonctionnelle a réduit la résolution des séquences de *GRIDSS* et *svABA*. *GRIDSS* ne parvient plus à rapporter la séquence lorsque l'homologie jonctionnelle atteint une taille supérieure à 20 paires de bases. Les insertions situées dans des régions répétées sont moins résolues que dans la simulation de référence pour tous les outils.

Ce facteur reste le facteur impactant le moins la capacité des *variant callers* à identifier la séquence de l'insertion. Enfin, la résolution de la séquence des insertions réelles simulées à leur localisation réel diminue par rapport au rappel du site d'insertion, *GRIDSS* subissant la plus grande perte (-33%).

4.2.2 Quantités variables de faux positifs

Les outils ayant fait l'objet des plus forts rappels sont également ceux qui ont produit le plus grand nombre de faux positifs (de l'ordre de plusieurs centaines pour *GRIDSS* et *svABA*, voir Table 4.4). Plus surprenant encore, la quantité de faux positifs n'est pas constante pour la plupart des outils entre les différents scénarios de simulation.

	Quantité de faux positif							
	GRIDSS		**Manta**		**SvABA**		**MindTheGap**	
	PASS	All	PASS	All	PASS	All	PASS	All
Simulation de référence	0	151	2	2	6	84	2	2
Scenario 1 : Taille de l'insertion	0	131 - 138	0 - 3	0 - 3	0 - 6	82 - 96	0 - 2	0 - 2
Scenario 2 : Type de l'insertion	3 - 400	233 - 591	0 - 18	0 - 201	4 - 451	92 - 1,157	1 - 2	1 - 2
Scenario 3 : Homologie jonctionnelle	2 - 9	128 - 163	0 - 4	0- 4	5 - 202	70 - 342	0 - 1	0 - 1
Scenario 4 : Localisation de l'insertion	0 - 4	143 - 166	0 - 5	0 - 5	4 - 13	74 - 643	0 - 4	0 - 4
Scenario 5 : Insertions réelles	382	2,052	101	148	523	9,314	3	3

TABLE 4.4 – Quantités de faux positifs détectés par les *variant callers* testés selon différents scénarios de simulation. Pour chaque scénario impliquant plusieurs ensembles de données simulées, les valeurs indiquent le nombre minimal et maximal de prédictions de faux positifs obtenues sur ces ensembles de données. Les cellules de la table sont colorées en fonction de la variation de la quantité de faux positif de l'outil donné par rapport à la quantité obtenue avec la simulation de référence (première ligne, colorée en bleu). Les cellules en rouge montrent une augmentation substantielle de la quantité de faux positifs, les cellules en gris montrent une petite différence ou une diminution de la quantité de faux positif par rapport à la simulation de référence.

Il augmente lorsque les insertions simulées présentent un schéma de duplication (élément mobile, duplication dispersée et homologies jonctionnelles supérieures à 50 paires de bases). La suppression du filtre de qualité entraîne une forte augmentation du nombre de faux positifs pour *GRIDSS* et *svABA* (5 à 17 fois plus respectivement).

4.2.3 Union et intersection des variant callers

Une stratégie classique pour améliorer la détection de variants sur des données réelles consiste à concilier plusieurs *callsets* de *variant callers* en ne conservant que les variants qui sont identifiés de la même manière par différents *variant callers*. Dans le dernier scénario de simulation, seuls 12% des sites d'insertion ont été validés par les trois outils, *GRIDSS*, *Manta* et *svABA*, et 39% par au moins deux outils. Cependant, l'union des trois méthodes permet de retrouver 65% des sites d'insertion réels, ce qui représente une augmentation de 20% du meilleur rappel obtenu par une seule méthode (Figure 4.6).

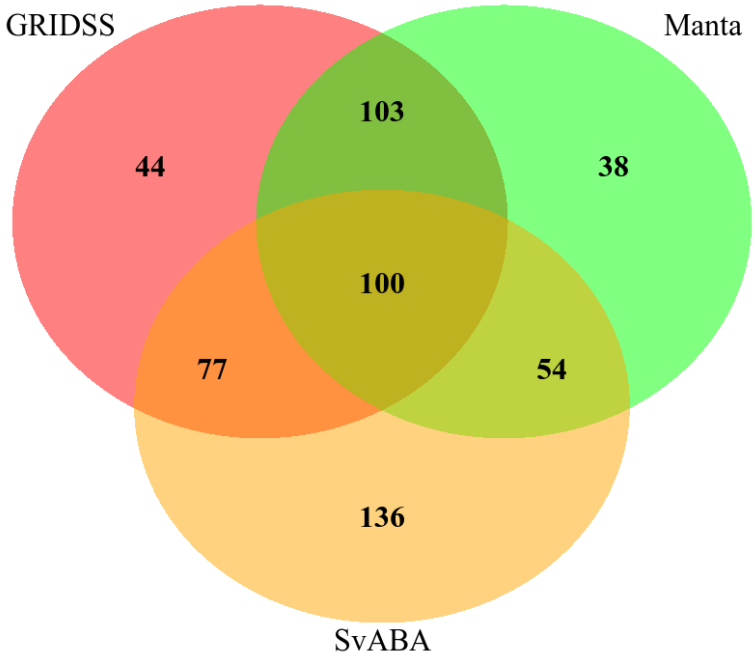

FIGURE 4.6 – Intersections des vrais positifs identifiés entre les différents *variant callers* testés sur données simulées. Intersections des vrais positifs identifiés entre *GRIDSS*, *svABA* et *Manta* dans la simulation du scénario 5 (insertions réelles à des endroits réels). Dans ce scénario, les 889 insertions situées sur le chromosome 3 de l'ensemble d'appels NA19240 sont simulées comme décrit dans le fichier au format vcf. Les vrais positifs représentent aux variants identifiés lors de l'évaluation basée sur le site d'insertion.

4.2.4 Évaluation avec des données longs reads simulées

Dans le but d'évaluer si les facteurs de difficulté précédemment identifiés pour les données reads courts ont également un impact sur le rappel avec des stratégies reads longs, nous avons également simulé un ensemble de données PacBio et utilisé le *variant caller Sniffles*[114]. Pour

la plupart des scénarios d'insertion, *Sniffles* rapporte avec précision 100 % des sites d'insertion, sauf pour le type de duplication en tandem et pour les insertions avec de grandes homologies jonctionnelles (rappel inférieur à 20%).

		Rappel : site d'insertion (%)		**Rappel : séquence résolue (%)**	
		reads courts GRIDSS	**reads longs Sniffles**	**reads courts GRIDSS**	**reads longs Sniffles**
Simulation de référence : 250 bp novel seq. in exons		83	100	81	27 (100)
Scenario 1 Taille de l'insertion	50 pb	56	100	56	33 (100)
	500 pb	100	100	0	19 (100)
	1,000 pb	100	100	0	15 (100)
Scenario 2 Type de l'insertion	Duplication dispersée	100	100	0	20 (100)
	Duplication en tandem	100	15	0	0 (8)
	Element mobile	100	100	0	23 (100)
	Répétition en tandem (graine : 6 pb)	100	100	0	14 (100)
	Répétition en tandem (graine : 25 pb)	99	95	0	10 (95)
Scenario 3 Homologie jonctionnelle	10 pb	100	100	99	9 (100)
	20 pb	100	91	100	5 (90)
	50 pb	77	47	6	2 (42)
	100 pb	100	24	0	0 (11)
	150 pb	100	11	0	0 (6)
Scenario 4 Localisation de l'insertion	Non répétées	83	100	77	22 (100)
	Répétitions simples (<300 pb)	82	100	77	25 (100)
	Répétition simples (>300 pb)	87	100	77	19 (100)
	SINE	90	100	77	21 (100)
	LINE	80	100	76	25 (100)
	Insertions proches(<150 bp)	85	54	75	8 (45)
Scenario 5 Insertions réelles	Insertions *de novo*, positions réelles	84	58	64	15 (46)
	Insertions réelles, régions exoniques	84	98	11	5 (21)
	Insertions réelles, positions réelles	39	58	6	7 (49)

TABLE 4.5 – Comparaison du rappel sur le site d'insertion et sur la séquence résolue entre *variant callers* basés sur des reads courts et des longs reads. Les cellules de la table sont colorées en fonction de la variation de la valeur de rappel de l'outil donné par rapport au rappel obtenu avec la simulation de base (première ligne, colorée en bleu). Les cellules en rouge montrent une perte de rappel >10%, les cellules en gris ne montrent aucune différence par rapport au rappel de base à +/- 10%. Les rappels résolus par séquence ont été calculés avec un seuil d'identité de séquence de 90%, sauf pour les nombres entre parenthèses pour lesquels le seuil a été abaissé à 80%.

Dans ces cas, les insertions sont effectivement signalées mais à plus de 10 paires de bases du site d'insertion simulé. Cela est probablement dû à une résolution de séquence imprécise empêchant la normalisation correcte à gauche des positions des points de cassure. Un autre facteur de difficulté était la proximité des points d'insertion, pour laquelle *Sniffles* rapporte un événement complexe au lieu de plusieurs insertions proches. Cela explique principalement le

faible taux de rappel de 58% pour le jeu de données avec les insertions réelles du chromosome 3 à leurs positions réelles. En ce qui concerne la résolution de la séquence à 90% d'identité, bien que les insertions identifiées par *Sniffles* contiennent systématiquement une séquence complète insérée, cette dernière est imprécise et contient des erreurs de séquençage conduisant à des rappels autour de 20% seulement. En abaissant le seuil d'identité à 80%, le rappel sur la séquence rapportée est identique au rappel du site d'insertion pour la plupart des scénarios d'insertion. Cela montre que la présence d'erreurs de séquençage conduit à une identification de séquence de moins bonne qualité.

4.3 Discussion

4.3.1 Apport des simulations

Les simulations restent une approche puissante pour identifier les forces et les faiblesses des *variant callers*, mais ne reflétent en aucun cas la complexité des données réelles. Dans nos simulations, plusieurs paramètres nous éloignent de la complexité réelle du reséquençage du génome humain, comme certains biais technologiques de séquençage, l'utilisation d'un chromosome au lieu du génome entier, et l'absence d'autres polymorphismes que les variants d'insertion (SNP, petits indel et autres variants de structure). En conséquence, les rappels signalés dans cette étude sont susceptibles d'être des sur-estimations de ceux obtenus avec des données réelles. Bien que les valeurs absolues doivent être interprétées avec prudence, elles peuvent facilement être comparées entre les *variant callers* et entre les scénarios de simulation.

De fortes différences dans les rappels entre outils et scénarios ont pu être observées, ce qui permet de fournir des informations intéressantes concernant les facteurs d'impact et le comportement des *variant callers*. Notre protocole de simulation a permis d'étudier chaque facteur de difficulté indépendamment et a mis en évidence l'impact plus important du type d'insertion par rapport à la localisation génomique de l'insertion. Cependant, l'ensemble des facteurs étudiés pris indépendamment ne peuvent pas expliquer la perte totale de rappel lors de la simulation des insertions réelles à leur localisation réelle. Un effet synergique important est suspecté où plusieurs des facteurs étudiés sont présents en un seul événement d'insertion. Par exemple, la découverte de séquences *de novo* dans des régions répétées ou à de réelles positions est réalisée avec succés pour une majorité d'outils testés. Cependant, lors de la simulation de véritables séquences insérées à leur réelles localisations, nous pouvons observer une réduction de moitié du rappel des *variant callers*, en comparaison avec la simulation de séquence *de novo*

à ces même positions.

Nos simulations ont révélé que des homologies jonctionnelles aussi petites que 20 paires de bases ont un impact sur le rappel de tous les outils testés. De telles séquences répétées sont susceptibles d'altérer les informations d'alignement utilisées par les *variant callers*. Bien que ces caractéristiques des points de cassure des variants et leurs relations avec les mécanismes moléculaires générant les variants aient été décrites depuis longtemps, elles semblent rarement prises en compte dans la conception des algorithmes des *variant callers*. A notre connaissance, seul *GRIDSS* présente une méthode pour détecter ce type d'évènement. Dans le chapitre précédent, nous avons pu montrer que de telles tailles d'homologies jonctionnelles sont relativement courantes, avec près de 40% des insertions ayant des homologies jonctionnelles supérieures à 10 paires de bases. Par conséquent, les algorithmes des *variant callers* gagneraient à prendre en compte ces propriétés des points de cassure, qui sont susceptibles de générer des signaux très spécifiques en termes d'alignement de reads.

4.3.2 Résolution de séquence

Un résultat frappant de nos simulations est l'absence de résolution de séquence pour la plupart des caractéristiques d'insertion simulées et pour la plupart des *variant callers* testés. Outre la perte évidente d'informations sur l'événement de l'insertion, elle limite également l'identification du type d'insertion, le génotypage et la validation de l'insertion prédite. Nous avons observé que la plupart des insertions, quelles que soient leur type et leur contexte génomique d'insertion, étaient détectables mais souvent non signalées avec une qualité suffisante en raison de ce manque de résolution. De plus, la résolution des séquences est une étape essentielle pour la comparaison et le génotypage des variants chez de nombreux individus. Comme ces tâches sont à la base des études d'association et du diagnostic médical, les efforts devraient être orientés vers une meilleure résolution de la séquence de ces variants [86, 19]. Les résultats obtenus avec l'outil d'assemblage local *MindTheGap* ont montré que l'utilisation de l'ensemble des données de séquençage permet d'assembler de nombreuses insertions et même des de grandes tailles. La restriction à un petit sous-ensemble de lecture pour effectuer l'assemblage local peut donc être le défaut des *variant callers* testés (*GRIDSS*, *Manta*, *svABA*). L'assemblage imparfait de *Manta*, qui n'arrive à assembler que le début et la fin de l'insertion, semble aller dans ce sens. La résolution de la séquence insérée est possible dans une certaine mesure, mais les répétitions en tandem plus importantes que la taille des *reads* resteront difficiles à résoudre avec la technologie des lectures courtes. Il reste intéressant de remarquer que *GRIDSS* est capable de rapporter des répétitions en tandem dont la taille maximale correspond à la taille des reads simulés. Ainsi,

même si la séquence est incomplète, l'utilisation de notre méthode d'annotation permettrait d'identifier la présence de répétition en tandem.

Il est intéressant de noter que la résolution des séquences semble également être un problème avec des *variant callers* utilisant des *reads longs*. *Sniffles* rapporte des séquences insérées complètes, mais avec une faible précision de séquence, en raison du taux d'erreur de séquençage plus élevé. Ce problème limite la normalisation à gauche des sites d'insertion, ce qui conduit à des erreurs de localisation des insertions. Cette faible précision des variants prédits est susceptible d'entraver le génotypage et la comparaison de variants entre individus. La correction de reads longs pourrait résoudre ce problème mais est peu conseillée car cette dernière peut conduire à la correction de variants identifiés comme des erreurs de séquençage. Nos résultats montrent qu'il est également nécessaire d'améliorer les *variant callers* basés sur des reads longs.

Dans cette étude, nous nous sommes principalement concentrés sur le rappel, mais nous avons également calculé le nombre de faux positifs. Nous avons choisi de comparer les quantités observées de faux positifs prédits par les outils, plutôt que d'utiliser la précision, car cette dernière dépend de la quantité de vrai positif. Il est intéressant de noter que les quantités de faux positifs sont affectées par le type d'insertions simulées. Les quantités les plus importantes ont été observées pour les insertions dupliquées, pour lesquelles certains *variant callers* ont prédit des insertions non seulement aux sites d'insertion mais aussi aux localisations des copies homologues des séquences insérées. Ces informations pourraient être très utiles pour les filtrer ou pour éviter l'utilisation de *variant callers* dans des situations spécifiques.

