Université de Rennes

École Doctorale MathStic – *Mention Informatique*

Mémoire d'Habilitation à Diriger des Recherches

---

# Méthodes bioinformatiques pour l'étude des Variants de Structure avec des données de séquençages génomiques

---


présentée par

***Claire Lemaitre***

Inria Rennes Bretagne Atlantique, UMR IRISA, équipe Genscale


HDR soutenue à Rennes le 2 décembre 2021, devant le jury composé de

| | | |
|---|---|---|
| **M. Cédric Chauve** | Professeur, Université Simon Fraser, Vancouver | Rapporteur |
| **M. Thierry Lecroq** | Professeur, Université de Rouen | Rapporteur |
| **Mme Hélène Touzet** | Directrice de Recherche, CNRS, Lille | Rapportrice |
| **M. Dominique Lavenier** | Directeur de Recherche, CNRS, Rennes | Président |
| **Mme Emmanuelle Lerat** | Chargée de Recherche, CNRS, Lyon | Examinatrice |
| **M. Denis Tagu** | Directeur de Recherche, INRAE, Rennes | Examinateur |

# Table des matières

**Préambule** **4**

**1 Contexte et problématiques** **6**
1.1 Les objets biologiques : les variants de structure . . . . . 6
1.1.1 Le génome et les variations génétiques . . . . . 6
1.1.2 Les différents types de variants de structure . . . . . 7
1.1.3 Pourquoi étudier les variants de structure ? . . . . . 7
1.2 Les données : données de séquençage et génomes de référence . . . . . 9
1.2.1 Le séquençage de l'ADN et ses différentes technologies . . . . . 9
1.2.2 Quelques exemples d'utilisation en génomique . . . . . 10
1.2.3 Assemblage et utilisation d'un génome de référence . . . . . 11
1.3 Les méthodes : état de l'art des méthodes de détection et d'analyse des variants de structure . . . . . 12
1.3.1 La base : l'alignement contre un génome de référence . . . . . 13
1.3.2 Différents signaux de mapping utilisés pour la détection . . . . . 13
1.3.3 Un problème difficile avec des lectures courtes . . . . . 15
1.3.4 Plus de 70 logiciels de détection pour les lectures courtes . . . . . 16
1.3.5 De nouvelles méthodes pour les lectures longues . . . . . 17
1.3.6 Après la découverte, les autres problèmes méthodologiques associés aux variants de structure . . . . . 17
1.4 Plan du manuscrit . . . . . 19

**2 Assemblage local de lectures courtes pour la détection d'insertions** **20**
2.1 Motivations et contexte . . . . . 20
2.1.1 Difficultés associées au type insertion . . . . . 20
2.1.2 État de l'art . . . . . 21
2.2 La méthode MindTheGap . . . . . 23
2.2.1 Deux étapes originales . . . . . 23
2.2.2 Nouvelles fonctionnalités postérieures à la publication . . . . . 25
2.3 Validation et applications de MindTheGap . . . . . 27
2.3.1 Validation de l'approche de détection d'insertions . . . . . 27
2.3.2 Applications sur des données réelles . . . . . 28
2.3.3 Passage à l'échelle sur un génome humain entier . . . . . 29
2.4 Conclusion . . . . . 31

**3 Vers une meilleure compréhension des faibles performances des outils avec des lectures courtes 32**
3.1 Motivations et contexte . . . 32
3.1.1 Apport des lectures longues pour la détection des variants de structure 32
3.1.2 2019-2020 : les premiers catalogues exhaustifs et précis chez l'homme . 33
3.2 Caractéristiques des vraies insertions chez l'homme . . . 34
3.2.1 Quatre niveaux de caractérisation des insertions . . . 34
3.2.2 Résultats : la majorité des insertions sont *difficiles* . . . 35
3.3 Identification des causes de la perte de rappel des outils de découverte . . . 37
3.3.1 Conception d'un benchmark à base de simulations . . . 37
3.3.2 Résultats . . . 38
3.3.3 Quelques leçons pour MindTheGap . . . 39
3.4 Conclusion . . . 40

**4 Génotypage des Variants de Structure avec des lectures longues 42**
4.1 Motivations et contexte . . . 42
4.1.1 Le problème du génotypage . . . 42
4.1.2 État de l'art : pas d'outil pour les lectures longues . . . 43
4.2 La méthode : SVJedi . . . 44
4.2.1 Principe et originalité . . . 44
4.2.2 Implémentation . . . 45
4.3 Résultats . . . 46
4.3.1 Données pour l'évaluation de la méthode . . . 46
4.3.2 SVJedi, une méthode efficace et robuste . . . 47
4.3.3 Comparaison avec d'autres approches . . . 48
4.4 Vers l'utilisation de graphes pour représenter les variants . . . 48
4.5 Conclusion . . . 49

**5 Discussion et perspectives 51**
5.1 Est-ce la fin des lectures courtes pour étudier les variants de structure ? . . . 51
5.2 Améliorer les méthodes de détection des variants de structure avec diverses données de séquençage . . . 53
5.2.1 Perspectives pour l'outil MindTheGap . . . 53
5.2.2 Developper des outils pour les données linked-reads . . . 53
5.2.3 Affiner les points de cassure avec des données de lectures longues . . . 55
5.3 Représentation et quantification des Variants de Structure dans les graphes de génome . . . 56
5.4 Applications et questions biologiques . . . 57

**Références bibliographiques 65**

**A Principales publications associées 66**
A.1 Publication 1 . . . 66
A.2 Publication 2 . . . 74
A.3 Publication 3 . . . 92

**Curriculum Vitæ 101**

# Préambule

Ce document présente une partie de mes travaux de recherche que j'ai effectués au sein de l'équipe Genscale, équipe de bioinformatique commune au centre Inria Rennes Bretagne Atlantique et au laboratoire d'informatique IRISA à Rennes. Mes travaux de recherche se situent à l'interface entre plusieurs disciplines : l'informatique, les mathématiques et la biologie. Cette pluri-disciplinarité est présente dès ma formation universitaire initiale et se retrouve tout au long de mon parcours professionnel, notamment dans les différents environnements de recherche dans lesquels j'ai travaillé : un laboratoire de biologie en thèse, une plateforme de services de bioinformatique en post-doctorat et un laboratoire d'informatique pour mon poste actuel de Chargée de Recherche. Mon activité consiste à développer et utiliser des méthodes informatiques pour répondre à des questions biologiques. Depuis mes premiers travaux en thèse, mes objets biologiques d'étude sont les génomes des organismes vivants et plus généralement les séquences d'ADN. Les questions méthodologiques portent sur leur analyse et leur comparaison.

Durant mes doctorat et post-doctorat, entre 2005 et 2010, les données de génomiques étaient sous la forme de quelques génomes complets, avec une grande qualité de séquence et d'annotation. Je me suis intéressée à l'évolution des génomes d'organismes divers, des mammifères aux bactéries, j'ai développé des méthodes permettant de les comparer et j'ai également analysé les résultats biologiques de ces comparaisons. Mon arrivée, en 2010, en tant que Chargée de Recherche au centre Inria Rennes Bretagne Atlantique a coïncidé avec l'apparition de nouvelles technologies de séquençage à haut débit qui ont posé des problèmes algorithmiques nouveaux pour traiter et comparer une quantité beaucoup plus importante de données génomiques. Mon activité de recherche s'est naturellement orientée vers ces nouveaux problèmes. Les questions biologiques sont similaires, il s'agit toujours d'identifier ce qui est commun et ce qui diffère entre deux ou plusieurs génomes. Mais les méthodes pour y répondre doivent s'adapter à deux différences majeures des données : leur qualité et leur quantité. D'une part, l'information est beaucoup plus morcelée et plus bruitée, ce qui nécessite de développer des méthodes spécifiques de pré-traitement et d'analyse des données. D'autre part, ces données sont beaucoup plus massives et nécessitent de développer de nouvelles heuristiques plus rapides et/ou de nouvelles structures d'indexation plus légères en mémoire. Ainsi, depuis 2010, mes travaux de recherche ont porté sur une grande diversité de problèmes relatifs au traitement et l'analyse des données de séquençage à haut débit : la correction des erreurs de séquençage, la compression des fichiers de séquençage, l'assemblage de génomes, la détection de variations génétiques, ponctuelles ou plus complexes, la comparaison massive de séquençages métagénomiques, etc.