4.3.3 Amélioration de l'évaluation des variant callers

Dans l'ensemble, les différents *variant callers* n'ont pas obtenu de bons résultats dans toutes les situations et dans tous les aspects de la détéction d'insertions. Chaque *variant caller* a montré ses propres forces et faiblesses, souvent différentes entre outils. L'identification précise de ces derniers en termes de caractéristiques des insertions et du contexte génomique permettra d'utiliser chaque outil au mieux. Pour ce faire, les études de référence doivent tenir compte de la grande variabilité des caractéristiques des insertions que ce travail a mis en évidence.

Deux évaluations récentes de *variant callers* ont permis de prendre conscience de la variabilité des performances des *variant callers* en fonction des ensembles de données et des approches[65, 15]. Cependant, ces derniers facteurs ont été analysés pour tous les types de *variant callers* combinés et aucune de ces études n'a pris en compte les différents types d'insertions.

Des meilleures pratiques, pour l'analyse comparative des indel, ont été suggérées sur la base de *callsets* de référence dans des régions à haut niveau de confiance, laissant les variations

structurelles de côté[66]. Cependant, c'est précisément ce type de variation qui nécessite des recommandations pour une amélioration des pratiques à utiliser et une normalisation de l'annotation, car elles sont plus difficiles à identifier et à signaler. Nous espérons que la caractérisation fine actuelle des variants au sein des individus de référence aidera à développer de meilleures pratiques pour l'évaluation comparative des *variant callers*.

4.3.4 Pistes d'améliorations des outils

Des conseils pour améliorer la détection en utilisant la technologie de reads courts ont déjà été décrits, comme la combinaison des *variant callers* complémentaires[65]. Les *meta variant callers* tels que Meta-sv[95], Parliament2[146] ou sv-callers[67] réconcilient les variants identifiés par différents *variant callers*. Cependant, seuls les variants qui sont découverts de manière concordante entre les différents outils sont rapportés. Cette stratégie permet d'augmenter la précision au détriment du rappel. Nos simulations ont montré que l'intersection de seulement trois *variant callers* réduisait le rappel de 30%, alors que leur union augmentait le rappel d'au moins 20%. Cependant, cette stratégie d'union nécessite un contrôle minutieux des faux positifs. Un meilleur contrôle pourrait probablement être obtenu avec des *variant callers* rapportant les séquences d'insertions et en écartant les types d'insertion générant le plus de faux positifs. Ainsi il serait possible de réaliser une union uniquement sur des types d'insertions dont les outils sont capables de détecter.

Une autre solution, moins décrite, pourrait être l'utilisation d'outils dédiés à chaque type d'insertion, au lieu de n'utiliser que des *variant callers* génériques. Parmi ceux-ci, Expansion Hunter a été conçu pour détecter les répétitions en tandem, Pamir et Popins pour les insertions *de novo* et TARDIS pour les grandes duplications[32, 60, 61, 120].

Les résultats de la comparaison présentés ici fournissent déjà quelques suggestions concrètes pour améliorer les *variant callers*. Premièrement, la détection du site d'insertion pourrait être améliorée en prenant en compte les signaux d'alignement atypiques générés par les grandes homologies jonctionnelles. Ensuite, le rappel de la résolution de la séquence pourrait être amélioré en utilisant l'ensemble des lectures au lieu de sous-ensembles recrutés pour l'assemblage de la séquence insérée. Notre protocole de simulation nous a également permis d'identifier les complémentarités entre les différents *variant callers* et a montré que le rappel d'insertion pouvait être amélioré en prenant l'union des variants détectés. Enfin, sur la base de ces complémentarités et avec une meilleure résolution de la séquence, des sélections plus intelligentes que la simple union des variants détectés, tenant compte du type d'insertion, de la taille et du contexte, pourraient être conçues pour atteindre un rappel élevé tout en contrôlant le taux de faux positifs.

De telles améliorations sont cruciales pour la généralisation de la génomique des populations et des études d'association à des variants autres que ponctuelles. Ces améliorations pourraient permettre le développement de la médecine personnalisée et aider à un meilleur diagnostic médical. Le chapitre suivant propose un exemple d'amélioration possible pour l'outil *MindTheGap*. Nous verrons comment, en modifiant quelques paramètres et fonctionnalités, il est possible d'améliorer la détection d'insertion.

Chapitre 5

Amélioration du variant caller MindTheGap

Dans le chapitre précédent nous avons pu identifier les limites de différents *variant callers* basés sur l'utilisation de *reads courts*. Dans ce chapitre nous nous concentrons sur un *variant caller* en particulier : *MindTheGap*. L'outil a été développé et publié par Rizk et al. en 2014[110]. Plusieurs raisons nous ont motivé à améliorer *MindTheGap*. Le premier est que cet outil a été développé par l'équipe Genscale, équipe dans laquelle est réalisée cette thèse. Ce fait favorise l'amélioration de l'outil grâce à l'interaction facilitée avec les développeurs à l'origine de l'outil. La seconde raison est que l'outil montre à la fois des forces dans l'assemblage d'insertions mais également des faiblesses dans sa détection, identifiées au chapitre précédent. Ses faiblesses ouvrent des voies d'améliorations possibles de l'outil. Nous verrons, dans un premier temps, le fonctionnement en profondeur de *MindTheGap*. Nous identifierons et appliquerons les améliorations possible pour *MindTheGap* dans le cadre d'une utilisation dans un contexte médical.

5.1 Fonctionnement détaillé de MindTheGap

MindTheGap est un outil de détection d'insertions, basé sur l'utilisation d'un graphe de De Bruijn pour représenter le génome de l'individu séquencé (Figure 5.1). La première étape, appelée *Find*, consiste à la recherche des points de cassure qui sont associés à des insertions homozygotes et hétérozygotes. Pour cela les k-mers du génome de référence sont comparés aux k-mers présents dans le graphe. La seconde étape, appelée *Fill*, consiste en un assemblage des insertions associées aux points de cassure détectés durant la première étape. Pour cela, le graphe est exploré en recherchant les chemins qui permettent de raccorder les k-mers de part et d'autre des points de cassure.

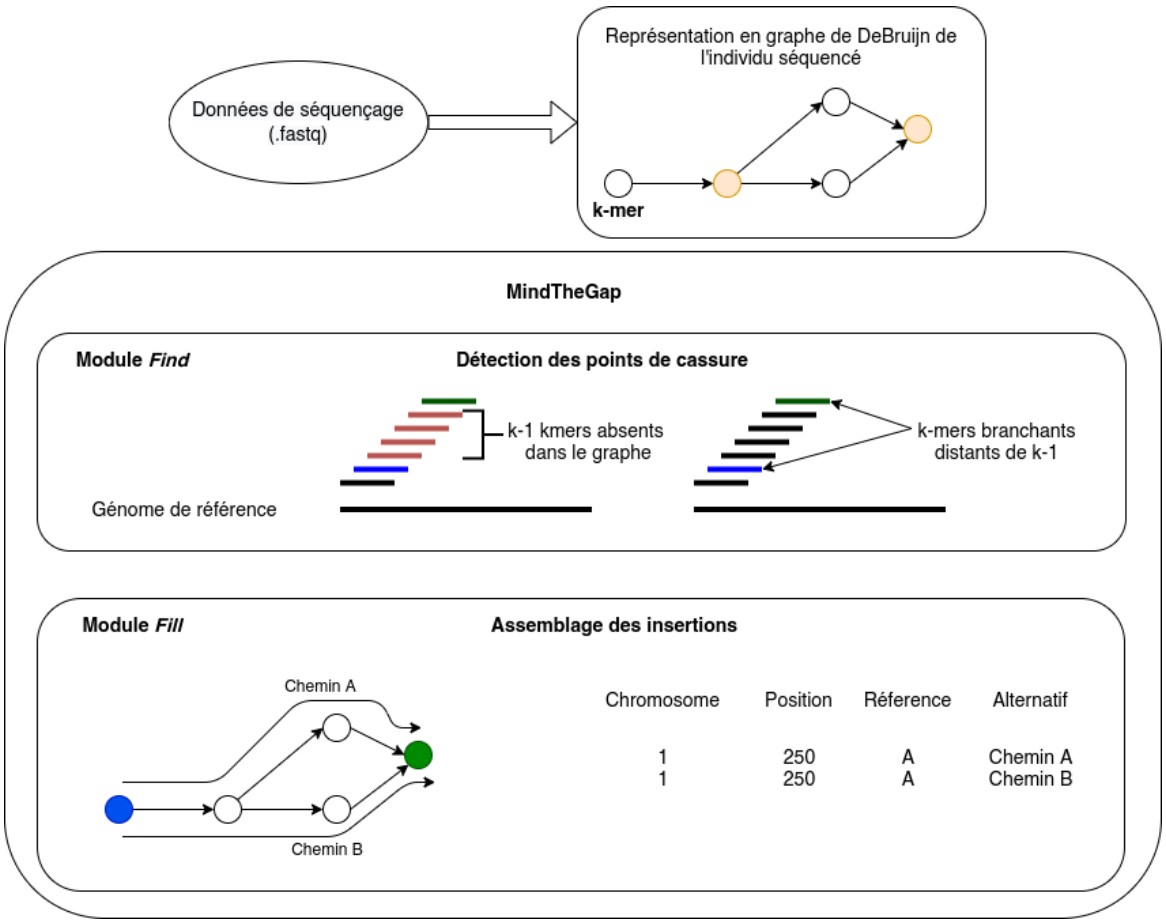

FIGURE 5.1 – Fonctionnement global de MindTheGap. Les données de séquençage sont représentées sous la forme d'un graphe de De Bruin. Un noeud dans ce graphe correspond à un k-mer, mot de taille k trouvé dans les données de séquençage. Les noeuds oranges représentent des k-mers qui possèdent plus de un k-mer branchant entrant ou sortant. Le module *Find* de MindTheGap identifie des points de cassure associés à des insertions homozygotes et hétérozygotes. Le module *Fill* utilise les points de cassure pour explorer le graphe afin d'assembler les séquences insérées.

5.1.1 Des données de séquençage au graphe de De Bruijn

La première opération que réalise l'outil *MindTheGap* est la conversion des *reads courts* du génome séquencé sous la forme d'un graphe de De Bruijn. Le graphe de De Bruijn est un graphe orienté dans lequel les noeuds sont des k-mers, des mots de taille k, et une arête relie un noeud A à un noeud B si le k-1 suffixe de A est exactement le k-1 préfixe de B (voir Chapitre 2). Cette structure de donnees permet donc de representer l'ensemble des k-mers contenus dans le genome sequence. Cependant, les technologies de séquençage de seconde génération

produisent des erreurs dans les *reads* qui sont aléatoires et avec un taux d'occurrences de 0.01%. L'intégration brute des k-mers présents dans les *reads* conduit donc à l'incorporation de k-mers contenant des erreurs et n'apparaissant pas dans le génome. Dans le but d'éliminer les k-mers associés à des potentielles erreurs de séquençage, un comptage des occurrences des k-mers est réalisé. L'amplification qui précède le séquençage permet à une même position du génome d'être couverte par plusieurs *reads*. Nous pouvons donc supposer que les k-mers issus de ces erreurs de séquençage présentent un nombre d'occurrences plus faible que les k-mers associès à des *reads* dépourvus d'erreurs. Les k-mers dont le nombre d'occurrences est inférieur à un seuil fixé ne sont pas conservés dans le graphe et ceux dont le nombre d'occurrences est supérieur au seuil fixé sont gardés et appelés k-mers solides.

L'implémentation est réalisée grâce à la *librairie GATB* qui se présente comme l'une des implémentations les plus lègères en mémoire pour travailler avec des graphes de De Bruijn[33]. La performance en mémoire est obtenue grâce à l'utilisation d'un filtre de Bloom pour stocker l'ensemble des noeuds du graphe, les k-mers solides, et par le fait que le graphe et ses arêtes ne sont pas explicitement représentés. Pour parcourir le graphe à partir d'un noeud donné, il suffit d'interroger la présence de ses quatres voisins possibles dans la structure d'indexation des k-mers. Cette structure très compacte est composée notamment d'un filtre de Bloom. Ce filtre est un tableau de *bits* associé à plusieurs fonctions de hachage qui permet de stocker et requêter la présence ou l'absence d'un ensemble d'éléments. Cette structure est également probabiliste qui peut renvoyer un faible taux de faux positifs, c'est-à-dire indiquer qu'un k-mer est présent alors qu'il ne l'est pas. L'implémentation du graphe de De Bruijn par la librairie GATB ajoute un second filtre qui est une table de faux positifs dits critiques. cette table permet de rendre le parcours du graphe exact, c'est-à-dire sans faux positifs. A titre d'exemple, seulement 6 Go de mémoire vive sont nécessaires pour représenter et parcourir le graphe de De Bruijn d'un génome humain séquencé à 47X[23].

5.1.2 Détection des points de cassure : module Find

Le module *Find* scanne les k-mers présents le long du génome de référence et recherche leur présence ou absence dans le graphe de De Bruijn du génome séquencé. Des variations génétiques par rapport au génome de référence induisent des motifs particuliers de présence, absence ou caractéristiques des k-mers du graphe. Deux motifs sont recherchés par *MindTheGap*, l'un pour les insertions homozygotes et le second pour les insertions hétérozygotes. Les insertions homozygotes sont caractérisées par l'absence contiguë (*gap*) de k-1 k-mers de la référence dans le graphe (Figure 5.2 A). Les insertions hétérozygotes sont caractérisées non pas par un *gap* de

k-1, mais par la présence de k-mers branchants à une distance de k-1 (Figure 5.2 B). On définit un k-mer branchant comme un noeud ayant un degré entrant ou un degré sortant supérieur ou égal à 2 (le degré entrant/sortant d'un noeud étant le nombre d'arêtes entrantes/sortantes du noeud). De ce fait, une insertion hétérozygote génére un premier k-mer avec un degré sortant de degré 2 et un second k-mer avec un degré entrant de degré 2, k-1 positions plus loin (Figure 5.2 C). Ce degré 2 est important puisqu'il atteste la présence de deux chemins possible suggérant une hétérozygotie. Cependant, ce motif n'est pas spécifique aux insertions hétérozygotes. Des régions répétées peuvent conduire à l'apparition d'un tel motif. Pour limiter cette situation, il est vérifié que les k-mers branchants ne sont pas répétés dans le génome de référence. Il est important de noter que ces deux motifs sont induits par des insertions quelque soit leur taille.