J'ai choisi de focaliser ce document sur une partie seulement de ces travaux, ceux relatifs à une problématique biologique qui m'intéresse particulièrement : les variations de structure dans les génomes. Mon intérêt pour ces variants en particulier remonte à mes tout premiers travaux de recherche, en thèse. La question qui m'a passionnée durant ma thèse est celle de

l'organisation des génomes et l'évolution de leur structure : lorsqu'on compare des génomes d'espèces proches, on observe un contenu en séquence relativement similaire mais l'ordre et l'orientation de ces séquences le long des génomes peuvent être très différents. Ces modifications d'organisation génomique sont générées par des réarrangements chromosomiques, appelés également des variants de structure à l'échelle intra-spécifique. Comprendre les mécanismes et les impacts de ces évènements sont des questions biologiques fondamentales mais qui nécessitent dans un premier temps d'être capable de détecter et caractériser ces variations dans les génomes. Cette problématique méthodologique est devenu majoritaire dans mes activités de recherche ces dernières années et constitue mon projet de recherche pour les années à venir. C'est pourquoi, j'ai fait ce choix de centrer ce document sur cette thématique. Cependant, mes autres travaux sur les données de séquençages ne sont pas complètement absents de ce document, puisqu'ils se basent notamment sur des concepts et des outils algorithmiques communs (par exemple, les représentations par ensembles de $k$-mers et les graphes de séquences).

Ce document est organisé en 5 chapitres, le premier présente le contexte biologique et les problématiques méthodologiques de mes travaux, les trois suivants offrent une synthèse de mes principales contributions pour l'étude des variants de structure avec des données de séquençage, et le dernier chapitre expose mes perspectives de recherche. Les trois publications principales sur lesquelles se basent ce document sont ajoutées en annexe du document. Enfin, un CV présentant les différents éléments nécessaires à l'établissement du dossier d'HDR clôture ce document.

# Chapitre 1

# Contexte et problématiques

Ce chapitre présente le contexte scientifique dans lequel s'inscrivent les travaux de recherche présentés dans ce document. Nous définissons dans un premier temps les objets biologiques qui nous intéressent, les génomes et leurs variants de structure. Puis, nous décrivons les données à disposition pour étudier ces objets et qui composent l'entrée de nos méthodes, avant de présenter les problématiques méthodologiques et un état de l'art des méthodes pour détecter et étudier les variants de structure dans les génomes.

## 1.1 Les objets biologiques : les variants de structure

### 1.1.1 Le génome et les variations génétiques

La molécule d'ADN est le support de l'information génétique pour la très grande majorité des organismes vivants. D'un point de vue moléculaire, c'est une chaîne orientée de nucléotides liés par des liaisons covalentes. L'ADN contient 4 nucléotides différents A, C, G et T. La succession des différents nucléotides le long d'un brin d'ADN forme une séquence orientée. D'un point de vue informatique, une molécule d'ADN est vue comme un texte construit sur un alphabet à 4 lettres, on parle d'une *séquence* d'ADN. Le génome d'un organisme est l'ensemble de ces textes, que l'on retrouve presque à l'identique dans chacune de ces cellules. L'information qu'il contient permet le développement et le fonctionnement de l'organisme. Le génome est répliqué et transmis aux descendants. Étant donné l'importance de l'information génétique pour le fonctionnement des organismes, la réplication de l'ADN et sa transmission sont des mécanismes très fidèles. Ainsi la séquence du génome d'une espèce est relativement stable au cours du temps (par exemple seulement 1 % des nucléotides dans les séquences des gènes sont différents entre l'homme et le chimpanzée alors que les deux espèces ont divergé depuis environ 6 millions d'années).

Les génomes subissent cependant constamment des mutations par divers mécanismes dont certaines échappent aux mécanismes de réparation de l'ADN. Ces mutations, si elles se produisent dans les cellules de la lignée germinale et si elles sont transmises aux descendants, pourront éventuellement se répandre dans la population au fil des générations. Avant qu'elles ne soient complètement fixées ou au contraire complètement éliminées dans la population par la sélection naturelle, on peut les retrouver à l'état polymorphe et elles sont la principale source de la diversité phénotypique des individus au sein d'une même espèce. Ces différences qu'on peut observer entre les génomes d'individus appartenant à une même espèce sont alors appelées des *variations génétiques*.

Il existe différents types de variations génétiques. On les distingue principalement par leur taille. Les plus petites sont les variations dites ponctuelles qui ne touchent qu'une seule position du génome isolée : ce sont les substitutions d'un nucléotide par un autre (SNP ou SNV), les délétions d'un nucléotide ou les insertions d'un nucléotide à une position donnée. Les délétions et insertions d'un ou plusieurs nucléotides successifs sont également appelés *petits indels*. Enfin, les plus grandes variations génétiques sont regroupées sous le terme de *Variations Structurales* (ou *variant de structure*) et forment un groupe très hétérogène en termes de type de mutation et de taille, allant généralement de 50 pb à des bras chromosomiques ou chromosomes entiers. En terme de fréquence dans les génomes, plus les variations sont petites, plus elles sont fréquentes. Ainsi, chez l'homme, sur les 3 milliards de pb qui composent son génome, on estime le nombre de SNPs de l'ordre du million, dix fois moins d'indels et 100 fois moins de variants de structure ($> 50$ pb). Cependant, comme les variants de structure portent sur des segments d'ADN plus longs, ils touchent globalement une proportion plus importante du génome. Ainsi, on estime que la diversité génomique entre deux individus humains est en moyenne de 0,1 % lorsqu'on ne considère que les mutations ponctuelles, mais elle augmente à 1,5% si on tient compte des variants de structure (Pang et al., 2010).

### 1.1.2 Les différents types de variants de structure

La définition communément utilisée d'un variant de structure, est une mutation d'un segment d'ADN d'une taille supérieure à 50 pb et qui peut-être déplacé, supprimé ou ajouté dans le génome.

Il est intéressant de s'attarder un moment sur ce seuil de taille de 50 pb. Il est tout à fait arbitraire et n'a pas de signification biologique par rapport à d'éventuels mécanismes moléculaires. Il a été arbitré à partir de considérations méthodologiques de détection de ces variations. De même, le seuil de taille pour être considéré comme un indel peut varier selon les définitions, mais il est généralement autour de 10 à 20 pb. Ces deux seuils laissent donc un certain flou sur la dénomination de variations d'une taille comprise entre 10 et 50 pb.

Le terme Variations Structurales regroupe un grand nombre de types différents, les principaux sont représentés dans la Figure 1.1. On peut tout d'abord distinguer les variants équilibrés des variants déséquilibrés. Les variants équilibrés ne modifient pas la quantité d'ADN et ne font que déplacer des segments d'ADN existants. On y trouve les inversions, les translocations réciproques et les modifications de type *couper-coller*, qui sont parfois appelées des transpositions. Parmi les variants déséquilibrés, les délétions réduisent la quantité d'ADN en supprimant des segments génomiques, alors que les insertions et duplications ajoutent des segments d'ADN, qu'ils soient nouveaux ou issus d'une duplication d'un segment déjà existant dans le génome. Les délétions et duplications peuvent également être qualifiées de variants de nombre de copies (CNV, pour *copy number variant*), le plus souvent lorsque leur taille est supérieure à 1 Kb, là encore cette définition relève de considérations méthodologiques de détection plutôt que biologiques.

### 1.1.3 Pourquoi étudier les variants de structure ?

Au même titre que les SNPs, les variants de structure étant des modifications du génome, ils peuvent avoir un impact fonctionnel important pour la cellule et l'organisme. Cependant, contrairement aux SNPs, la prédiction de cet impact peut être plus ardue : pour évaluer l'impact d'un SNP, on regarde s'il est localisé dans un élément fonctionnel (séquence codante

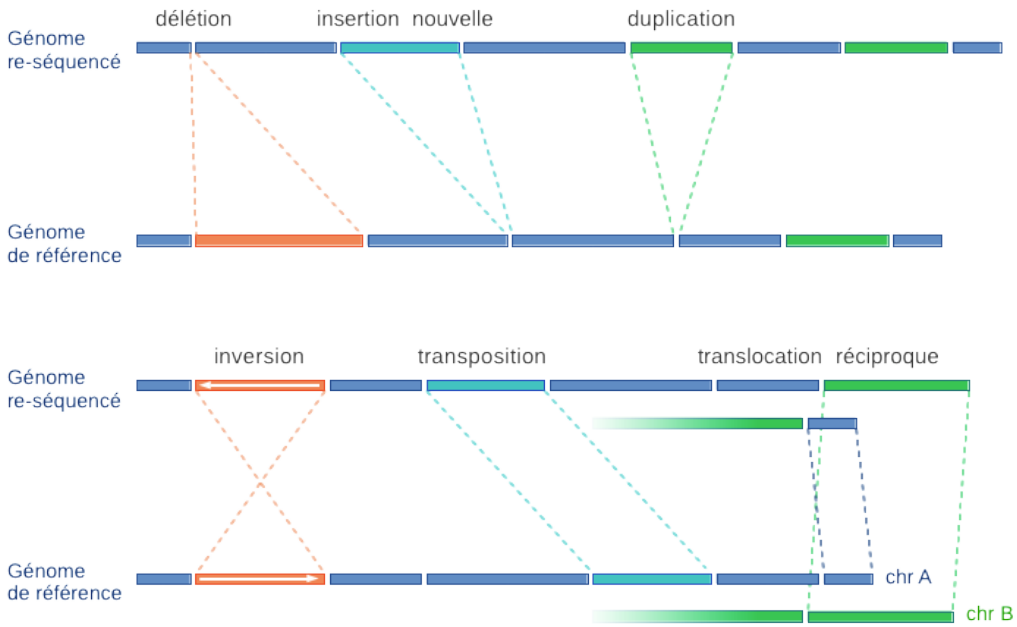


FIGURE 1.1 – Représentation schématique des principaux types de variants de structure, en haut ceux qui modifient le nombre de copies (variants déséquilibrés), en bas ceux qui ne font que déplacer des segments d'ADN (variants équilibrés).

ou régulatrice par exemple) et si c'est le cas on évalue si la modification ponctuelle modifie ou supprime la fonction de cet élément. Un réarrangement touche une région du génome plus grande qui peut contenir plus d'un élément fonctionnel dont les séquences peuvent ne pas être altérées mais seulement déplacées ou leur nombre de copies altéré. Ainsi, plusieurs types d'impact sont possibles, comme par exemple :

- la suppression définitive d'un gène ou élément fonctionnel par une délétion,
- la perte d'un gène par l'éclatement de sa séquence dans différents *loci* lorsqu'un point de cassure d'un grand réarrangement tombe dans celui-ci (inversion, translocation, insertion),
- la modification de l'expression d'un gène par une modification de son nombre de copies (délétion ou duplication),
- la modification de la régulation de l'expression d'un gène si son environnement génomique change (par exemple déplacement dans une région avec un statut de chromatine ouvert/fermé différent),
- etc.

Ainsi, les exemples (et les références) ne manquent pas démontrant l'implication de variations structurales dans des variations de traits phénotypiques d'intérêt (par exemple chez les plantes lire la revue de Gabur et al. (2019)), dans les phénomènes d'adaptation des espèces (par exemple, lire l'étude à grande échelle chez l'homme de Yan et al. (2021)) ou encore dans des maladies génétiques chez l'homme (lire la revue de Weischenfeldt et al. (2013)). De nombreuses références sont également présentes dans l'introduction du papier de revue de Mahmoud et al. (2019).

Certains réarrangements jouent également un rôle dans la spéciation et l'évolution des espèces. C'est le cas notamment des grandes inversions, qui en réduisant la recombinaison

homologue dans le segment inversé, peuvent générer une barrière reproductive et conduire à la spéciation des espèces, être associées à de l'adaptation locale, ou encore favoriser la divergence entre chromosomes sexuels (lire par exemple (Kirkpatrick, 2010) ou encore le numéro spécial de la revue Molecular Ecology (Wellenreuther et al., 2019)).

Malgré ces nombreux impacts, les variants de structure ont été largement moins étudiés que les variations ponctuelles et cela est principalement du à des difficultés techniques pour les détecter dans les données génomiques.

## 1.2 Les données : données de séquençage et génomes de référence

Les données actuelles pour étudier les variants de structure proviennent du séquençage de génomes. Le séquençage d'un fragment d'ADN consiste à déterminer sa séquence, c'est-à-dire la succession des nucléotides, parmi l'alphabet $\{A, C, G, T\}$.

### 1.2.1 Le séquençage de l'ADN et ses différentes technologies

Actuellement, aucune technologie de séquençage n'est capable de séquencer de manière contigüe une molécule aussi longue qu'un chromosome humain par exemple. Les molécules d'ADN sont donc fragmentées en fragments plus petits avant d'être séquencées. Le séquençage d'un génome complet repose principalement sur la stratégie appelée le séquençage *shotgun*, qui consiste à fragmenter aléatoirement un grand nombre de molécules d'ADN provenant du même génome. Après séquençage des différents fragments, on obtient donc un ensemble de séquences, appelées *lectures de séquençage*, qui sont échantillonnées aléatoirement le long du génome ciblé, de telle sorte qu'une même position génomique peut être couverte par plusieurs lectures différentes. Le nombre moyen de lectures qui couvrent une position donnée du génome est appelée la *profondeur de séquençage*. Les caractéristiques, et notamment la taille, de ces lectures dépendent de la technologie de séquençage utilisée.

Depuis la découverte de la molécule d'ADN dans les années cinquante, trois générations de technologies de séquençage se sont succédées. Elles se différencient par leur coût, leur débit et leurs caractéristiques des lectures produites en termes de taille et d'erreurs de séquençage. La première, dont la technologie la plus connue est la technologie SANGER, date des années soixante dix, produit des lectures de taille environ 1 000 pb avec un taux d'erreur très faible ($<0.1\%$), pour un coût et un temps assez élevé.

La deuxième génération de technologie de séquençage, aussi appelée NGS pour *Next Generation Sequencing*, est apparue dans les années 2005-2008, la plus connue et encore largement utilisée étant la technologie Illumina. Cela a été une révolution en génomique car les débits ont été massivement augmentés pour un coût beaucoup plus faible, on parle aussi de *séquençage à haut débit*. La contre-partie de cette technologie est qu'elle produit des lectures beaucoup plus petites (100 à 250 pb). Son taux d'erreurs de séquençage est assez faible, autour de 1 % initialement mais plus proche de 0,1% actuellement. Une caractéristique de la technologie Illumina est qu'elle peut produire des lectures appariées ou pairées (*paired-end* en anglais) : les fragments d'ADN de taille contrainte autour de 300 voire 500 pb sont séquencés depuis les deux extrémités dans des directions opposées, produisant deux lectures appariées.

Enfin, la troisième génération est apparue dans les années 2015 et est toujours en développement, avec les technologies des entreprises Pacific Biosciences (PacBio) et Oxford

Nanopore (ONT). Cette génération se démarque des NGS car elle permet le séquençage direct de grandes molécules, produisant des lectures de tailles allant de 10 Kb à plusieurs Mb. Leur principal inconvénient est leur plus fort taux d'erreurs, initialement autour de 20 à 30 % mais qui tombe actuellement en dessous de 13 %, jusqu'à 1 % pour les données PacBio HiFi. Le type des erreurs est également différent par rapport aux autres générations : il s'agit principalement d'erreurs d'insertions et de délétions. Enfin, actuellement, leur débit est plus faible et leur coût plus élevé que la technologie Illumina (voir Table 1 [1] de la revue de Logsdon et al. (2020), pour plus de détails sur les technologies les plus récentes, leurs caractéristiques et leur coût). Classiquement, on distingue ces deux dernières générations par les termes *lectures courtes* versus *lectures longues*.