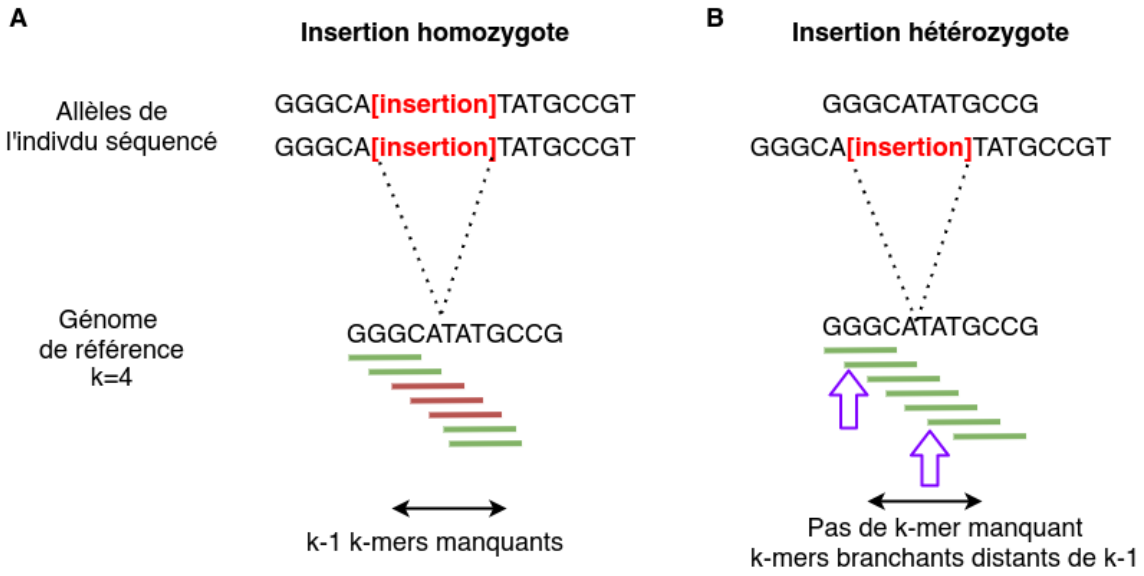

FIGURE 5.2 – Signatures induites par les insertions homozygotes et hétérozygotes dans un graphe de De Bruin comparé à un génome de référence. Figure adaptée de [110]. A : une insertion homozygote conduit à l'absence dans le graphe de k-1 k-mers consécutifs du génome de référence (ligne rouge), B : une insertion hétérozygote ne présente pas de k-mer du génome de référence absent dans le graphe, l'insertion hétérozygote est reconnaissable par la présence de deux k-mers branchants (flèche violette) à une distance de k-1 dans le génome de référence.

La détection de ces motifs conduit à la récupération des k-mers aux abords du *gap* ou des k-mers branchants dans le graphe. Ces k-mers sont écrits dans un fichier FASTA, où le premier k-mer est identifié comme la graine gauche et le second comme la graine droite du point de cassure.

La présence d'homologies jonctionnelles exactes au niveau des insertions altèrent le signal

utilisé pour détecter les insertions. Pour les insertions homozygotes, les homologies jonctionnelles réduisent la taille du *gap* (Figure 5.3 A).

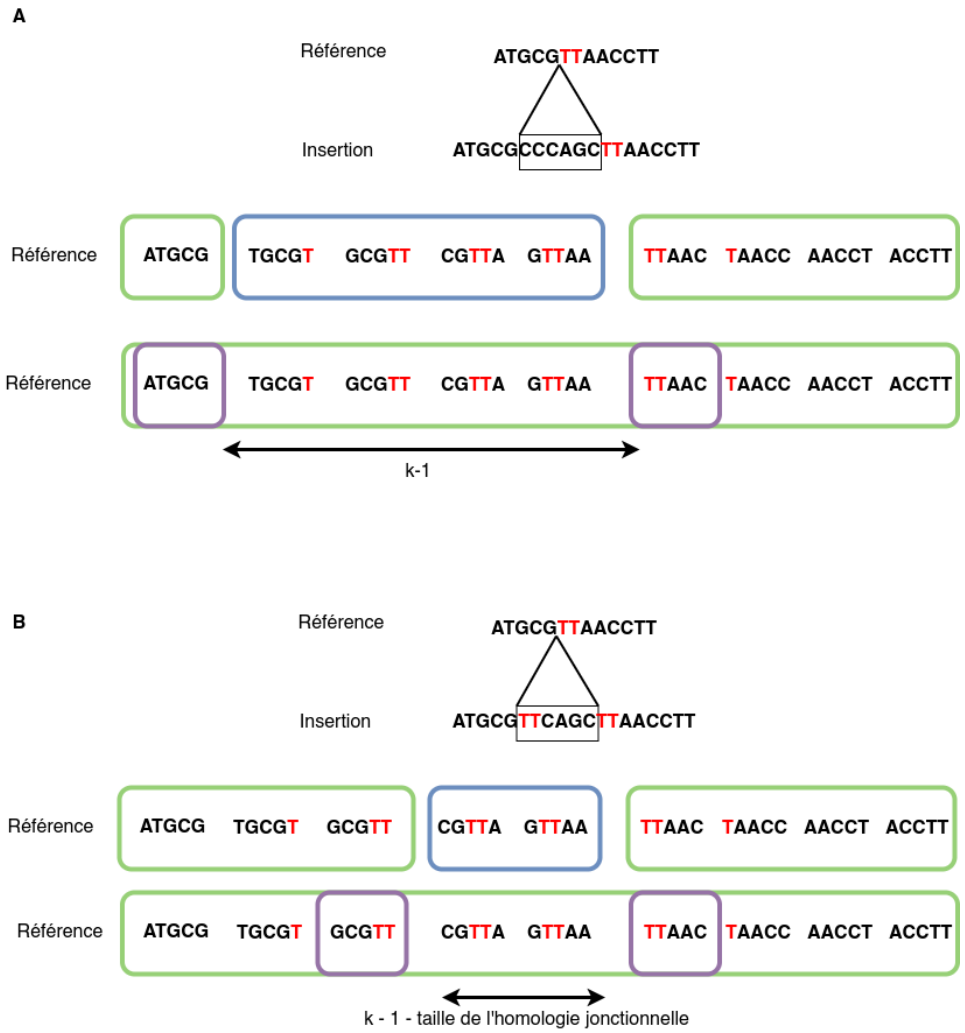

FIGURE 5.3 – Impact des homologies jonctionnelles sur les signature des insertions homozygotes et hétérozygotes dans MindTheGap. Schéma comparant l'impact d'une homologie jonctionnelle de taille 2 (TT, en rouge) sur les motifs des insertions homozygotes et hétérozygotes. Les k-mers du génome de référence absents dans le graphe de De Bruijn sont encadrés en bleu, ceux présents en vert et ceux qui sont branchants en violet. A : motifs attendus pour des insertions homozygotes et hétérozygotes sans homologies jonctionnelles. B : altération des motifs décrit en A par la présence d'une homologie jonctionnelle de taille 2.

Le motif recherché n'est donc plus un *gap* de k-1 k-mers mais un intervalle de k-1-n à k-1 k-mers absents, où n est la taille maximale de l'homologie jonctionnelle. Pour les insertions hétérozygotes, l'homologie jonctionnelle impacte la distance entre les deux k-mers branchants

(Figure 5.3 B). La distance recherchée n'est plus de k-1 mais un intervalle k-1-n à k-1, où n est la taille de l'homologie jonctionnelle. Par défaut *MindTheGap* recherche des insertions qui peuvent avoir au maximum une homologie jonctionnelle de taille 5. La paramètre associé noté *max-repeat* peut être modifié par l'utilisateur.

Suite à sa publication, *MindTheGap* a été amélioré pour détecter également des SNP et des délétions homozygotes. En effet, leur proximité à moins de k-1 d'une insertion peuvent empêcher la détection de l'insertion (Figure 5.4). Cette proximité induit une augmentation de la taille du *gap* pour les insertions homozygotes, qui n'est plus de k-1 mais de k + la distance entre le SNP et l'insertion (Figure 5.4 A).

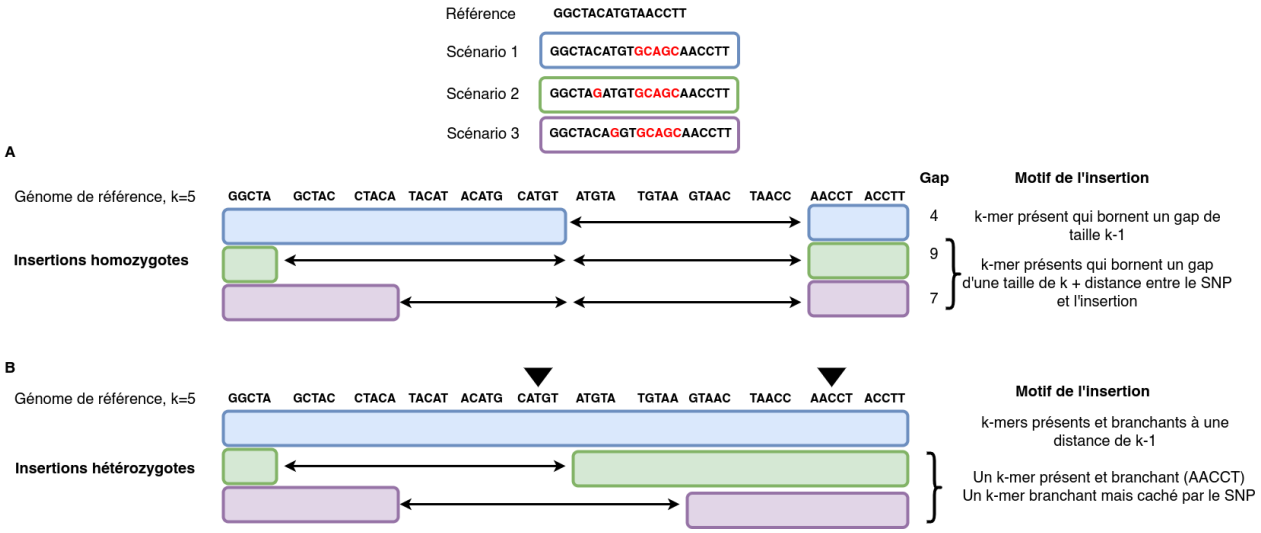

FIGURE 5.4 – Impact de la présence d'un SNP à proximité d'une insertion pour la détection d'insertions par MindTheGap. Trois cas de figures sont représentés, appelés scénario 1 (en bleu), 2 (en vert) et 3 (en violet). Le scénario 1 présente le cas d'une insertion isolée. Le scénario 2 présente le cas d'une insertion avec un SNP homozygote à une distance de 4 paires de bases. Le scénario 3 présente le cas d'une insertion avec un SNP homozygote à une distance de 2 paires de bases. Les SNP et insertions sont écrits en rouge. Les rectangles de couleurs indiquent la présence des k-mers de la référence dans le graphe. Les triangles noirs indiquent la position des k-mers branchants associés à l'insertion hétérozygote. A : Modification du motif causée par la présence de SNP dans la détection d'insertions homozygotes. B : Modification du motif causée par la présence de SNP dans la détection d'insertions hétérozygotes.

La présence de SNP ou de délétions homozygotes proche d'insertions hétérozygotes a également un impact sur la détection. Ces derniers masquent un des k-mers branchants nécessaires pour identifier le point de cassure. Ce k-mer branchant dans le graphe contient l'allèle alternatif du SNP qui n'est pas présent dans le génome de référence (Figure 5.4 B). Il devient donc impos-

sible d'identifier le point de cassure avec l'algorithme de détection d'insertions hétérozygotes. Les améliorations apportées à *MindTheGap* permettent désormais de détecter des insertions avec un SNP homozygote à une distance inférieure à k-1. Pour cela, *MindTheGap* détecte dans un premier temps le SNP, puis le nucléotide du génome de référence associé au SNP est altéré par l'allèle alternatif. La région modifiée du génome de référence est de nouveau analysée pour détecter les insertions homozygotes et hétérozygotes.

Pour détecter les SNP homozygotes, l'algorithme recherche des *gaps* de taille supérieure ou égale à k. Pour vérifier l'existence d'un SNP, le k-mer précédent le *gap* est tronqué de son dernier nucléotide. Ce k-mer tronqué se voit ajouter un des trois autres nucélotides. Ce k-mer altéré est ensuite recherché dans le graphe. S'il est présent dans le graphe, les k-mers suivants sont également altérés avec le nouveau nucléotide afin de vérifier leur présence dans le graphe. S'ils sont présents, il est notifié que ce point de cassure est associé à un SNP et le SNP est écrit dans un fichier au format vcf.

La détection de délétions est également faite uniquement sur des délétions homozygotes. Une délétion est suspectée lorsqu'un *gap* d'une taille supérieure à k-1 est identifiée. La délétion est vérifiée en concaténant les k-mers présents qui bornent le *gap*. Les k-mers de cette concaténation sont ensuite recherchés dans le graphe pour vérifier leur présence dans le graphe. S'ils sont présents, la délétion est validée et écrite dans un fichier au format vcf.

5.1.3 Identification des séquences insérées : module Fill

A l'issue du module *Find* un ensemble de points de cassure, caractérisés par un k-mer en amont et un k-mer en aval de l'insertion, est obtenu. Le module *Fill* a pour objectif d'assembler les insertions grâce à une exploration du graphe de De Bruijn. Le principe est d'utiliser les k-mers associés à chaque point de cassure, appelés graines, pour borner cette exploration.

Dans son implémentation, la connexion au graphe est réalisée à partir de la graine de gauche. Le graphe de De Bruijn qui est exploré est simplifié en un graphe de *contigs*. Pour cela une première exploration est réalisée dans le but d'assembler des *contigs*. Cette exploration est basée sur un algorithme de parcours en largeur (*Breadth First Search*, Figure 5.5). Ces *contigs* sont des séquences représentatives d'un morceau du graphe pour lesquelles une étape d'écrasement de polymorphisme a été réalisé. Plus simplement, lors de l'exploration du graphe, il peut être observé des noeuds avec des branchements qui ouvrent vers plusieurs chemins et des noeuds avec des branchements qui ferment ces multiples chemins. La structure comprise entre ces deux noeuds branchants sont appelés des bulles. La construction du graphe de *contigs* a pour objectif d'écraser les petites bulles, afin de réduire le nombre de solutions possibles durant l'assemblage.

Des limites d'explorations sont appliquées lors de la construction de ce graphe afin de ne pas construire un graphe de *contigs* de l'ensemble du graphe de De Bruijn.

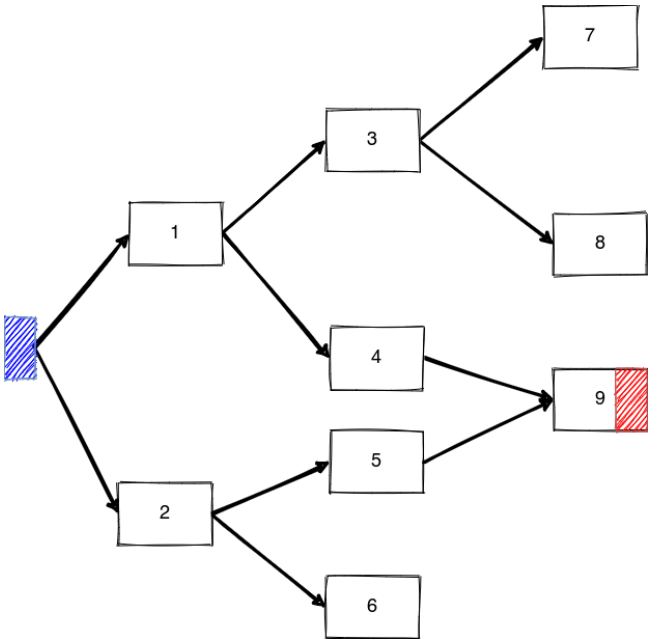

FIGURE 5.5 – Algorithme de parcours pour l'exploration du graphe de contigs. La connexion au graphe est réalisé à partir de la graine gauche du point de cassure (en bleu). L'exploration du graphe est réalisé par un parcours du graphe en largeur (*Breadth First Search*). Chaque rectangle représente un *contig* et les numéros associés donne l'ordre dans lequel le graphe est exploré. Les conditions d'arrêts d'exploration sont : la découverte de la graine droite du point de cassure (en rouge), l'exploration de 100 *contigs*, l'exploration de 10 000 nucléotides. Les solution renvoyée par *MindTheGap* sont la séquence des *contigs* 1, 4, 9 et la séquence des *contigs* 2,5 et 9. La graine droite est retirée de la séquence du *contig* 9 dans l'insertion assemblée.

La première limitation est la découverte du k-mer exact du point de cassure de droite, tandis que la seconde et la troisième correspondent à des limites d'exploration du graphe. Cette limitation fixe un nombre maximal de *contigs* produits (100 par défaut) et de nucléotides explorés (10 000 par défaut). Lorsque l'ensemble des *contigs* est construit, une recherche du k-mer approché du point de cassure de droite dans les *contigs* est réalisée. Si une graine approchée est trouvée, les chemins entre les *contigs* contenant les deux graines du point de cassure sont recherchés. Si la graine de droite n'a pas été trouvée, le graphe de *contigs* est construit dans le sens inverse, en commencant par la graine de droite pour aller vers la graine de gauche. *MindTheGap* renvoie l'ensemble des chemins possibles qui permettent de relier les deux graines de chaque point de cassure. Afin de réduire les chemins redondants, un alignement est réalisé entre les différentes solutions. Les solutions qui présentent une similarité supérieure à 90% sont

retirées et une version représentative est conservée.

5.1.4 Améliorations de l'utilisation de MindTheGap apportées durant la thèse

MindTheGap a été peu évalué sur des données réelles lors de sa publication en 2014. Une expérience a été réalisée sur des données humaines de l'individu NA12878 séquencé à plus de 90 X mais le but était de montrer que l'outil était capable d'assembler certaines insertions très longues >5 kb. Ainsi la performance du module Find n'a pas été évalué précisément sur données réelles. De plus, dans cette expérience où la profondeur de séquençage était très importante, *MindTheGap* n'avait pas été utilisé avec les paramètres par défaut. Depuis la publication, l'outil n'avait pas été appliqué sur des données réelles humaines. Il était donc nécessaire d'estimer l'utilisation potentielle de *MindTheGap* dans un contexte d'usage médical.

MindTheGap a nécessité de nombreuses améliorations afin de répondre aux besoins d'un usage dans un contexte médical. La première modification que nous avons apporté est le formatage et la normalisation à gauche des insertions assemblées dans un format vcf. Les insertions assemblées dans la version native de *MindTheGap* n'étaient pas présentés dans un fichier au format vcf, ce qui compliquait la comparaison avec d'autres outils. Nous avons développé une sortie au format vcf, en ajoutant également la normalisation à gauche des insertions. Cette normalisation compare les séquences assemblées avec les k-mers associés aux points de cassure. De ce fait, la recherche d'homologie pour la normalisation est faite de façon exacte, ne peut pas dépasser la taille des k-mers et est sensible au polymorphisme.

Comme nous l'avons vu, *MindTheGap* identifie l'ensemble des chemins qui permettent de relier deux k-mers. La seconde modification que nous avons apportée est l'implémentation d'une option qui permet de filtrer les insertions qui présentent un nombre supérieur à la ploïdie de l'homme. Nous avons estimé et constaté que ces insertions sont donc de moins bonne qualité que celles qui respectent cette contrainte. De ce fait, les insertions avec un nombre de solutions supérieures à 2 ont leur champs *QUAL* égale à *Low Qual*. Cette information permet à l'utilisateur de l'aider dans la priorisation des insertions à regarder.

Les premiers tests réalisés sur des données de séquençage de génome entier de l'individu NA12878 affichent une détection d'un très grand nombre de points de cassure (>250 000). Le module *Fill* n'a pas pu aboutir car le temps d'éxécution estimé dépassait 36 jours. Ces premiers résultats ont révélé que MindTheGap dans son état actuel n'était pas capable d'analyser des données de séquençage humain. Dans l'objectif de réduire le nombre d'insertions à assembler

dans le module *Fill*, qui sont principalement des petites insertions (voir Introduction), nous avons décidé d'implémenter un micro assemblage dans le module *Find*. Ce micro assemblage se focalise sur les petites insertions d'une taille de 1 à 2 paires de base. Le fonctionnement est très similaire à celui utilisé pour la détection de délétions. Les deux graines du point de cassure sont concaténées avec l'ajout tour à tour des différentes insertions de taille 1 à 2 paires de base possibles entre ces graines. Les k-mers de la séquence concaténée sont recherchés dans le graphe. Si l'ensemble des k-mers sont présents, l'insertion est écrite dans le *callset* avec les SNP et les autres délétions détectées. Ces points de cassure ne seront donc pas envoyés au module *Fill*. Cette implémentation a permis de résoudre 24% des points de cassure qui auraient nécessité une étape d'assemblage beaucoup plus longue dans le module *Fill*.

Nous avons observé que la détection de SNP par *MindTheGap* n'était pas parfaite. De plus, L'outil découvrait un certain nombre de points de cassure localisés aux positions de SNP dans des simulations de SNP dans un génome entier. La principale cause de ces points de cassure faux positifs était la taille du *gap* produit par ces SNP. Nous avons observé que la taille des *gap* était en général inférieure ou égale à k-1, correspondant à un motif d'une insertion homozygote. Dans le contexte médical, la détection de SNP est généralement réalisée, en amont de la détection de variants de structure, par *GATK*[89]. Dans le but de réduire ces points de cassure faux postifis associés à des SNP, nous avons utilisé une approche d'altération du génome pour masquer ces SNP. Nous modifions le génome de référence en altérant les nucléotides par les SNP détectés comme homozygotes par *GATK*. *MindTheGap* est ensuite utilisé avec le génome altéré qui contient les allèles alternatifs des SNP homozygotes du génome de l'individu et les données de séquençage du génome de l'individu. Cette imputation des SNP a ainsi réduit le nombre de points de cassure de 12%, majoritairement homozygotes. La combinaison de l'imputation des SNP et du micro assemblage des petites insertions ont permis une réduction du nombre de points de cassure de 31%.

5.2 Limites de MindTheGap

5.2.1 Retour sur l'évaluation de MindTheGap

Dans le chapitre précédent, nous avons évalué *MindTheGap* dans différents scénarios de simulations d'insertions. Dans cette partie nous revenons sur ces résultats dans le but de comprendre l'origine des limites de *MindTheGap*. Comme nous l'avons vu dans la partie dédiée au fonctionnement de *MindTheGap*, le rappel peut être limité dans deux situations. La première

situation est que l'outil ne détecte pas le point de cassure dans le module *Find*. La seconde situation est celle où *MindTheGap* détecte le point de cassure mais ne parvient pas à assembler l'insertion. Nous présentons dans la Table 5.1, les résultats détaillés des deux modules.

Le scénario de référence indique que *MindTheGap* identifie 94% des points de cassure associés aux insertions simulée. L'ensemble de ces points de cassure sont également correctement assemblés avec le module *Fill*. MindTheGap produit les mêmes résultats pour le scénario 1, modifiant la taille des insertions simulées, que ceux observés dans le scénario de référence. L'évaluation sur les différents types d'insertions présente des résultats plus contrastés. L'analyse des résultats du module *Find* pour ce scénario indique que *MindTheGap* trouve plus de 90% des points de cassure simulés. Une perte de rappel suite à l'assemblage des points de cassure est observé pour les duplication en tandem (-100%), les éléments mobiles (-41%) et les répétitions en tandem (-91% et -92%). L'étape d'assemblage de ces types d'insertion est donc l'étape qui limite *MindTheGap*. L'évaluation sur les homologies jonctionnelles révèle que les points de cassure ne sont pas identifiés, ce qui ne permet pas l'assemblage des insertions simulées. La limite de la détection se situe donc dans la détection des points de cassure dans le module *Find*. La localisation de l'insertion impacte le module *Find* dans sa capacité à détecter des insertions mais n'impacte pas l'assemblage du module *Fill*.

MindTheGap présente un taux d'insertions assemblées faux positifs stable et très faible quelque soit le scénario simulé. Le module *Find* produit des faux positifs mais très peu conduisent à une insertion assemblée. Nous nous sommes interrogés si les différents scénarios généraient des points de cassure faux positifs qui ne sont pas assemblés dans le module *Fill*. En moyenne, 405 points de cassure faux positifs sont générés pour l'ensemble des scénarios (Table 5.1). La déviation standard est de 20, ce qui révèle que le nombre de points de cassure est sensiblement identique entre les différents scénarios. Il semble donc que les différents scénarios n'impactent pas le nombre de faux positifs détectés.

5.2.2 Passage à l'échelle de MindTheGap

Nos premières évaluations ont été réalisées sur le chromosome 3 du génome humain dans un objectif de simplifier les simulations. Nous nous intéressons maintenant au comportement de *MindTheGap* avec des données de séquençage de génome entier simulées. Le scénario de référence a été reproduit mais à l'échelle du génome entier. 200 insertions de 250 paires de bases ont été produites dans des régions exoniques de chaque autosome du génome Hg38, soit au total 4 304 insertions.

		Module Find		Module Fill	
		Rappel (%)	**Nombre de faux positifs**	**Rappel** (%)	**Nombre de faux positifs**
Simulation de référence : insertions *de novo* de 250 pb dans des exons		94	402	94	2
Scenario 1 Taille de l'insertion	50 pb	94	404	95	1
	500 pb	94	397	93	0
	1,000 pb	95	412	94	2
Scenario 2 Type de l'insertion	Duplication dispersée	98	424	95	1
	Duplication en tandem	97	410	0	1
	Element mobile	97	348	56	2
	Répétition en tandem (graine : 6 pb)	91	419	0	1
	Répétition en tandem (graine : 25 pb)	93	400	1	2
Scenario 3 Homologie jonctionnelle	10 pb	0	397	0	0
	20 pb	0	410	0	1
	50 pb	0	392	0	1
	100 pb	0	403	0	1
	150 pb	0	434	0	1
Scenario 4 Localisation de l'insertion	Non répétées	98	387	97	1
	Répétitions simples (<300 pb)	63	396	63	2
	Répétitions simples (>300 pb)	63	394	55	2
	SINE	53	406	51	2
	LINE	87	340	85	0
	Insertions proches (<150 pb)	74	412	72	4
Scenario 5 Insertions réelles	Insertion *de novo*, positions réelles	44	393	35	9
	Insertions réelles, régions exoniques	93	441	9	127
	Insertions réelles, positions réelles	18	443	6	4

TABLE 5.1 – Rappel de la détection des points de cassure et de l'assemblage de l'outil *MindTheGap* pour les différents scénarios de simulation d'insertions.

Echelle	**Module Find**				**Module Fill**		
	Nombre de points de cassure	Rappel (%)	Précision (%)	Temps (min)	Rappel (%)	Précision (%)	Temps (min)
Chromosome 3	187/402	94	32	8	94	99	7
Génome entier	3728/9942	81	27	140	NA	NA	> 4000

TABLE 5.2 – Impact du passage à l'échelle du génome entier sur le rappel, la précision et le temps d'éxécution de *MindTheGap*. La première valeur du nombre de points de cassure représente les points de cassure homozygotes et la seconde ceux qui sont hétérozygotes.

Le passage à une analyse de données de séquençage du génome entier induit une multiplication par 25 du nombre de faux positifs en comparaison à la simulation sur le chromosome 3 (Table 5.2). Des pertes de 7% du rappel et de 5% de la précision sont observées à l'issue du module *Find*. L'augmentation du nombre de points de cassure détectés induit un temps d'éxécution estimé à plus de 67 heures. Le nombre de faux positifs détectés dans le module *Find* représente donc un frein pour l'utilisation du module *Fill*.

5.3 Améliorations de MindTheGap

5.3.1 Résolution de l'impact des homologies jonctionnelles

Nous avons évalué l'impact de la modification du paramètre *max-repeat*, associé à la gestion des homologies jonctionnelles, sur les performances de *MindTheGap*. Huit tailles d'homologies jonctionnelles ont été simulées : 3, 5, 6, 10, 28, 29, 30 et 31 paires de bases. Deux valeurs du paramètre *max-repeat* sont testées : 5 et 30 paires de bases. L'augmentation du paramètre *max-repeat* permet de détecter des insertions qui présentent des homologies jonctionnelles inférieures à la valeur du paramètre (Table 5.3).

Homologie jonctionnelle (pb)	MindTheGap (max-repeat = 5 pb)		MindTheGap (max-repeat = 30 pb)	
	Rappel(%)	Précision(%)	Rappel(%)	Précision(%)
3	92	95	94	94
5	77	100	94	94
6	50	100	93	93
10	0	0	94	94
28	0	0	93	94
29	0	0	84	92
30	0	0	59	64
31	0	0	26	85

TABLE 5.3 – Rappel et précision de *MindTheGap* en fonction de la taille de l'homologie fonctionnelle simulée et de la valeur du paramètre *max-repeat*.

Par exemple, l'augmentation du paramètre *max-repeat* à 30 paires de bases permet d'identifier la quasi totalité des insertions simulées. Cependant, nous pouvons observer que le rappel de l'outil est impacté dès lors que l'homologie a une taille proche ou supérieure à la valeur du paramètre *max-repeat*. Lorsque les insertions possèdent une homologie jonctionnelle égale à 30 paires de bases, une perte d'un tier en rappel et en précision est observée. A 31 paires de bases, les 26 % de rappel correspondent à des insertions détectées dont le génotype est erroné. Les insertions sont détectées en tant qu'hétérozygote alors qu'elles sont simulées homozygotes. L'augmentation du paramètre *max-repeat* s'accompagne d'une détection 14 fois plus importante de faux positifs dans le module *Find*(Table 5.4). Cette augmentation du nombre de points de cassure conduit à une multiplication par 10 du temps d'éxécution du module *Fill*. Il est important de noter que la valeur du paramètre *max-repeat* possède une limite : sa valeur ne peut pas

être supérieure à la taille des k-mers. Il n'est donc pas possible d'identifier des insertions avec des homologies jonctionnelles de taille supérieures à k.

Homologie jonctionnelle (pb)	MindTheGap v2.2.2 (max-repeat = 5 pb)		MindTheGap v2.2.0 (max-repeat = 30 pb)	
	Homozygote	Hétérozygote	Homozygote	Hétérozygote
3	184	410	192	5747
5	155	387	191	5748
6	99	376	192	5724
10	0	399	193	5770
20	0	397	198	5699
28	0	376	188	5799
29	0	401	173	5731
30	0	409	117	5770
31	0	410	0	5963

TABLE 5.4 – Quantité de points de cassure détectés par le module *Find* selon la taille de l'homologie jonctionnelle simulée et de la valeur du paramètre *max-repeat*.

5.3.2 Origine des faux positifs

Nous avons pu observer que *MindTheGap* détecte en moyenne 400 points de cassure faux positifs quelque soit le scénario simulé (Table 5.1). Ces faux positifs représentent un temps inutilement alloué à l'assemblage lors de l'éxécution du module *Fill*. La réduction du nombre de points de cassure faux positifs est devenu un problème majeur à résoudre suite à l'augmentation de ces derniers avec l'analyse de génome entier et l'augmentation du paramètre *max-repeat*.

Afin de limiter des assemblages inutiles de faux positifs, nous recherchons des caractéristiques qui peuvent être associées à ces points de cassure dans le but d'éviter leur détection dès le module *Find*. La première caractéristique que nous avons pu observer est que ces faux positifs sont principalement identifiés comme des insertions hétérozygotes. Cette caractéristique n'est pas suffisante pour discerner des insertions hétérozygotes vrais positifs des faux positifs. Néanmoins, cette information révèle que ces faux positifs sont détectés par la présence de k-mers branchants distant de k-1. Ces branchements dans le graphe peuvent être induits par la présence d'une hétérozygotie à un même locus ou par la présence de divergences entre copies d'une région répétée. Nous émettons l'hypothèse que ces faux positifs sont localisés dans des régions répétées. Dans le graphe, ces régions répétées se traduiraient par une présence anormale

de k-mer branchants. Nous avons donc analysé les degrés entrants et sortants des k-mers autour des points de cassure vrais et faux positifs du scénario de référence sur le chromosome 3 (Figure 5.6 et 5.7).