Un type intermédiaire de données de séquençage, entre les lectures courtes et les lectures longues, sont les lectures liées ou lectures longues synthétiques (nous utiliserons le terme anglais plus commun *linked-reads*). Ce type de données est apparu en même temps que la troisième génération de séquençage et a été popularisé par l'entreprise 10X Chromium Genomics. Ce n'est pas une technologie de séquençage à part entière, l'innovation réside dans la préparation des librairies des fragments d'ADN avant le processus de séquençage. Le principe est de marquer des longues molécules, typiquement de taille 50 Kb, par des barcodes moléculaires (courtes séquences de 10 à 20 pb) avant de les séquencer avec la technologie Illumina. Le résultat est un ensemble de lectures courtes, dont le début de la séquence correspond au barcode. Les lectures issues d'une même molécule possèdent le même barcode et on en tire une information longue distance. En 2020, l'entreprise 10X Chromium Genomics a arrêté les ventes de ce produit pour des conflits de propriété intellectuelle, mais depuis 2019, d'autres technologies du même type ont été développées, telles que TELL-seq (Chen et al., 2020), stLFR (Wang et al., 2019) et Haplotagging (Meier et al., 2021).

### 1.2.2 Quelques exemples d'utilisation en génomique

La première génération de technologie de séquençage a permis d'obtenir les séquences des premiers génomes d'organismes modèles : d'abord des virus, des bactéries, puis à partir des années 1990 quelques génomes d'eucaryotes, organismes modèles. En particulier, le projet du séquençage du génome humain a débuté en 1990 et s'est terminé en 2003, il a mobilisé de nombreuses équipes dans le monde en entier et a coûté près de 3 milliards de dollars. Il a abouti notamment en 2001 à la publication du premier assemblage de génome humain (Lander et al., 2001). À partir des années 2008, les NGS ont ensuite permis de compléter notre connaissance des génomes d'organismes vivants en augmentant considérablement le nombre de génomes d'espèces différentes séquencés. Du fait de la baisse importante des coûts, la génération de génomes de référence n'était plus limité aux seules espèces modèles ou à fort intérêt économique. Ainsi, plusieurs grands projets internationaux de séquençage ont été lancés pour augmenter les ressources génomiques dans tous les domaines du vivant, comme par exemple les lancements en 2008 du projet 1001 génomes de plantes et du projet du séquençage du microbiote humain (HMP), et en 2011 du projet i5K de séquençage de 5 000 génomes d'arthropodes.

Les NGS ont également permis de séquencer plusieurs individus pour une même espèce et donc d'accéder aux variations génomiques dans les populations à l'échelle des génomes entiers. Ainsi, en 2008 a été lancé le projet international 1 000 génomes qui avait pour objectif d'établir un catalogue approfondi de la diversité génétique au sein des populations humaines.

1. `https://www.nature.com/articles/s41576-020-0236-x/tables/1`

Ce projet compte actuellement plus de 3 000 génomes humains séquencés. D'autres initiatives centrées sur des populations en particulier ont vu le jour par la suite, comme par exemple UK10K au Royaume-Unis (10 000 génomes) ou encore GoNL aux Pays Bas (1 000 génomes).

### 1.2.3 Assemblage et utilisation d'un génome de référence

De part leur petite taille, les lectures de séquençage ne représentent chacune qu'une information très partielle du génome. Des méthodes bioinformatiques sont nécessaires pour exploiter ces données et en extraire de l'information biologique.

Le premier traitement est celui de l'assemblage des séquences. Il consiste à reconstruire à partir d'un ensemble de petites séquences, les lectures, une plus grande séquence dont elles sont issues, idéalement les chromosomes en entier. Pour cela, on exploite le fait qu'il y a de la redondance dans les données, chaque position du génome étant couverte par plusieurs lectures différentes. Ainsi, les lectures de séquençage se chevauchent deux à deux et de proche en proche on peut reconstruire de plus longues séquences. C'est néanmoins un problème très difficile. La principale difficulté vient de la nature très répétée des génomes. Si une séquence de taille plus grande que les lectures est présente à l'identique à plus d'une position dans le génome, plusieurs reconstructions du génome sont possibles. Avec l'augmentation du nombre de séquences répétées et du nombre de leurs copies, le nombre de solutions possibles explose rapidement de manière combinatoire. À cela s'ajoutent d'autres difficultés comme les erreurs de séquençage dans les données, le polymorphisme dans les données (par exemple, pour un organisme diploïde, un certain nombre de sites sont hétérozygotes), l'hétérogénéité de couverture le long du génome (certaines régions peuvent être très peu représentées dans les données de séquençage), le fait que les lectures sont séquencées à partir des deux brins de l'ADN, etc.

Le problème de l'assemblage de génome est un problème algorithmique qui date des premiers séquençages d'ADN et qui occupe encore à l'heure actuelle de nombreuses recherches méthodologiques. Plusieurs approches ont été proposées au cours du temps, qui se sont adaptées aux caractéristiques des différentes technologies de séquençage. Ces méthodes sont basées principalement sur une première étape de construction d'un graphe qui représente les chevauchements de séquences, puis une étape de recherche de chemins dans ce graphe. Deux types de graphes sont principalement utilisés : le graphe de chevauchements de lectures et le graphe de de Bruijn. Dans le premier, chaque sommet est une lecture de séquençage, les arcs sont des chevauchements inexacts entre les lectures. Dans le graphe de de Bruijn, chaque sommet est un mot de taille $k$ et un arc relie un sommet à un autre si le suffixe de taille $k-1$ du premier est exactement le préfixe du deuxième. Le graphe de chevauchements est principalement utilisé pour les lectures longues (première et troisième génération de séquençage). Le graphe de de Bruijn, quant à lui, est principalement utilisé avec des lectures courtes, pour des raisons de passage à l'échelle : la construction du graphe de chevauchements est très coûteuse en temps de calcul et en mémoire pour des très grands nombres de lectures, comme c'est le cas des données NGS qui possèdent typiquement plusieurs centaines de milliers voire des milliards de lectures pour un génome humain.

Grâce à ces méthodes, et souvent grâce à d'autres types de données (cartes génétiques, physiques), on a généré des génomes pour de nombreux organismes. Pour chaque espèce, on choisit généralement un génome en particulier qu'on appelle le *génome de référence*, qui constitue une base commune pour la communauté scientifique pour pouvoir ensuite étudier son contenu et ses variations dans les populations de manière standardisée et comparable. Ainsi, sur ce génome, on concentre beaucoup d'efforts pour obtenir la séquence la plus

complète et contigüe possible. L'annotation du génome consiste ensuite à localiser et décrire ses éléments fonctionnels tels que les gènes, les séquences régulatrices, les éléments répétés.

Si on prend l'exemple du génome humain, suite à la publication en 2001 de la première version du génome de référence, de nombreux efforts internationaux ont permis d'améliorer à la fois sa séquence et son annotation. La version de 2001 ne représentait que la partie euchromatique (sans les télémoères et centromères), soit environ 92 % du génome, et c'est seulement en 2021 que l'intégralité des chromosomes a été assemblée sans aucun trou par le consortium T2T (Telomere-to-telomere) (Nurk et al., 2021). Du côté de l'annotation, environ 20 000 gènes ont été identifiés dans la version de 2001. Cependant, les séquences codantes ne représentent que 1,5 % des 3 milliards de paires de bases du génome et le projet ENCODE, en particulier, a permis d'améliorer l'annotation des éléments fonctionnels non codants. Enfin, il est important de noter que près de la moitié du génome est composé d'éléments répétés, pour la plupart des éléments transposables répétés en un grand nombre de copies.

Lorsqu'on séquence plusieurs individus d'une même espèce et qu'on possède déjà un génome de référence pour cette espèce, on parle alors de *re-séquençage*. L'objectif de tels séquençages est souvent d'identifier les variations génomiques entre les différents individus re-séquencés. Pour cela la stratégie usuelle est d'utiliser le génome de référence comme base de comparaison, avec une première étape de *mapping des lectures* sur le génome de référence. Le mapping consiste à identifier d'une part la position génomique d'où provient une lecture donnée et d'autre part les différences s'il y en a entre la séquence de la lecture et sa version dans le génome de référence. Cette tâche s'effectue par *alignement de séquences*, un autre problème algorithmique fondamental de la bioinformatique. Là encore les algorithmes développés dans les années 70-80 pour les premières données de séquences ont du être adaptés pour faire face à la grande quantité de données à traiter. Dans le cas du mapping de lectures de séquençage sur un génome de référence, la problématique majeure est de réduire le temps de calcul et la consommation mémoire grâce à des structures de données d'indexation des séquences.