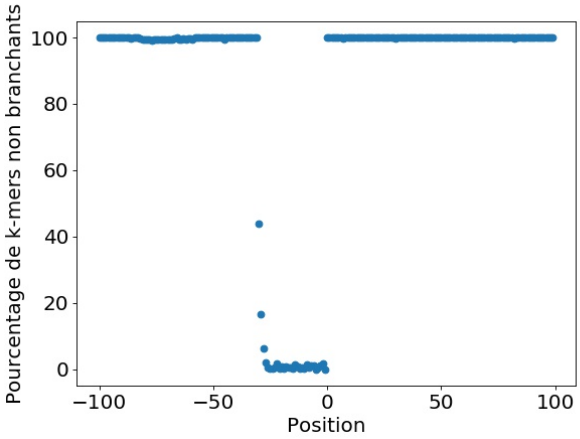

FIGURE 5.6 – Pourcentage des k-mers non branchants autour des points de cassure vrais positifs. L'axe des abscisses correspond à la position des k-mers autour du point de cassure. La position 0 correspond à la position du k-mer associé au point de cassure de gauche. L'axe des ordonnées correspond au pourcentage des k-mers non branchants parmi l'ensemble des vrais positifs.

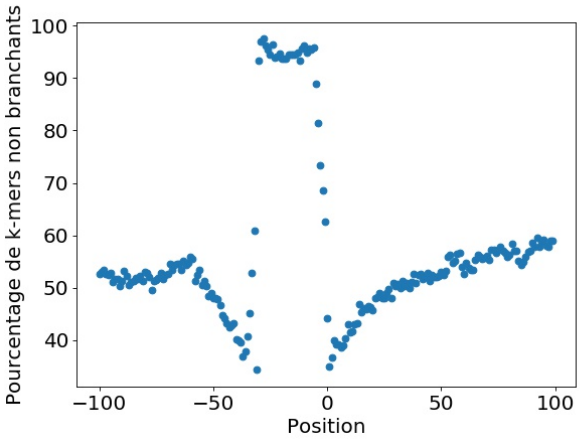

FIGURE 5.7 – Pourcentage de k-mers non branchants autour des points de cassure faux positifs. L'axe des abscisses correspond à la position des k-mers autour du point de cassure. La position 0 correspond à la position du k-mer associé au point de cassure de gauche. L'axe des ordonnées correspond au pourcentage des k-mers non branchants parmi l'ensemble des faux positifs.

Le pourcentage de k-mers non branchants autour des points de cassure vrais positifs est

homogène (Figure 5.6). Les k-mers du motif du point de cassure sont bien absents du graphe. Plus de 99% des k-mers environnants sont non branchants. A l'inverse, le nombre de k-mers non branchants dans les régions associées à des faux positifs est plus faible (Figure 5.7). Les k-mers du motif des points de cassure sont bien présents et ce motif est également bornée par des k-mers branchants. Ces deux conditions expliquent pourquoi ces points de cassure ont été identifiés comme des insertions hétérozygotes potentielles. Les k-mers autour de ces points de cassure présentent des pourcentages de k-mers non branchants plus faibles que ceux observés avec les vrais positif, oscillant entre 35 et 60%. Les régions associées aux points de cassure faux positifs comportent donc plus de k-mers branchants que dans les régions associées aux vrais positifs.

L'analyse de la complexité du graphe que nous avons observé s'est réduite au scénario de référence. Dans ce scénario, les insertions simulées sont localisées dans des exons, régions connues pour être peu complexes. Afin de savoir si ces observations sont reproductibles dans des régions plus complexes, nous avons étendu cette analyse sur les simulations du scénario 4, où différentes localisations d'insertions sont simulées.

Nous avons donc regardé la complexité du graphe autour des points de cassure vrais positifs dans ces différentes régions répétées (Figure 5.8). La pourcentage de k-mers non branchants autour des insertions dans les régions non réptées est semblable aux observations réalisées sur les vrais positifs dans le scénario de référence. On peut néanmoins noter que le pourcentage de k-mers non branchants présente une plus grande fluctuations dans les régions répétées telles que les SINE, LINE et répétitions simples. Cette fluctuation oscille entre 80 et 99% mais reste plus faible que celle observée avec les points de cassure faux positifs (Figure 5.7). Ce pourcentage de k-mers non branchants autour des points de cassure représente donc une piste pour réduire ces faux positifs.

5.3.3 Réduction des faux positifs

Suit aux résultats précédents, nous nous sommes intéressés à réduire le nombre de points de cassure faux positifs détectés dans le module *Find* en analysant la complexité du graphe autour des points de cassure. Nous avons implémenté un filtre dans le module *Find* qui quantifie le pourcentage de k-mers non branchants des 50 k-mers qui précédent chaque k-mer scanné. Les points de cassure dont ce pourcentage de branchements est inférieur à un seuil fixé ne sont pas conservés. Quatre seuils sont évalués : 60%, 70%, 80% et 90% dans le scénario de référence (Table 5.5).

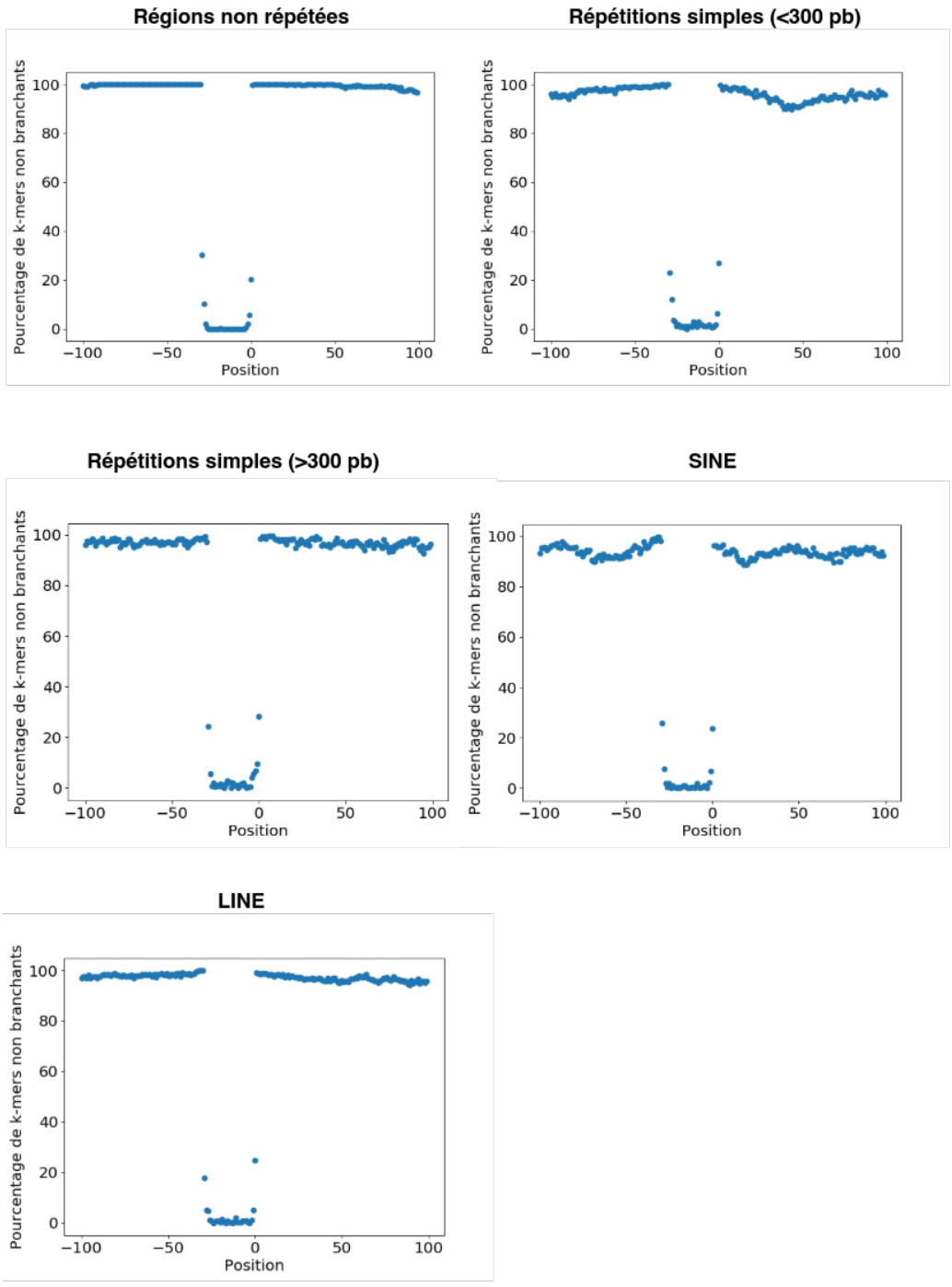

FIGURE 5.8 – Pourcentage des k-mers non branchants autour des points de cassure vrais positifs des simulations du scénario 4. L'axe des abscisses correspond à la position des k-mers autour du point de cassure. La position 0 correspond à la position du k-mer associé au point de cassure de gauche. L'axe des ordonnées correspond au pourcentage des k-mers non branchants parmi l'ensemble des vrais positifs.

Echelle	Seuil du filtre(%)	Module Find				Module Fill		
		Nb. de points de cassure	Rappel (%)	Précision (%)	Temps (min)	Rappel (%)	Précision (%)	Temps (min)
Chr. 3	0	187/402	94	32	8	94	99	7
	60	187/221	94	46	13	94	100	6
	70	187/152	94	55	13	94	100	4
	80	186/93	93	66	13	93	100	4
	90	186/32	93	85	13	93	100	2
Génome entier	0	3728/9942	87	27	140	NA	NA	> 4000
	90	3704/615	86	86	194	52	100	1075

TABLE 5.5 – Impact du filtre analysant la complexité du graphe autour des points de cassure sur le rappel, la précision et le temps d'éxécution de *MindTheGap*. La première valeur du nombre de points de cassure représente les points de cassure homozygotes et la seconde ceux qui sont hétérozygotes.

Sur le chromosome 3, le filtre sur l'analyse du contexte génomique permet de réduire le nombre de faux positifs de 45% à 92% pour des seuils allant de 60% à 90% respectivement. L'ajout du filtre ne réduit pas le temps d'éxecution général de l'outil sur des simulations d'insertions dans le chromosome 3. Les différents seuils n'impactent ni le rappel, ni la précision de l'assemblage des insertions. L'utilisation du filtre sur la simulation du génome entier induit une perte de 1% du rappel et un gain de 59% de la précision lors de la détection des points de cassure. 18 heures ont été nécessaires pour assembler les 4 319 points de cassure alors qu'il en nécessiterait environ quatre fois plus pour assembler les 13 670 points de cassure non filtrés (Table 5.5).

5.3.4 Réduction de l'espace de recherche

Une réduction de l'espace de recherche dans le module *Find* a été apportée pour permettre une utilisation de *MindTheGap* pour l'analyse de séquençages d'exomes. Cette réduction nécessite l'apport par l'utilisateur d'un fichier au format BED pour indiquer les régions génomiques à scanner.

La réduction de l'espace de recherche appliquée au scénario de référence sur le chromosome 3 n'a pas impacté le rappel de l'outil (Table 5.6). La précision sur les points de cassure est multipliée par 3 et le temps d'éxécution des deux modules a été divisé par 7 grâce à ce filtre. Les observations réalisées sur le chromosome 3 sont similaires lors de la simulation d'insertions sur

génome entier. La quantité de faux positifs générés dans le module *Find* est principalement causée par l'identification de motifs d'insertions hétérozygotes erronés. Le nombre de faux positifs a été multiplié par 25 en comparaison à ceux obtenus lors de la simulation sur le chromosome 3. Cette quantité conduit a un temps d'éxécution du module *Fill* qui ne lui permet pas d'aboutir. La réduction de l'espace de recherche sur génome entier permet d'éviter la détection de ces faux positifs. Le temps d'éxécution du module *Fill* est divisé par 4, atteignant 15 heures. L'assemblage permet d'obtenir la séquence de 53 % des insertions simulées. Ce résultat révèle que 33% des insertions dont les points de cassure ont été identifiés n'ont pas été assemblées. Le passage à l'échelle du génome entier produit un graphe plus complexe à explorer. L'analyse de l'historique de l'exploration de ces points de cassure montre que les limites d'exploration ont été atteintes avant de trouver la graine.

		Module Find			**Module Fill**		
Echelle	Outil	Rappel (%)	Précision (%)	Temps (min)	Rappel (%)	Précision (%)	Temps (min)
Chromosome 3	MTG	94	32	8	94	94	7
	MTG+fichier bed	94	100	1	94	94	1.3
Génome entier	MTG	87	27	140	NA	NA	> 4000
	MTG+fichier bed	87	100	10	53	100	900

TABLE 5.6 – Impact de la réduction de l'espace de recherche sur le rappel, la précision et le temps d'éxécution.

5.4 Discussion

5.4.1 Améliorations de MindTheGap

A travers ce chapitre, nous avons essayé d'explorer des voies d'amélioration pour le *variant caller MindTheGap*. Pour identifier ces voies, il a été nécessaire de comprendre le fonctionnement précis de l'outil et de l'évaluer dans différentes situations. Tandis que des améliorations ne nécessitaient qu'une modification des paramètres disponibles, d'autres ont nécessité des changements en profondeur de l'implémentation de l'outil.

L'implémentation modulaire de *MindTheGap* simplifie l'identification des limites de l'outil. Le module *Find* renvoie l'ensemble des points de cassure dans un fichier. Ce fichier permet d'avoir un premier rappel et précision. Il devient alors possible de comparer les résultats du premier module avec les insertions assemblées du second module. Cette comparaison permet

d'identifier si les limites de l'outils se situent dans la détection des points de cassure ou dans l'assemblage.

Nos évaluations permettent de définir les différents contextes d'utilisation de *MindTheGap*, ainsi que ses limites. La réduction de l'espace de recherche s'est montrée pertinente dans la détection d'insertions dans des régions exonique et qui pourrait être applicable pour un usage médical. L'amélioration de la détection des insertions avec de grandes homologies jonctionnelle offre des résultats contrastés. L'augmentation du paramètre associé *max-repeat* de l'outil conduit à la recherche de motifs moins spécifiques, sujets à une plus grande découverte de faux positifs. Cependant, cette augmentation de détection des points de cassure conduit naturellement à un temps dédié à l'assemblage plus important. La réduction des faux positifs est devenue une priorité dans le but d'utiliser *MindTheGap* sur des données de séquençage du génome entier. Les premiers résultats du filtre basé sur l'analyse de la complexité du graphe sont encourageants. Ce filtre permet de réduire les faux positifs sans impacter le rappel dans la détéction de vrais positifs. La simulation d'insertions hétérozygotes sera nécessaire afin d'identifier un seuil de complexité qui permettra de dissocier les vrais insertions des fausses.

Nos simulations sur génome entier nous ont permis d'identifié que l'exploration du graphe limite l'assemblage d'insertions. Nous avons pu ainsi observer que les limites de *MindTheGap* sur génome entier sont doubles : les points de cassure sont plus difficilement détectables et le graphe est plus complexe que sur un chromosome, limitant l'assemblage des insertions.