L'étape suivant l'alignement de toutes les lectures sur le génome de référence s'appelle l'appel des variants (*variant calling* en anglais). Elle diffère selon le type de variants recherché. Pour les mutations ponctuelles comme les SNPs et les petits indels, cela consiste principalement à parcourir le génome de référence et à identifier des positions pour lesquelles un nombre significatif d'alignements indiquent une même différence. Pour les variants de structure, nous verrons dans la Section suivante que les stratégies sont plus diverses et complexes.

## 1.3 Les méthodes : état de l'art des méthodes de détection et d'analyse des variants de structure

Dans cette section, nous présentons les différentes approches, ainsi que quelques outils de l'état de l'art, pour détecter et analyser les variants de structure lorsqu'on dispose de données de séquençage de deuxième et troisième générations et d'un génome de référence (lire également l'excellent article de revue de Mahmoud et al. (2019)). Avant l'arrivée des NGS, les variants de structure étaient relativement peu étudiés car il y avait peu de données de re-séquençage. Des approches moléculaires basées sur l'hybridation d'ADN, telles que le FISH ou les puces CGH, permettaient d'identifier certains types de variants de structure mais à des résolutions très faibles (de l'ordre de la centaine de Kb au Mb). Les données de séquençage haut débit permettent en théorie d'accéder à l'ensemble des variants de structure

à une très forte résolution, jusqu'au nucléotide près.

### 1.3.1 La base : l'alignement contre un génome de référence

Une première idée naïve, lorsqu'on veut identifier les variants de structure d'un génome re-séquencé par rapport à un génome de référence, est d'abord d'obtenir l'assemblage complet du nouveau génome, puis de l'aligner sur le génome de référence pour identifier toutes les différences. Cela parait judicieux d'avoir une vue globale du génome re-séquencé pour identifier et analyser des grands remaniements. De plus, avant l'arrivée des NGS, des méthodes d'alignements de génomes complets avaient été développées pour comparer des génomes d'espèces différentes (domaine de la génomique comparée). Cependant, ce n'est pas l'approche qui a été retenue pour la raison principale que l'assemblage *de novo* de génomes est une tâche complexe, très coûteuse en ressources de calcul et avec des résultats de faible qualité notamment lorsqu'il est réalisé avec des lectures courtes. De plus, le nombre de variants de structure attendus dans un génome re-séquencé étant limité, la très grande majorité du génome ne devrait pas différer en structure du génome de référence, ainsi la majorité du temps et d'efforts alloués au ré-assemblage serait alors gaspillée pour des régions sans intérêt. Un autre inconvénient de cette approche est la non-détection d'une partie des variants présents à l'état hétérozygote chez l'individu diploïde re-séquencé, puisque l'assemblage ne représente qu'un seul des deux haplotypes. Des approches d'assemblage des deux haplotypes sont maintenant développées en particulier avec les données de séquençage de troisième génération, mais la complexité du problème n'en est que plus importante.

En conséquence, la majorité des approches de détection des variants de structure se base sur une représentation locale du génome re-séquencé par l'analyse des lectures mappées sur le génome de référence. Ainsi, une fois que toutes les lectures sont positionnées sur le génome de référence, il suffit de parcourir ce dernier pour identifier des positions ou régions du génome avec des différences de séquence observées dans un nombre significatif de lectures. Ce principe est très efficace pour la détection de petits variants, tels les SNPs et petits indels, car les différences sont facilement identifiables au sein de chaque alignement. La difficulté est alors de distinguer les différences réelles de celles dues à des erreurs de séquençage. Lorsque le taux d'erreur est faible et uniforme le long du génome, comme c'est le cas pour le séquençage de deuxième génération (lectures courtes), le problème est considéré comme résolu. À l'inverse, le problème est bien plus complexe pour les variants de structure et en particulier avec les lectures courtes, car les variants recherchés sont bien souvent plus grands que la taille des lectures et également car les réarrangements génomiques ont tendance à être co-localisés avec les régions répétées du génome, des régions où il est plus difficile de mapper les lectures correctement et de manière non-ambigüe (plusieurs positions de mapping possibles pour une même lecture).

### 1.3.2 Différents signaux de mapping utilisés pour la détection

Même si la plupart des variants de structure sont trop grands pour être identifiés et caractérisés directement au sein d'un alignement d'une seule lecture courte, contrairement à un SNP, leur présence génère certains signaux de mapping anormaux qui les différencient de régions sans différence structurale. Trois types de signaux peuvent être observés : l'alignement d'une lecture en deux morceaux (split-read), la discordance de mapping entre lectures pairées, et la profondeur locale de séquençage. Ils sont représentés schématiquement dans le cas d'une délétion dans la Figure 1.2.

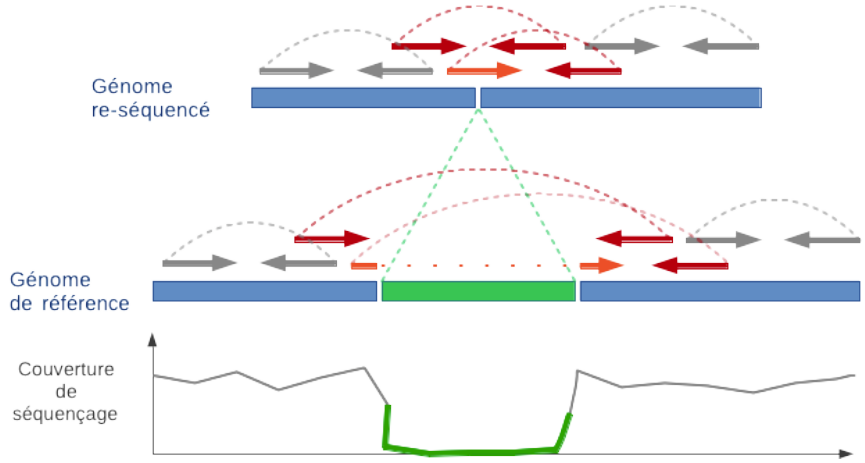


FIGURE 1.2 – Les trois types de signaux de mapping aberrants sont représentés dans le cas d'une délétion d'un segment (en vert) dans le génome re-séquencé par rapport au génome de référence. Les lectures de séquençage sont représentées par des flèches, comme elles sont échantillonnées sur le génome re-séquencé et comme elles sont mappées sur le génome de référence, et leur appariement est représenté par un arc en pointillé. Les lectures grises sont concordantes et n'indiquent aucune variation de structure. La paire de lectures rouges est discordante car la distance de mapping sur le génome de référence est plus grande qu'attendu. La lecture orange est mappée en deux morceaux de part et d'autre du segment supprimé (split-read). En bas, la variation de la couverture de séquençage est représentée avec une chute au niveau du segment vert indiquant l'absence de cette séquence dans le génome re-séquencé.

Les deux premiers signaux sont caractéristiques de *points de cassure*, ce sont des adjacences de régions génomiques différentes entre le génome re-séquencé et le génome de référence. Dans le premier cas, d'une lecture alignée en deux morceaux, le début et la fin de la lecture sont adjacents dans le génome re-séquencé mais distants ou orientés différemment dans le génome de référence produisant deux alignements distincts pour ces deux parties de la lecture. Par exemple, dans le cas d'une délétion d'un segment génomique, une lecture séquencée à cheval sur le point de cassure de la délétion dans le génome re-séquencé, sera alignée sur le génome de référence en deux alignements de part et d'autre du segment supprimé.