A l'issue du chapitre précédent, *MindTheGap* se présentait comme un bon candidat pour apporter une solution à l'assemblage limité des variant callers génériques. Notre exploration de cet outil a révélé que la méthode d'assemblage de *MindTheGap* offre une meilleure opportunité d'obtenir la séquence des insertions. Néanmoins l'approche proposée possède plusieurs faiblesses dont la première concerne la détection des points de cassure. L'utilisation de k-mers dans le but de détecter des points de cassure est une approche intéressante mais qui réduit encore plus l'information portée par les *reads* entiers. La détection de motifs associés à des insertions hétérozygotes s'est montrée peu spécifique conduisant à un grand nombre de faux positifs. La seconde faiblesse est l'assemblage. Il est vrai qu'en théorie, l'utilisation de l'ensemble des *reads* permet l'assemblage de la séquence insérée. Cependant, le graphe produit est si complexe que son exploration est rendue difficile. Une exploration appuyée par un alignement des *reads* sur le graphe serait susceptible d'aider à la décision de l'exploration de chemins.

5.4.2 Application de MindTheGap à des données cliniques

Des collaborations avec l'Inserm de Grenoble, ainsi qu'avec le Centre Hospitalier Université de Dijon ont permis l'utilisation de *MindTheGap* sur des données de séquençage d'exomes dont certaines variations impliquées dans la maladie des patients sont connues. Les essais réalisés sur une trentaine de séquençages d'exomes n'ont pas permis de retrouver les variations attendues. Les variations à détecter dans les jeux de données étaient des CNV dont les localisations ne sont pas précisemment connues, ce qui rend difficile une comparaison avec *MindTheGap*. Ces premiers essais révèlent les difficultés à évaluer les *variant callers* sur des données réelles. Les insertions comprises entre 50 paires de bases et 1 kilobase sont peu recherchées dans le diagnostic clinique car aucune méthode d'analyse standardisée n'existe. Les données de séquençage de patients qui contiennent de tels variants connus sont rares, il est donc difficile d'utiliser ces données dans un but d'évaluation de l'outil. Cependant, une réussite de cette expérience a été l'utilisation de MindTheGap sur des données réelles par d'autres utilisateurs que son développeur et d'avoir des retours sur l'outil.

Pour surpasser les limites actuelles d'évaluation sur données réelles, il devient nécessaire de prospecter, avec des *variant callers*, des données de séquençage dont la variation responsable de la maladie n'a pas pu être identifiée. L'identification de ces nouveaux variants permettra de définir des données de référence pour évaluer les *variant callers* sur des données réelles. Cependant, la recherche de matériel de référence et d'évaluation des *variant callers* constitue pour l'heure actuelle une question de recherche et non de diagnostic. Il devient donc important que des collaborations entre chercheurs et cliniciens se poursuivent dans le but d'améliorer le diagnostic de maladies rares.

Chapitre 6

Conclusion et perspectives

Cette thèse s'inscrit dans un objectif d'apporter des pistes d'améliorations du *variant calling* dans un usage clinique. La réalisation de cet objectif a nécessité de comprendre en amont les limites de la détection actuelle. Bien que de nouvelles technologies de séquençage se développent, la technologie de séquençage de seconde génération reste la plus démocratisée. Cette contrainte conduit à l'utilisation de *variant callers* basés sur cette technologie. La littérature mentionne plus de 70 *variant callers* qui suggère une difficulté à identifier efficacement des variants à partir de ce type de données.

La connaissance de l'ensemble des facteurs induisant une faible détection des variants complexes tels que les variants de structure sont encore méconnus. Parmi ces variants, les insertions représentent un des types de variants les moins caractérisés à l'heure actuelle. Grâce à de nouveaux ensemble de variants de référence, obtenus grâce à de multiples technologies de séquençage, il a été possible de comprendre les raisons de la faible identification des insertions par des *variant callers*.

Les travaux menés dans le cadre de cette thèse se sont donc décomposés en trois contributions. Le premier fut de comprendre, à partir de ces jeux de données récents, les caractéristiques des différentes insertions. A l'aide de simulations, nous avons pu identifié les facteurs qui impactaient la détection des *variant callers*, ainsi que les limites de ces outils. Enfin la dernière contribution fut de proposer une exploration d'améliorations possibles d'un *variant caller*.

6.1 Facteurs impactant la détection d'insertion

Le premier travail, présenté au Chapitre 3, fut la caractérisation détaillée des insertions identifiables chez un individu humain. Pour cela, nous avons développé la première méthode d'annotation standardisée des variants de type insertion. Une telle méthode a permis de caractériser les insertions au delà d'une simple séquence insérée. Grâce à cette méthode, il devient possible d'observer la distribution des différents types d'insertions détectés au sein d'un jeu de données et des caractéristiques associées à chaque type d'insertion. Nous avons pu aisni

identifier que la taille, la localisation et la présence d'homologie jonctionnelle diffère entre type d'insertion. La similarité de ces distributions entre les individus, provenant d'études et de méthodologie différentes, révèle une représentativité des insertions pouvant être attendue dans un génome humain.

Les travaux réalisés durant cette thèse concernant les insertions, ainsi que ceux réalisés par Chaisson et al. sur les inversion ont permis de dresser des profils de ces variants. La méthode d'annotation s'est montrée indispensable pour séparer les différents variants des *callsets* et nous permettre de dresser ces profils. Nous avons récemment étendue cette annotation aux délétions, ce qui va permettre de caractériser les délétions. Identifier les propriétés des variants de structure représente donc un enjeu fondamental pour améliorer les algorithmes de détection et pour obtenir des *variant callers* performants sur des données réelles.

6.2 Evaluation des limitations des outils de détection courts reads

Le second travail, présenté au Chapitre 4, fut de vérifier et de quantifier l'impact des facteurs identifiés au Chapitre 3 grâce à des simulations. Cet objectif a nécessité le développement d'un simulateur d'insertions, où chaque facteur devait pouvoir se traiter de manière indépendante. En effet, bien que des simulateurs de variants existent (voir Etat de l'art), aucun ne proposait l'ensemble des caractéristiques que nous souhaitions simuler.

Les outils testés présentent des résultats hétérogènes qui peuvent expliquer les différences de rappel observées dans le Chapitre 3. Tous les facteurs, simulés indépendamment, affecte le rappel d'un ou de plusieurs outils. La synergie de plusieurs facteurs, contenues dans les insertions réelles présente le scénario le plus difficile pour l'ensemble des outils. Les faibles performances de ces outils peuvent avoir de multiples explications, dont la première est une évaluation limitée lors du design et du développement de l'outil. En cause, l'absence de *callset* de référence de haute qualité avant la production de Chaisson et al. et du GiaB. La seconde peut être la limitation de la technologie de séquençage, dont l'alignement des *reads* ne permet pas la détection des points de cassure d'une façon exacte. Dans l'ensemble, les outils possèdent des capacités limitées à assembler et rapporter la séquence des insertions. Cette limite a peu été décrite dans la littérature et est essentiellement soulevée dans les outils qui visent à raffiner les *variant calls*. Notre étude suggère que la selection d'un échantillon de *reads* pourrait être responsable de cette faiblesse.

A travers cette étude, nous avons démontré la nécessité de réaliser des simulations les plus

exhaustives possibles. Pour cela, il est nécessaire de réaliser l'ensemble des perspectives proposés dans la partie précédente. La caractérisation fine des différents variants sur données réelles permettra la déconstruction des différentes propriétés des variants. Ceci permettra une évaluation des facteurs pris indépendamment, puis combinés. Il sera alors possible d'adapter les outils de *variant calling* afin de les rendre performants sur données réelles.

6.3 Améliorations du variant caller MindTheGap

Le troisième et dernier travail, présenté au Chapitre 5, fut de proposer des pistes d'améliorations pour le *variant calling*. Nous avons donc sélectionné un *variant caller* et identifié des voies d'améliorations pour la détection d'insertions. *MindTheGap* s'est révélé être un bon candidat pour la recherche d'améliorations. Son fonctionnement est séparé en deux modules, l'un est dédié à la détection de points de cassure, et le second est dédié à l'assemblage. Cette séparation permet d'identifier si la baisse de rappel est induite par une absence de reconnaissance de points de cassure ou par une difficulté d'assemblage.

La première amélioration fut d'explorer la gestion des homologies jonctionnelles par MindTheGap. Cette caractéristique a déjà été prise en compte et implémentée dans l'outil lors de son développement. Néanmoins le paramètre par défaut associé aux homologies permet uniquement de détecter des homologies inférieures à 5 paires de bases. L'utilisateur peut, par la modification de ce paramètre, permettre la détection d'insertions avec de plus grandes homologies. L'augmentation de la valeur du paramètre associé aux homologies jonctionnelles conduit également à une augmentation du nombre de faux positifs hétérozygotes détectés. Une meilleure gestion des faux positifs est ainsi devenue nécessaire pour améliorer le rappel sur données réelles.

La seconde piste d'amélioration que nous avons exploré est la réduction de l'espace de recherche. Cette amélioration a été motivée par une volonté d'utiliser MindTheGap dans un contexte d'analyse de séquençage d'exomes. L'utilisation de cette réduction a permis une réduction significative du temps d'éxécution des deux modules de l'outil. La simulation d'insertion sur génome entier a révélé que *MindTheGap* ne passe pas à l'échelle dans sa version native. Sans la réduction de l'espace de recherche, l'outil détecte de nombreux points de cassure hétérozygotes. La principale cause est attribuée aux répétitions dans le génome qui conduisent à la présence de branchements dans le graphe. Cette augmentation de points de cassure induit naturellement une augmentation importante du temps d'assemblage. Ce temps d'éxécution ne permet pas une utilisation de *MindTheGap* et nous a conduit au développement d'une troisième amélioration.

La troisième amélioration fut de réduire le nombre de faux positifs afin de rendre utilisable MindTheGap sur des données de séquençage de génomes. Nous avons basé cette amélioration sur l'hypothèse que les faux positifs sont induits par des régions qui contiennent beaucoup de branchements dans le graphe. Un seuil maximum de branchements autour de chaque point de cassure a été ajouté pour réduire le nombre de faux positif. Ce seuil a permis de réduire significativement le nombre de faux positifs à assembler, sans impact sur le rappel. Il est intéressant de noter que les faux positifs ne semblent pas liés à des régions répétées. En effet, la simulation d'insertions dans des régions répétées présentaient le même profil de k-mers branchants que la simulation d'insertions dans des régions exoniques. La récupération des k-mers dans les régions produisant des faux positifs et la localisation de ces derniers sur le génome de référence permettra d'identifier ce qui les relie.

34% des insertions simulées dans un génome entier et dont le point de cassure a été identifié n'ont pas pu être assemblées à cause des limites imposées lors de l'exploration du graphe. Plusieurs pistes sont disponibles pour améliorer l'assemblage des insertions d'une façon générale. La première est d'améliorer la séléction de *reads* pour réaliser un assemblage rapide et simple à explorer. Pour que cette approche fonctionne deux étapes itératives sont nécessaires. La première est l'assemblage des *reads* sélectionnés pouvant être associés au début et à la fin de l'insertion. La seconde est l'alignement des *reads* sur les fragments assemblés pour receuillir de nouveaux *reads* qui seront utilisés pour un second assemblage. Les deux étapes sont ensuite utilisées de manière itératives pour permettre l'assemblage d'insertions longues. La seconde est d'améliorer l'exploration du graphe, en évitant des chemins qui ne sont pas soutenus par les *reads*. Cette approche nécessite un alignement des reads sur graphe, qui peut être réalisé par des outils tels que BGREAT[82]. Il est important de noter que l'ensemble de ces approches pourront difficilement surpasser les limites imposées par la technologie de séquençage. Par exemple, l'assemblage de répétition en tandem dans leur intégrité ne pourront être obtenues.

Une grossière approximation de notre évaluation révèle qu'une insertion nécessite en moyenne 4 minutes pour être assemblée. L'état actuel de l'outil ne permet pas son utilisation avec des données de séquençage réelles où plus de 15 000 insertions sont attendues. L'outil reste une preuve que l'utilisation de l'ensemble des *reads* peut permettre d'améliorer l'assemblage. La mise en place des approches que nous avons proposées pour améliorer l'assemblage présente également le désavantage d'augmenter encore plus le temps d'assemblage.

6.4 Perspectives pour le diagnostic clinique

Le Plan France Médecine Génomique a pour objectif de mettre en place des plateformes dont le rôle sera de séquencer et de réaliser la détection de variants. L'absence de recommandation pour la détection de variants complexes, d'une taille entre 50 paires de bases et 1 kilobase, ainsi que les faibles performances des *variant callers* compliquent le développement de *pipelines* d'analyse. Nos travaux ont mis en évidence les précautions à prendre lors de l'utilisation de *variant callers* et les limites de ces outils. Il est, pour l'heure actuelle, difficile de conseiller l'utilisation d'un ou plusieurs *variant callers* dans un but d'usage clinique. Chacun possède des avantages et des inconvénients qui sont nécessaires d'identifier en amont des analyses sur données réelles. Il est important que l'utilisateur connaisse l'ensemble des possibilités et des limites de l'outil qu'il souhaite utiliser. Pour cela, un oeil critique doit être appliqué aux publications d'outils qui présentent systématiquement des résultats meilleurs aux outils précédents. Un oeil critique est également requis concernant les publications d'évaluations d'outils. La compréhension du contexte et des méthodes de simulations et d'évaluations peut révéler des biais cachés des outils de détection. Nous recommandons d'utiliser des simulations les plus proches des données réelles et de ce que recherche l'utilisateur. Ainsi l'utilisateur sera capable d'identifier les outils pouvant correspondre à ses besoins.

Le diagnostic médical nécessite la détection précise des variants susceptibles d'avoir un rôle dans les maladies génétiques. La caractérisation de la position et de la séquence sont indispensables pour permettre d'identifier l'impact des variants sur le fonctionnement des gènes. La capacité de *MindTheGap* à assembler des grandes insertions se présente comme un atout majeur pour un usage médical. Nos premières évaluations sur données de séquençage réelles n'ont pas permis de détecter les variations attendues. Les variants se composaient principalement de CNV, détectable par *MindTheGap* mais dont la position attendue n'était pas connue avec les méthodes de détection standard de CNV. Il était donc difficile de comparer les variants attendus avec ceux découverts par *MindTheGap*. Cette expérience ne démontre pas l'absence d'efficacité de *MindTheGap* mais de la difficulté à évaluer un outil sur des données qui rentrent dans le champ d'utilisation de l'outil. Cependant, ce type de données est rare et les variants associés encore trop peu identifiés dû au manque de méthodes d'analyses standardisés. L'une des voies à explorer pour surpasser cette limite serait l'intégration de multiples *variant callers* dans la routine du diagnostic médical. Ces *variant callers* permettront de prospecter de nouveaux variants rares et de générer de nouvelles données réelles de référence qui pourront être utilisées pour évaluer les *variant callers*.

Cette thèse avait pour premier but d'apporter des contributions méthodologiques pour améliorer la détection de variants complexes. Au cours de cette thèse nous avons pu observer et comprendre la difficulté à détecter un type de variant parmi tant d'autres. L'amélioration de la détection a nécessité un travail conséquent de compréhension des outils actuels mais également de caractériser les variants pour mieux les détecter. Cette thèse n'a donc pas permis d'apporter une solution au problème de la détection de variants à partir de données de séquençage de seconde génération. Ce travail a apporté un ensemble de pistes pour aller vers cet objectif. Il est ainsi nécessaire de poursuivre ces efforts afin d'améliorer notre compréhension du génome et d'en améliorer son analyse à travers des outils de bioinformatique.

BIBLIOGRAPHIE

[1] Daniel AIRD et al. « Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries ». In : *Genome biology* 12.2 (2011), p. 1-14.