Le deuxième type de signal suit le même principe mais à l'échelle du fragment de séquençage, représenté par deux lectures appariées (voir Section 1.2.1 p. 9), plutôt que de la lecture individuelle. La taille des fragments étant contrainte par la technologie, la distance sur le génome re-séquencé entre deux lectures provenant d'un même fragment suit une distribution particulière. Ainsi, lorsque deux lectures issues d'un même fragment ne mappent pas sur le génome de référence de façon à former un fragment d'une taille raisonnable, c'est-à-dire que leur distance de mapping ne suit pas la distribution attendue ou leur orientation relative n'est pas compatible avec un fragment séquencé aux deux extrémités (de façon tête-bêche), on dit qu'elles sont *discordantes* (voir une exemple dans la Figure 1.2). Cette discordance de mapping indique là encore une adjacence de séquences dans le génome re-séquencé qui n'existe pas dans le génome de référence. Si on reprend l'exemple de la délétion, une paire de lectures avec une lecture de part et d'autre du point de cassure sur le génome re-séquencé, donnera un mapping discordant car les deux lectures seront alignées de part et d'autre du

segment supprimé sur le génome de référence, donc à une distance plus grande qu'attendue. On distingue plusieurs types de discordances en fonction de la distance et de l'orientation relative des deux lectures, qui sont générées par les différents types de variants de structure mais ne leur sont pas toujours spécifiques.

Enfin, le dernier signal n'est généré que par certains types de variants de structure, les délétions et les duplications. Ces variants modifient le nombre de copies d'une région génomique donnée. La couverture de séquençage, ou le nombre de lectures mappées couvrant une position donnée du génome, est un proxy pour estimer ce nombre de copies. Les régions du génomes ayant subi une diminution ou une augmentation de leur nombre de copies présentent une couverture de séquençage qui ne suit pas la distribution observée sur le reste du génome. Dans le cas d'une délétion, à l'état homozygote, le signal est encore plus fort, puisqu'on observe l'absence de lectures mappées dans cette région.

### 1.3.3 Un problème difficile avec des lectures courtes

Ces différents signaux indiquent une région sur le génome de référence dont la structure est différente dans le génome re-séquencé, mais ils ne permettent pas toujours de caractériser précisément le variant présent, c'est-à-dire de définir précisément son type, la séquence de l'allèle alternatif et la ou les positions au nucléotide près du ou des points de cassure. Par exemple, dans le cas d'une insertion d'une nouvelle séquence (c'est-à-dire une séquence absente du génome de référence), la présence de paires de lectures dont une des deux lectures ne s'aligne pas du tout sur le génome de référence permet de localiser le site d'insertion mais ne permet pas de caractériser la nouvelle séquence insérée. Une augmentation de la couverture de séquençage indique la duplication d'un segment mais ne permet pas d'identifier à quel *locus* dans le génome cette duplication s'est insérée. De plus, pour les variants de structure qui possèdent plusieurs points de cassure, comme c'est aussi le cas pour les inversions, les translocations réciproques ou encore les transpositions de type "couper-coller", il faut combiner des signaux de différents types et provenant de différentes régions du génome de référence pour reconstruire l'évènement de réarrangement. Enfin, ce qui complique encore plus la tâche est qu'il arrive souvent qu'un même signal de mapping anormal puisse être généré par plusieurs types différents de variants de structure. Par exemple, deux lectures appariées mappées à une trop grande distance peuvent indiquer une délétion ou une duplication. De la même façon deux lectures appariées mappées avec la même orientation peuvent indiquer aussi bien une inversion qu'une duplication insérée sur le brin inverse.

À ces difficultés théoriques, s'ajoutent des difficultés pratiques liées au fait que les données de mapping en entrée ne sont pas sans erreurs et sans ambiguïtés : les lectures ne sont pas toujours alignées à leur *locus* d'origine et certaines lectures ont plusieurs positions de mapping équivalentes (mapping ambigu). Ces erreurs et imprécisions proviennent de deux éléments, d'une part le fait que les algorithmes de mapping sont des heuristiques qui peuvent parfois privilégier le temps de calcul au détriment de la sensibilité et de la précision des résultats, et d'autre part la nature très répétée des génomes. Une lecture provenant d'une région répétée plus grande que la lecture sera plus difficile (voire impossible si les copies sont identiques) à localiser de manière non ambiguë entre les différentes copies de la répétition. On peut par exemple pré-calculer sur le génome de référence les régions avec une bonne *mappabilité*, c'est-à-dire pour une taille de lecture donnée, les régions où les lectures peuvent être mappées de manière non ambiguë. Pour le génome humain, qui est composé à 45 % de séquences répétées, pour des lectures de taille 100, la proportion du génome correctement *mappable* est autour de 90 % (Lee and Schatz, 2012). La propor-

tion non mappable peut paraître négligeable, et elle l'est dans d'autres contextes, comme la recherche de mutations ponctuelles impactant les portions codantes du génomes (fraction mappable de 97.4 %), mais dans le cas des variations structurales, même une petite fraction de répétitions peut générer un grand nombre de fausses détections (faux positifs). Et si on décide de simplement ne pas regarder ces régions, on réduit fortement la sensibilité de détection, car les variants de structure ont tendance à être associés aux éléments répétés. D'une part, les répétitions sont souvent impliquées dans les mécanismes de génération des variants de structure (par exemple par recombinaison homologue non allélique), et d'autre part une grande partie des variants de structure sont eux-mêmes des éléments répétés (les duplications, les transpositions d'éléments transposables). Plus la lecture est petite, plus la probabilité d'erreur de mapping ou de mapping ambigu est grande. Cela explique pourquoi l'alignement par morceaux (split-mapping) est complexe et limité pour les lectures courtes.

### 1.3.4 Plus de 70 logiciels de détection pour les lectures courtes

À l'heure actuelle, il existe plus 70 logiciels de détection de variants de structure avec des lectures courtes ont été publiés. Historiquement, les premiers logiciels se basaient sur un seul type de signal de mapping, par exemple les paires discordantes pour PEMer (Korbel et al., 2009) et BreakDancer (Chen et al., 2009), les split-reads pour Pindel (Ye et al., 2009), la profondeur de séquençage pour CNVnator (Abyzov et al., 2011). Puis, ils ont été remplacés par des méthodes plus complexes qui combinent l'information de plusieurs types de signaux de mapping, on peut citer parmi les plus connus Delly (Rausch et al., 2012) et Lumpy (Layer et al., 2014). En effet, l'information d'un seul type de signal n'est pas suffisamment spécifique dans le contexte où les régions répétées du génome génèrent un grand nombre de faux positifs. Ce qui différencient ces dernières méthodes ce sont principalement les modèles statistiques et les filtres appliqués pour tenter de discriminer les vrais positifs des faux positifs. On assiste également au développement de logiciels "méta" qui cherchent un consensus entre plusieurs méthodes existantes, par exemple metaSV (Mohiyuddin et al., 2015) et Parliament (English et al., 2015). En effet, les résultats obtenus avec différentes méthodes se recoupent peu, laissant présager de forts taux de faux positifs. Ces derniers outils permettent à l'utilisateur de ne pas se soucier du choix de l'outil ni de leur lancement, tout en améliorant la précision des prédictions, mais au détriment de la sensibilité. Enfin, la dernière génération des logiciels de détection de variants de structure se démarque par l'utilisation d'algorithmes d'assemblage local en plus des signatures de mapping (cette génération est composée du logiciel d'Illumina Manta (Chen et al., 2016), et les logiciels GRIDSS (Cameron et al., 2017) et SVaba (Wala et al., 2018)). Dans ces approches, l'assemblage local d'un sous-ensemble de lectures d'intérêt permet de valider la présence d'un variant de structure et d'affiner sa position sur le génome.

Le nombre impressionnant de logiciels développés pour détecter des variants de structure témoigne de la difficulté de la tâche. Paradoxalement, très peu de travaux ont été menés pour comparer et évaluer les performances de ces trop nombreux logiciels, et permettre ainsi de guider l'utilisateur dans le choix de l'outil. Par exemple, les premières études des variants de structure effectuées par le projet 1 000 génomes ont utilisé plusieurs logiciels de prédiction mais les auteurs des papiers ne se risquent pas à des comparaisons d'outils (1000 Genomes Project Consortium et al., 2010, 2012). En effet, la principale difficulté pour évaluer les outils provient du manque de catalogues de variants de structure de référence qui permettent d'estimer des métriques de sensibilité et de précision. On peut néanmoins citer deux études très récentes qui évaluent sur des données simulées et réelles un grand nombre

de logiciels (Kosugi et al., 2019; Cameron et al., 2019). Ces études confirment la supériorité des dernières méthodes publiées utilisant l'assemblage local. Elles montrent que les outils ont des performances très différentes en fonction des types de variants de structure, en particulier les variants de type insertions sont les plus difficiles. Enfin, même si les performances sur données réelles restent faibles globalement à cause de la petite taille des lectures, elles suggèrent qu'il reste de la place pour encore améliorer les méthodes.

### 1.3.5 De nouvelles méthodes pour les lectures longues

Avec l'arrivée des technologies de lectures longues, la donne change pour l'étude des variants de structure. En effet, de part leur grande taille, ces lectures sont beaucoup plus adaptées pour détecter ces grands variants que les lectures courtes. D'une part, ces lectures s'alignent de manière beaucoup plus spécifique et sont donc moins sensibles aux artefacts de mapping dus aux séquences répétées dans les génomes. D'autre part elles peuvent inclure l'intégralité de certains variants de structure ce qui facilite l'identification du type de variant de structure et la caractérisation des différents allèles. En termes algorithmiques, le problème se résout principalement par des algorithmes de mapping de ces lectures. Des mappers spécialisés pour les longues lectures ont été développés qui permettent de gérer le plus fort taux d'erreurs de séquençage, le plus rapide et le plus utilisé étant Minimap2 (Li, 2018). Dans le cas de la détection des variants de structure il est important qu'ils permettent également d'obtenir des alignements locaux (split-read) et/ou d'autoriser des grands gaps. Ainsi, le logiciel de découverte des variants de structure le plus connu, Sniffles (Sedlazeck et al., 2018b), est largement basé sur le développement d'un mapper dédié appelé NGMLR. En plus de Sniffles, on peut citer trois autres logiciels importants, le logiciel de Pacific Biosciences PBsv [2], nanoSV (Stancu et al., 2017) et SVIM (Heller and Vingron, 2019). Les différentes méthodes se différencient encore une fois principalement sur l'approche statistique ou algorithmique utilisée pour affecter un score de qualité aux prédictions et les filtrer. Le nombre de méthodes publiées est beaucoup plus faible (moins d'une dizaine) que pour les lectures courtes. Cela s'explique par la jeunesse de ces données mais également car les défis méthodologiques sont moindres et les premières méthodes ont donné d'emblée de bons résultats. Ainsi, de nombreuses études ont montré leurs performances nettement meilleures en termes de sensibilité et de précision par rapport aux approches de lectures courtes. Certains types de variants de structure restent néanmoins encore problématiques mais ils sont peu fréquents dans les données humaines, comme les grandes insertions d'une taille supérieure à 5 Kb et les inversions bornées par des répétitions inversées (Mahmoud et al., 2019).

### 1.3.6 Après la découverte, les autres problèmes méthodologiques associés aux variants de structure

L'essentiel des recherches méthodologiques sur les variants de structure a porté ces dernières années sur leur découverte dans des génomes re-séquencés par rapport à un génome de référence. Mais une fois, ces variants découverts, ils posent encore d'autres problèmes qui ne sont pas tous résolus. Par exemple, la représentation et le partage des données de prédiction de ces variants ne sont pas simples. Actuellement les prédictions sont reportées dans des fichiers au format VCF, mais ce format a été conçu principalement pour les SNPs et les petits indels et est peu adapté pour représenter des variants plus grands. Ainsi, il existe plusieurs façons de décrire un même évènement dans ce format malgré un document

2. `https://github.com/PacificBiosciences/pbsv`

de spécification très dense [3]. Par exemple, la position de fin d'une délétion peut être indiquée dans un champ libre, soit en indiquant la position de fin sur le génome, soit en indiquant la taille de la délétion. Une duplication peut être définie par l'un des trois types suivant : DUP, INS ou CNV. Une inversion peut être décrite par un variant de type INV ou deux variants de type BND (Break-end, équivalent à un point de cassure).

Il est important également de mentionner le problème de la normalisation de la représentation qui est rencontré lorsqu'il existe des répétitions aussi petites qu'une paire de base au niveau des points de cassure. Dans ce cas, plusieurs positions associées à différentes séquences alternatives représentent de manière équivalente le même évènement (un exemple est donné dans la Figure 1.3). Ce type de répétition s'appelle également de l'*homologie jonctionnelle* et n'est pas rare. Ne serait-ce que par hasard, l'alphabet ne comportant que quatre lettres, ce problème de normalisation est censé se poser pour près de la moitié des variants de type insertion par exemple.

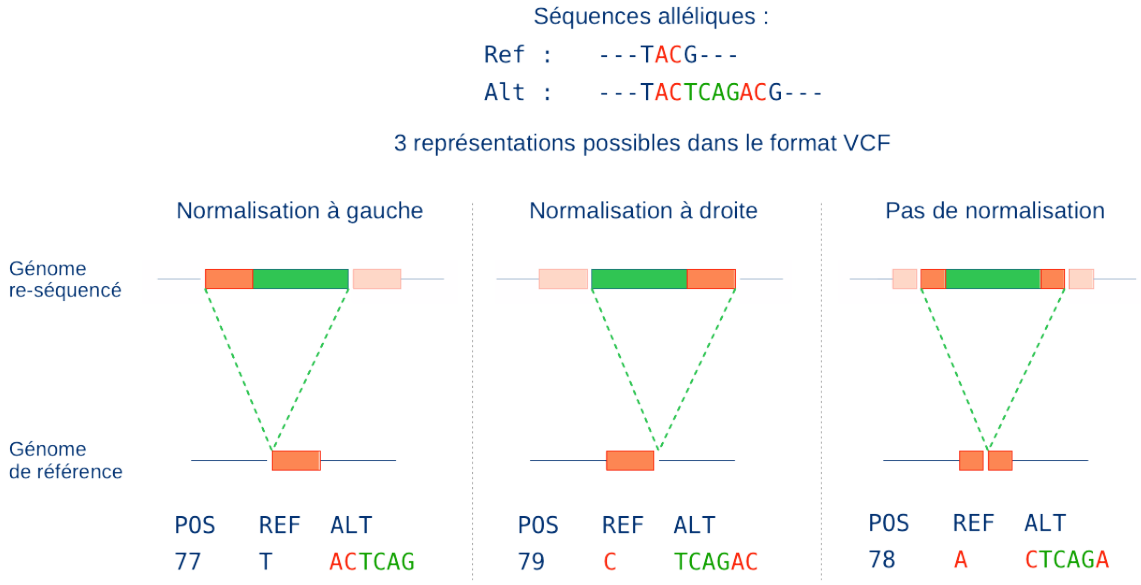


FIGURE 1.3 – Exemples des différentes représentations possibles au format VCF d'un variant de type insertion dont les deux séquences alléliques sont représentées en haut. Cette insertion possède une répétition de taille 2 (AC, en rouge) aux points de cassure. Dans ce cas, trois évènements d'insertion sont possibles et aboutissent à la même paire de séquences alléliques. Leurs représentations au format VCF diffèrent sur la position et les séquences REF et ALT. La normalisation à gauche (resp. droite) consiste à choisir la représentation avec la position la plus à gauche (resp. droite).

Même lorsqu'on s'entend sur le format de représentation, la comparaison d'ensembles de variants de structure prédits par des méthodes différentes ou sur des données différentes est une tâche encore complexe, car les variants ne sont pas toujours caractérisés au nucléotide près, et il est très probable que les prédictions d'un même évènement ne soient pas décrites à l'identique. Ainsi plusieurs outils ont été développés récemment pour effectuer de ma-

3. `https://samtools.github.io/hts-specs/VCFv4.3.pdf`

nière intelligente ces comparaisons, tels que SVanalyzer [4], SURVIVOR [5] et Jasmine (Kirsche et al., 2021). D'autres problématiques relèvent de la visualisation de ces variants ou de la quantification des allèles dans de nouveaux échantillons. Ce dernier problème s'appelle le génotypage et sera décrit plus en détails dans le Chapitre 4.