[2] Bruce ALBERTS et al. « Molecular biology of the cell, 5th edn. Garland Science ». In : *New York* (2008).

[3] Can ALKAN, Bradley P COE et Evan E EICHLER. « Genome structural variation discovery and genotyping ». In : *Nature Reviews Genetics* 12.5 (2011), p. 363-376.

[4] Stephen F ALTSCHUL et al. « Basic local alignment search tool ». In : *Journal of molecular biology* 215.3 (1990), p. 403-410.

[5] Jonathan W ARTHUR, Florence SG CHEUNG et Juergen KV REICHARDT. « Single nucleotide differences (SNDs) continue to contaminate the dbSNP database with consequences for human genomics and health ». In : *Human mutation* 36.2 (2015), p. 196-199.

[6] Jeffrey A BAILEY, Ge LIU et Evan E EICHLER. « An Alu transposition model for the origin and expansion of human segmental duplications ». In : *The American Journal of Human Genetics* 73.4 (2003), p. 823-834.

[7] Jørgen BANG-JENSEN, Gregory GUTIN et Anders YEO. « When the greedy algorithm fails ». In : *Discrete optimization* 1.2 (2004), p. 121-127.

[8] Christoph BARTENHAGEN et Martin DUGAS. « RSVSim : an R/Bioconductor package for the simulation of structural variations ». In : *Bioinformatics* 29.13 (2013), p. 1679-1681.

[9] Sabina BENKO et al. « Highly conserved non-coding elements on either side of SOX9 associated with Pierre Robin sequence ». In : *Nature genetics* 41.3 (2009), p. 359-364.

[10] Gary BENSON. « Tandem repeats finder : a program to analyze DNA sequences ». In : *Nucleic acids research* 27.2 (1999), p. 573-580.

[11] Miles C BENTON et al. « Variant Call Format–Diagnostic Annotation and Reporting Tool : A Customizable Analysis Pipeline for Identification of Clinically Relevant Genetic Variants in Next-Generation Sequencing Data ». In : *The Journal of Molecular Diagnostics* 21.6 (2019), p. 951-960.

[12] C Binquet et al. « Faisabilité et efficience du séquençage du génome en première intention pour le diagnostic étiologique des déficiences intellectuelles : l'étude DEFIDIAG ». In : *Revue d'Épidémiologie et de Santé Publique* 66 (2018), S171.

[13] Michael Burrows et David J Wheeler. « A block-sorting lossless data compression algorithm ». In : (1994).

[14] Keith W Caldecott. « Single-strand break repair and genetic disease ». In : *Nature Reviews Genetics* 9.8 (2008), p. 619-631.

[15] Daniel L Cameron, Leon Di Stefano et Anthony T Papenfuss. « Comprehensive evaluation and characterisation of short read general-purpose structural variant calling software ». In : *Nature communications* 10.1 (2019), p. 1-11.

[16] Daniel L Cameron et al. « GRIDSS : sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly ». In : *Genome research* 27.12 (2017), p. 2050-2060.

[17] Mark JP Chaisson et al. « Multi-platform discovery of haplotype-resolved structural variation in human genomes ». In : *Nature communications* 10.1 (2019), p. 1-16.

[18] Mark JP Chaisson et al. « Resolving the complexity of the human genome using single-molecule sequencing ». In : *Nature* 517.7536 (2015), p. 608-611.

[19] Varuna Chander, Richard A Gibbs et Fritz J Sedlazeck. « Evaluation of computational genotyping of structural variation for clinical diagnoses ». In : *GigaScience* 8.9 (sept. 2019). doi : 10.1093/gigascience/giz110.

[20] Xiaoyu Chen et al. « Manta : rapid detection of structural variants and indels for germline and cancer sequencing applications ». In : *Bioinformatics* 32.8 (2016), p. 1220-1222.

[21] Xun Chen et Dawei Li. « ERVcaller : identifying polymorphic endogenous retrovirus and other transposable element insertions using whole-genome sequencing data ». In : *Bioinformatics* 35.20 (2019), p. 3913-3922.

[22] Chia-Yi Cheng et al. « Araport11 : a complete reannotation of the Arabidopsis thaliana reference genome ». In : *The Plant Journal* 89.4 (2017), p. 789-804.

[23] Rayan Chikhi et Guillaume Rizk. « Space-efficient and exact de Bruijn graph representation based on a Bloom filter ». In : *Algorithms for Molecular Biology* 8.1 (2013), p. 22.

[24] Chong Chu, Xin Li et Yufeng Wu. « SpliceJumper : a classification-based approach for calling splicing junctions from RNA-seq data ». In : *BMC bioinformatics* 16.S17 (2015), S10.

[25] Megan H Cleveland et al. « Determining performance metrics for targeted next-generation sequencing panels using reference materials ». In : *The Journal of Molecular Diagnostics* 20.5 (2018), p. 583-590.

[26] Irun R Cohen, Henri Atlan et Sol Efroni. « Genetics as explanation : limits to the human genome project ». In : *eLS* (2009).

[27] Commons wikimedia. *wikimedia commons, the free media repository*. 2020.

[28] Donald F Conrad et al. « Mutation spectrum revealed by breakpoint sequencing of human germline CNVs ». In : *Nature genetics* 42.5 (2010), p. 385.

[29] Richard Cordaux et Mark A Batzer. « The impact of retrotransposons on human genome evolution ». In : *Nature Reviews Genetics* 10.10 (2009), p. 691-703.

[30] Jacob F Degner et al. « Effect of read-mapping biases on detecting allele-specific expression from RNA-sequencing data ». In : *Bioinformatics* 25.24 (2009), p. 3207-3212.

[31] Elie Dolgin. « Human genomics : The genome finishers ». In : *Nature* 462.7275 (2009), p. 843-846.

[32] Egor Dolzhenko et al. « ExpansionHunter : a sequence-graph-based tool to analyze variation in short tandem repeat regions ». In : *Bioinformatics* 35.22 (2019), p. 4754-4756.

[33] Erwan Drezen et al. « GATB : genome assembly & analysis tool box ». In : *Bioinformatics* 30.20 (2014), p. 2959-2961.

[34] Evan E Eichler. « Genetic variation, comparative genomics, and the diagnosis of disease ». In : *New England Journal of Medicine* 381.1 (2019), p. 64-74.

[35] Jesper Eisfeldt et al. « TIDDIT, an efficient and comprehensive structural variant caller for massive parallel sequencing data ». In : *F1000Research* 6 (2017).

[36] Beverly S Emanuel et Tamim H Shaikh. « Segmental duplications : an'expanding'role in genomic instability and disease ». In : *Nature Reviews Genetics* 2.10 (2001), p. 791-800.

[37] Xian Fan et al. « BreakDancer : Identification of genomic structural variation from paired-end read mapping ». In : *Current protocols in bioinformatics* 45.1 (2014), p. 15-6.

[38] Gregory G Faust et Ira M Hall. « SAMBLASTER : fast duplicate marking and structural variant read extraction ». In : *Bioinformatics* 30.17 (2014), p. 2503-2505.

[39] Paolo Ferragina et Giovanni Manzini. « Opportunistic data structures with applications ». In : *Proceedings 41st Annual Symposium on Foundations of Computer Science*. IEEE. 2000, p. 390-398.

[40] Eugene J Gardner et al. « The Mobile Element Locator Tool (MELT) : population-scale mobile element discovery and biology ». In : *Genome research* 27.11 (2017), p. 1916-1929.

[41] *GENCODE - Human Release Statistics*. URL : `https://www.gencodegenes.org/human/stats.html` (visité le 10/08/2020).

[42] Véronique Geoffroy et al. « AnnotSV : an integrated tool for structural variations annotation ». In : *Bioinformatics* 34.20 (2018), p. 3572-3574.

[43] Christian Gilissen et al. « Disease gene identification strategies for exome sequencing ». In : *European Journal of Human Genetics* 20.5 (2012), p. 490-497.

[44] Christian Gilissen et al. « Unlocking Mendelian disease using exome sequencing ». In : *Genome biology* 12.9 (2011), p. 228.

[45] Wenli Gu, Feng Zhang et James R Lupski. « Mechanisms for human genomic rearrangements ». In : *Pathogenetics* 1.1 (2008), p. 4.

[46] Deepti Gurdasani et al. « The African genome variation project shapes medical genetics in Africa ». In : *Nature* 517.7534 (2015), p. 327-332.

[47] Iman Hajirasouliha et al. « Detection and characterization of novel sequence insertions using paired-end next-generation sequencing ». In : *Bioinformatics* 26.10 (2010), p. 1277-1283.

[48] Julien Häsler et Katharina Strub. « Alu elements as regulators of gene expression ». In : *Nucleic acids research* 34.19 (2006), p. 5491-5497.

[49] Manuel Holtgrewe, Leon Kuchenbecker et Knut Reinert. « Methods for the detection and assembly of novel sequence in high-throughput sequencing data ». In : *Bioinformatics* 31.12 (2015), p. 1904-1912.

[50] Weichun Huang et al. « ART : a next-generation sequencing read simulator ». In : *Bioinformatics* 28.4 (2012), p. 593-594.

[51] Robert Hubley et al. « The Dfam database of repetitive DNA families ». In : *Nucleic acids research* 44.D1 (2016), p. D81-D89.

[52] *Human Genome Project FAQ*. Genome.gov. URL : `https://www.genome.gov/human-genome-project/Completion-FAQ` (visité le 10/08/2020).

[53] Miten Jain et al. « Nanopore sequencing and assembly of a human genome with ultra-long reads ». In : *Nature biotechnology* 36.4 (2018), p. 338-345.

[54] FA Janssens, Romain Koszul et Denise Zickler. « La Theorie de la Chiasmatypie : Nouvelle interprétation des cinèses de maturation ». In : *Genetics* 191.2 (2012), p. 319.

[55] Emre Karakoc et al. « Detection of structural variants and indels within exome data ». In : *Nature methods* 9.2 (2012), p. 176-178.

[56] Zarir E Karanjawala et al. « Oxygen metabolism causes chromosome breaks and is associated with the neuronal apoptosis observed in DNA double-strand break repair mutants ». In : *Current biology* 12.5 (2002), p. 397-402.

[57] Mehran Karimzadeh et al. « Umap and Bismap : quantifying genome and methylome mappability ». In : *Nucleic acids research* 46.20 (2018), e120-e120.

[58] Donna Karolchik et al. « The UCSC genome browser database ». In : *Nucleic acids research* 31.1 (2003), p. 51-54.

[59] Masaharu Kataoka et al. « Alu-mediated nonallelic homologous and nonhomologous recombination in the BMPR2 gene in heritable pulmonary arterial hypertension ». In : *Genetics in Medicine* 15.12 (2013), p. 941-947.

[60] Pınar Kavak et al. « Discovery and genotyping of novel sequence insertions in many sequenced individuals ». In : *Bioinformatics* 33.14 (2017), p. i161-i169.

[61] Birte Kehr, Páll Melsted et Bjarni V Halldórsson. « PopIns : population-scale detection of novel sequence insertions ». In : *Bioinformatics* 32.7 (2016), p. 961-967.

[62] W James Kent. « BLAT—the BLAST-like alignment tool ». In : *Genome research* 12.4 (2002), p. 656-664.

[63] Jeffrey M Kidd et al. « A human genome structural variation sequencing resource reveals insights into mutational mechanisms ». In : *Cell* 143.5 (2010), p. 837-847.

[64] Jan O Korbel et al. « PEMer : a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data ». In : *Genome biology* 10.2 (2009), R23.

[65] Shunichi Kosugi et al. « Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing ». In : *Genome Biology* 20.1 (juin 2019). doi : 10.1186/s13059-019-1720-5.

[66] Peter Krusche et al. « Best practices for benchmarking germline small-variant calls in human genomes ». In : *Nature biotechnology* 37.5 (2019), p. 555-560.

[67] Arnold Kuzniar et al. « sv-callers : a highly portable parallel workflow for structural variant detection in whole-genome sequence data ». In : *PeerJ* 8 (2020), e8214.

[68] Eric S Lander et al. « Initial sequencing and analysis of the human genome ». In : (2001).

[69] Ben Langmead et Steven L Salzberg. « Fast gapped-read alignment with Bowtie 2 ». In : *Nature methods* 9.4 (2012), p. 357.

[70] Ilkka Lappalainen et al. « DbVar and DGVa : public archives for genomic structural variation ». In : *Nucleic acids research* 41.D1 (2012), p. D936-D941.

[71] Ryan M Layer et al. « LUMPY : a probabilistic framework for structural variant discovery ». In : *Genome biology* 15.6 (2014), R84.

[72] Hane Lee et al. « Clinical exome sequencing for genetic identification of rare Mendelian disorders ». In : *Jama* 312.18 (2014), p. 1880-1887.

[73] Jennifer A Lee, Claudia MB Carvalho et James R Lupski. « A DNA replication mechanism for generating nonrecurrent rearrangements associated with genomic disorders ». In : *cell* 131.7 (2007), p. 1235-1247.

[74] Heng Li. « FermiKit : assembly-based variant calling for Illumina resequencing data ». In : *Bioinformatics* 31.22 (2015), p. 3694-3696.

[75] Heng Li. « Minimap2 : pairwise alignment for nucleotide sequences ». In : *Bioinformatics* 34.18 (2018), p. 3094-3100.

[76] Heng Li et Richard Durbin. « Fast and accurate short read alignment with Burrows–Wheeler transform ». In : *bioinformatics* 25.14 (2009), p. 1754-1760.

[77] Heng Li, Jue Ruan et Richard Durbin. « Mapping short DNA sequencing reads and calling variants using mapping quality scores ». In : *Genome research* 18.11 (2008), p. 1851-1858.

[78] Heng Li et al. « The sequence alignment/map format and SAMtools ». In : *Bioinformatics* 25.16 (2009), p. 2078-2079.

[79] Yingrui Li et al. « Structural variation in two human genomes mapped at single-nucleotide resolution by whole genome de novo assembly ». In : *Nature biotechnology* 29.8 (2011), p. 723-730.

[80] Zhenyu Li et al. « Comparison of the two major classes of assembly algorithms : overlap–layout–consensus and de-bruijn-graph ». In : *Briefings in functional genomics* 11.1 (2012), p. 25-37.

[81] Michael R Lieber. « The mechanism of double-strand DNA break repair by the non-homologous DNA end-joining pathway ». In : *Annual review of biochemistry* 79 (2010), p. 181-211.

[82] Antoine Limasset et al. « Read mapping on de Bruijn graphs ». In : *BMC bioinformatics* 17.1 (2016), p. 1-12.

[83] Glennis A Logsdon, Mitchell R Vollger et Evan E Eichler. « Long-read human genome sequencing and its applications ». In : *Nature Reviews Genetics* (2020), p. 1-18.

[84] James R Lupski et al. « Whole-genome sequencing in a patient with Charcot–Marie–Tooth neuropathy ». In : *New England Journal of Medicine* 362.13 (2010), p. 1181-1191.

[85] Medhat Mahmoud et al. « Structural variant calling : the long and the short of it ». In : *Genome biology* 20.1 (2019), p. 246.

[86] Medhat Mahmoud et al. « Structural variant calling : the long and the short of it ». In : *Genome Biology* 20.1 (nov. 2019). doi : 10.1186/s13059-019-1828-7.

[87] Tobias Marschall et al. « CLEVER : clique-enumerating variant finder ». In : *Bioinformatics* 28.22 (2012), p. 2875-2882.

[88] Allan M Maxam et Walter Gilbert. « A new method for sequencing DNA ». In : *Proceedings of the National Academy of Sciences* 74.2 (1977), p. 560-564.

[89] Aaron McKenna et al. « The Genome Analysis Toolkit : a MapReduce framework for analyzing next-generation DNA sequencing data ». In : *Genome research* 20.9 (2010), p. 1297-1303.

[90] Gregor Mendel. « Versuche uber pflanzen-hybriden ». In : *Verhandlungen des naturforschenden Vereins in Brunn fur* 4 (1866), p. 3-47.

[91] Michael L Metzker. « Sequencing technologies—the next generation ». In : *Nature reviews genetics* 11.1 (2010), p. 31-46.

[92] Jacob J Michaelson et Jonathan Sebat. « forestSV : structural variant discovery through statistical learning ». In : *Nature methods* 9.8 (2012), p. 819-821.

[93] Karen H Miga et al. « Telomere-to-telomere assembly of a complete human X chromosome ». In : *Nature* 585.7823 (2020), p. 79-84.

[94] Takahiro Mimori et al. « iSVP : an integrated structural variant calling pipeline from high-throughput sequencing data ». In : *BMC systems biology* 7.6 (2013), p. 1-8.

[95] Marghoob Mohiyuddin et al. « MetaSV : an accurate and integrative structural-variant caller for next generation sequencing ». In : *Bioinformatics* 31.16 (2015), p. 2741-2744.

[96] John C Mu et al. « VarSim : a high-fidelity simulation and validation framework for high-throughput genome sequencing with cancer applications ». In : *Bioinformatics* 31.9 (2015), p. 1469-1471.

[97] Maria Nattestad et Michael C Schatz. « Assemblytics : a web analytics tool for the detection of variants from an assembly ». In : *Bioinformatics* 32.19 (2016), p. 3021-3023.

[98] Anna C Need et al. « Clinical application of exome sequencing in undiagnosed genetic conditions ». In : *Journal of medical genetics* 49.6 (2012), p. 353-361.

[99] Saul B. Needleman et Christian D. Wunsch. « A general method applicable to the search for similarities in the amino acid sequence of two proteins ». In : *Journal of Molecular Biology* 48.3 (1970), p. 443-453. issn : 0022-2836. doi : https://doi.org/10.1016/0022-2836(70)90057-4. url : http://www.sciencedirect.com/science/article/pii/0022283670900574.

[100] André Nussenzweig et Michel C Nussenzweig. « A backup DNA repair pathway moves to the forefront ». In : *Cell* 131.2 (2007), p. 223-225.

[101] Nuala A O'Leary et al. « Reference sequence (RefSeq) database at NCBI : current status, taxonomic expansion, and functional annotation ». In : *Nucleic acids research* 44.D1 (2016), p. D733-D745.

[102] Diego Ottaviani, Magdalena LeCain et Denise Sheer. « The role of microhomology in genomic structural variation ». In : *Trends in Genetics* 30.3 (2014), p. 85-94.

[103] Daniel Pinkel et Donna G Albertson. « Array comparative genomic hybridization and its applications in cancer ». In : *Nature genetics* 37.6 (2005), S11-S17.

[104] Kim D Pruitt et al. « NCBI Reference Sequences (RefSeq) : current status, new features and genome annotation policy ». In : *Nucleic acids research* 40.D1 (2012), p. D130-D135.

[105] Ji Qi et Fangqing Zhao. « inGAP-sv : a novel scheme to identify and visualize structural variation from paired end mapping data ». In : *Nucleic acids research* 39.suppl_2 (2011), W567-W575.

[106] Maochun Qin et al. « SCNVSim : somatic copy number variation and structure variation simulator ». In : *BMC bioinformatics* 16.1 (2015), p. 1-6.

[107] Aaron R Quinlan et al. « Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome ». In : *Genome research* 20.5 (2010), p. 623-635.

[108] Tobias Rausch et al. « DELLY : structural variant discovery by integrated paired-end and split-read analysis ». In : *Bioinformatics* 28.18 (2012), p. i333-i339.

[109] Guénola Ricard et al. « Phenotypic consequences of copy number variation : insights from Smith-Magenis and Potocki-Lupski syndrome mouse models ». In : *PLoS Biol* 8.11 (2010), e1000543.

[110] Guillaume Rizk et al. « MindTheGap : integrated detection and assembly of short and long insertions ». In : *Bioinformatics* 30.24 (2014), p. 3451-3457.

[111] Somak Roy et al. « Standards and guidelines for validating next-generation sequencing bioinformatics pipelines : a joint recommendation of the Association for Molecular Pathology and the College of American Pathologists ». In : *The Journal of Molecular Diagnostics* 20.1 (2018), p. 4-27.

[112] Joel Rozowsky et al. « AlleleSeq : analysis of allele-specific expression and binding in a network framework ». In : *Molecular systems biology* 7.1 (2011), p. 522.

[113] Frederick Sanger, Steven Nicklen et Alan R Coulson. « DNA sequencing with chain-terminating inhibitors ». In : *Proceedings of the national academy of sciences* 74.12 (1977), p. 5463-5467.

[114] Fritz J Sedlazeck et al. « Accurate detection of complex structural variations using single-molecule sequencing ». In : *Nature methods* 15.6 (2018), p. 461-468.

[115] Andrew J Sharp et al. « Segmental duplications and copy-number variation in the human genome ». In : *The American Journal of Human Genetics* 77.1 (2005), p. 78-88.

[116] Stephen T Sherry et al. « dbSNP : the NCBI database of genetic variation ». In : *Nucleic acids research* 29.1 (2001), p. 308-311.

[117] Supriya Sinha et al. « Risky business : Microhomology-mediated end joining ». In : *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis* 788 (2016), p. 17-24.

[118] Temple F Smith, Michael S Waterman et al. « Identification of common molecular subsequences ». In : *Journal of molecular biology* 147.1 (1981), p. 195-197.

[119] Janet HT Song, Craig B Lowe et David M Kingsley. « Characterization of a human-specific tandem repeat associated with bipolar disorder and schizophrenia ». In : *The American Journal of Human Genetics* 103.3 (2018), p. 421-430.

[120] Arda Soylev et al. « Discovery of tandem and interspersed segmental duplications using high-throughput sequencing ». In : *Bioinformatics* 35.20 (2019), p. 3923-3930.

[121] Malte Spielmann, Darıo G Lupiáñez et Stefan Mundlos. « Structural variation in the 3D genome ». In : *Nature Reviews Genetics* 19.7 (2018), p. 453-467.

[122] Noah Spies et al. « Svviz : A read viewer for validating structural variants ». In : *Bioinformatics* 31.24 (2015), p. 3994-3996.

[123] Pawel Stankiewicz, Amber N Pursley et Sau Wai Cheung. « Challenges in clinical interpretation of microduplications detected by array CGH analysis ». In : *American Journal of Medical Genetics Part A* 152.5 (2010), p. 1089-1100.

[124] Lincoln Stein. « Genome annotation : from sequence to biology ». In : *Nature reviews genetics* 2.7 (2001), p. 493-503.

[125] Bianca K Stöcker, Johannes Köster et Sven Rahmann. « SimLoRD : simulation of long read data ». In : *Bioinformatics* 32.17 (2016), p. 2704-2706.

[126] Patrick Sung et Hannah Klein. « Mechanism of homologous recombination : mediators and helicases take on regulatory functions ». In : *Nature reviews Molecular cell biology* 7.10 (2006), p. 739-750.

[127] Adrian Tan, Gonçalo R Abecasis et Hyun Min Kang. « Unified representation of genetic variants ». In : *Bioinformatics* 31.13 (2015), p. 2202-2204.

[128] Mustafa Taşkesen et al. « Novel Alu retrotransposon insertion leading to Alström syndrome ». In : *Human genetics* 131.3 (2012), p. 407-413.

[129] Danielle Thierry-Mieg et Jean Thierry-Mieg. « AceView : a comprehensive cDNA-supported gene and transcripts annotation ». In : *Genome biology* 7.S1 (2006), S12.

[130] Kathrin Trappe et al. « Gustaf : detecting and correctly classifying SVs in the NGS twilight zone ». In : *Bioinformatics* 30.24 (2014), p. 3484-3490.

[131] Michael C Velarde et al. « Mitochondrial oxidative stress caused by Sod2 deficiency promotes cellular senescence and aging phenotypes in the skin ». In : *Aging (Albany NY)* 4.1 (2012), p. 3.

[132] J Craig Venter et al. « The sequence of the human genome ». In : *science* 291.5507 (2001), p. 1304-1351.

[133] Marco Venturin et al. « Evidence for non-homologous end joining and non-allelic homologous recombination in atypical NF1 microdeletions ». In : *Human genetics* 115.1 (2004), p. 69-80.

[134] Hannah Verdin et al. « Microhomology-mediated mechanisms underlie non-recurrent disease-causing microdeletions of the FOXL2 gene or its regulatory domain ». In : *PLoS Genet* 9.3 (2013), e1003358.

[135] Julien Vignard, Gladys Mirey et Bernard Salles. « Ionizing-radiation induced DNA double-strand breaks : a direct and indirect lighting up ». In : *Radiotherapy and Oncology* 108.3 (2013), p. 362-369.

[136] Jeremiah A. Wala et al. « SvABA : genome-wide detection of structural variants and indels by local assembly ». In : *Genome Research* (mar. 2018). doi : 10.1101/gr.221028.117.

[137] Jianmin Wang et al. « CREST maps somatic structural variation in cancer genomes with base-pair resolution ». In : *Nature methods* 8.8 (2011), p. 652-654.

[138] Yifan Wang et al. « BRCA1 intronic Alu elements drive gene rearrangements and PARP inhibitor resistance ». In : *Nature communications* 10.1 (2019), p. 1-12.

[139] Kim Wong et al. « Enhanced structural variant and breakpoint detection using SVMerge by integration of multiple detection methods and local assembly ». In : *Genome biology* 11.12 (2010), R128.

[140] Lisa Woodbine, Andrew R Gennery et Penny A Jeggo. « The clinical impact of deficiency in DNA non-homologous end-joining ». In : *DNA repair* 16 (2014), p. 84-96.

[141] Li Charlie Xia et al. « SVEngine : an efficient and versatile simulator of genome structural variations with features of cancer clonal evolution ». In : *GigaScience* 7.7 (2018), giy081.

[142] Yuan Xue et al. « FGFR3 mutation frequency in 324 cases from the International Skeletal Dysplasia Registry ». In : *Molecular genetics & genomic medicine* 2.6 (2014), p. 497-503.

[143] Yaping Yang et al. « Clinical whole-exome sequencing for the diagnosis of mendelian disorders ». In : *New England Journal of Medicine* 369.16 (2013), p. 1502-1511.

[144] Kai Ye et al. « Pindel : a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads ». In : *Bioinformatics* 25.21 (2009), p. 2865-2871.

[145] Yuxuan Yuan, Claire Yik-Lok Chung et Ting-Fung Chan. « Advances in optical mapping for genomic research ». In : *Computational and Structural Biotechnology Journal* (2020).

[146] Samantha Zarate et al. « Parliament2 : Fast Structural Variant Calling Using Optimized Combinations of Callers ». In : *bioRxiv* (2018). doi : 10.1101/424267. eprint : https://www.biorxiv.org/content/early/2018/09/23/424267.full.pdf. url : https://www.biorxiv.org/content/early/2018/09/23/424267.

[147] Denise Zickler et Nancy Kleckner. « Meiotic chromosomes : integrating structure and function ». In : *Annual review of genetics* 33.1 (1999), p. 603-754.

[148] Justin M. Zook et al. « A robust benchmark for detection of germline large deletions and insertions ». In : *Nature Biotechnology* (juin 2020). doi : 10.1038/s41587-020-0538-8.

PUBLICATIONS

Articles

— Delage, Wesley, Thevenon, J., Lemaitre, C. (2020). Towards a better understanding of the low recall of insertion variants with short-read based variant callers. BMC Genomics.

— Guyomar, C.,Delage, Wesley, Legeai, F., Mougel, C., Simon, J.-C., Lemaitre, C. (2020). Minys : Mine your symbiont by targeted genome assembly in symbiotic communities. NAR Genomics and Bioinformatics.

— Meline, V.,Delage, Wesley, Brin, C., Li-Marchetti, C., Sochard, D., Arlat, M., Rousseau, C., Darrasse, A.,Briand, M., Lebreton, G. Et al. (2019). Role of the acquisition of a type 3 secretion system in the emergence of novel pathogenic strains of xanthomonas.Molecular plant pathology,20(1), 33–50.

Présentations

— Wesley Delage, J. T., Lemaitre, C. (2020). Towards a better understanding of the low discovery rate of short-read based insertion variant callers, Présentation orale aux journées ouvertes de Biologie, Informatique et Mathématique (JOBIM). Montpellier, 2020.

Posters

— Wesley Delage, J. T., Lemaitre, C. (2020). Towards a better understanding of the low discovery rate of short-read based insertion variant callers, Poster à la conférence de la société européenne degénétique humaine (ESHG). Berlin, 2020.

— Wesley Delage, J. T., Lemaitre, C. (2019). Comparison of variant callers for detection of large insertions in medical diagnosis context, Poster aux journées ouvertes de Biologie, Informatique etMathématique (JOBIM). Nantes, 2019.

Liste des figures

Liste des tableaux

Titre : Caractérisation et détection d'insertions constitutionnelles de grande taille dans le cadre d'un usage médical

Mot clés : Bioinformatique ; séquençage de génome ; détection d'insertions génomiques

Résumé : La détection de variations génétiques est un enjeu majeur dans le diagnostic des maladies génétiques chez l'homme. Certains types de variations sont détectés dans la routine d'analyse. D'autres, comme les variations de structure de type insertion sont bien plus complexes à identifier. Le développement de nouvelles technologies de séquençage dites longs reads permet de faciliter la détection de ces insertions. Elles ont notamment permis la génération d'ensembles de variants de référence d'une qualité sans précédent. Néanmoins, cette technologie possède encore des faiblesses qui ne permettent pas son utilisation pour la détection de variants dans un usage clinique. Il est donc essentiel d'améliorer les outils de détection basés sur les technologies de séquençage de courtes lectures utilisées dans un contexte médical. Cette thèse présente la caractérisation des différentes insertions et des facteurs limitant leur détection, basée sur ces jeux de données de référence de haute qualité. L'utilisation de simulations d'insertions a permis de quantifier l'impact de ces facteurs et mis en lumière la faiblesse des outils actuels à détecter et assembler la séquence des insertions. Ces résultats ont permis de proposer des pistes d'améliorations des outils de détection d'insertions. Plusieurs améliorations ont ainsi été implémentées dans l'outil existant MindTheGap et ont permis de surpasser certaines de ses limites.

Title: Characterization and detection of large constitutional insertions for medical use

Keywords: Bioinformatics; genome sequencing; detection of genomic insertions

Abstract: The detection of genetic variations is a major challenge in the diagnosis of human genetic diseases. Some types of variations are detected in the analysis routine. Others, such as insertion-type structural variations, are much more complex to identify. The development of new sequencing technologies known as long reads facilitates the detection of these insertions. In particular, they have made it possible to generate reference callsets with an unprecedented quality. Nevertheless, this technology still has weaknesses that make it impossible to use it for the variant calling in clinical use. It is therefore essential to improve detection tools based on short read sequencing technologies used in a medical context. This thesis presents the characterization the different insertions and the factors limiting their detection, based on these high quality reference callsets. The use of insertion simulations has allowed to quantify the impact of these factors and highlighted the weakness of current tools to detect and assemble the sequences of insertions. These results have allowed to propose ways to improve insertion detection tools. Several improvements have been implemented in the existing MindTheGap tool and have overcome some of its limitations.