## 1.4 Plan du manuscrit

Les variants de structure jouent un rôle de plus en plus reconnu dans la diversité génétique des individus, et à ce titre, ils suscitent un intérêt croissant pour diverses applications dans les domaines allant de l'agronomie, l'écologie à la santé humaine. Cependant, leur détection et analyse avec des données de séquençage soulèvent différents problèmes méthodologiques en fonction du type de variants de structure, des caractéristiques des données de séquençage et de la question biologique posée. Dans la suite de ce manuscrit, je présente trois contributions que j'ai menées ces dernières années et qui se distinguent notamment sur ces différentes dimensions. Dans le Chapitre 2, je présente le développement d'une méthode de découverte dédiée à un type particulier de variant de structure, les insertions, avec des données de lectures courtes. Puis dans le Chapitre 3, l'arrivée des données de lectures longues nous permet de mieux caractériser ces variants en particulier chez l'homme et de revisiter les performances des outils de découvertes avec des lectures courtes. Dans le Chapitre 4, je présente une méthode dédiée cette fois aux lectures longues mais pour un problème différent : le génotypage des variants de structure dans des génomes re-séquencés. Enfin, dans le dernier Chapitre (Chapitre 5), je discuterai les perspectives de ces travaux et présenterai mon projet de recherche pour les années à venir.

4. `https://github.com/nhansen/SVanalyzer/`
5. `https://github.com/fritzsedlazeck/SURVIVOR`

# Chapitre 2

# Assemblage local de lectures courtes pour la détection d'insertions

Dans ce chapitre, nous présentons une méthode que nous avons développée pour détecter un type particulier de variant de structure, les insertions, dans un génome re-séquencé en lectures courtes par rapport à un génome de référence. Ce travail a débuté dans les années 2012-2013, en 2014 la méthode a été publiée dans la revue *Bioinformatics* (Rizk et al., 2014) (publication en Annexe A.1) et le logiciel associé, MindTheGap, a été diffusé à la communauté. Puis, dans les années qui ont suivi et encore à l'heure actuelle, la méthode a été améliorée et de nouvelles fonctionnalités ont été développées.

## 2.1 Motivations et contexte

### 2.1.1 Difficultés associées au type insertion

Parmi les différents types de variants de structure, le type *insertion* fait partie des types les plus abondants, mais également les plus difficiles à détecter. Il consiste en l'ajout d'une séquence dans le génome, qu'elle soit nouvelle (sous-type appelé *insertion nouvelle*) ou déjà existante dans le génome (sous-type appelé *duplication* ou *insertion duplicative*). C'est donc l'évènement inverse de la délétion qui supprime une séquence du génome. Lorsqu'on compare un génome à un autre, on s'attend à observer ces deux types de variant de structure en quantités similaires, puisque une délétion dans le premier génome est vue comme une insertion dans le deuxième et le choix du génome de référence est arbitraire. Cependant, il est beaucoup plus facile de détecter une délétion qu'une insertion. Le type délétion est en fait le type de variant de structure le plus facile à détecter et à caractériser. En terme de détection, une délétion génère notamment un signal de mapping qui lui est très spécifique : la diminution de la profondeur de séquençage tout le long du segment génomique supprimé. En terme de caractérisation, la séquence supprimée est obtenue directement avec le génome de référence et la séquence de l'allèle alternatif est courte et peut être comprise dans une seule lecture, même courte : c'est la jonction entre les deux séquences adjacentes au segment supprimé. Au contraire, les signaux de mapping anormaux générés par des insertions sont peu spécifiques de ce type de variant et sont variés en fonction des types d'insertion (nouvelle, duplication dispersée ou en tandem), rendant la détection du site d'insertion difficile. La séquence insérée est encore plus difficile à caractériser car elle n'est souvent pas contenue dans une lecture seule, les lectures concernées soit ne mappent pas du tout sur le génome

de référence dans le cas d'une insertion nouvelle, soit apparaissent correctement mappées à un *locus* différent dans le cas d'une duplication. La Figure 2.1 montre deux exemples d'insertions avec des signaux de mapping très différents et/ou similaires à d'autres types de variants. La caractérisation de la séquence insérée nécessite donc une étape d'assemblage *de novo* d'un ensemble de lectures courtes (à identifier).

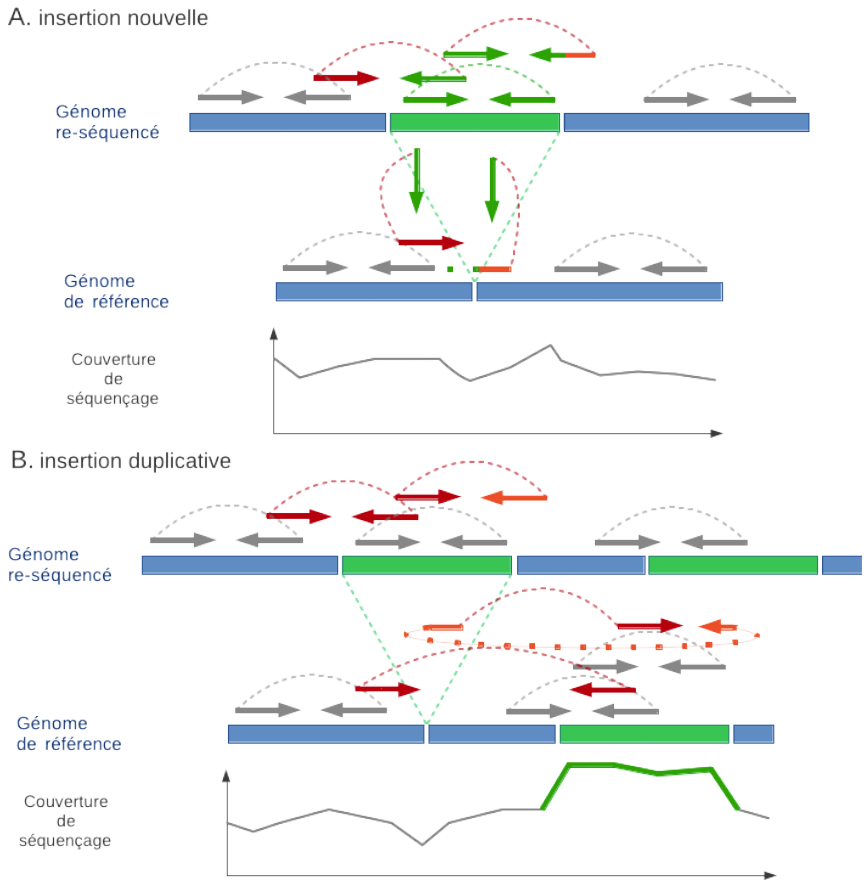


FIGURE 2.1 – Types de signaux de mapping aberrants générés par deux types d'insertions (segment inséré en vert), une insertion nouvelle (A) et une insertion duplicative (B). Contrairement aux délétions qui génèrent les 3 types de signaux de mapping aberrants (Figure 1.2), on en observe ici beaucoup moins. Pour une insertion nouvelle (A), la majorité des lectures provenant du segment inséré ne sont pas mappées et il n'y a pas ou très peu de signal dans la couverture de séquençage. Pour l'insertion duplicative (B), la couverture de séquençage est modifiée mais elle ne permet pas d'identifier le site d'insertion et les signaux de mapping discordants peuvent être interprétés comme provenant d'autres types de variants de structure (une délétion pour le signal rouge et une transposition pour le signal orange).

### 2.1.2 État de l'art

Au moment de la publication de MindTheGap, en 2014, les outils de détection génériques étaient essentiellement basés sur la détection de signaux de mapping anormaux et n'avaient pas de module d'assemblage de séquences. En effet, l'assemblage *de novo* de lectures est une tâche complexe, qui était réputée pour demander d'importantes ressources

de calcul, notamment en termes de mémoire pour représenter le graphe d'assemblage. Les outils génériques détectaient donc très peu de variant de structure de type insertion et les quelques insertions renvoyées étaient définies uniquement par leur site d'insertion, jamais avec la séquence insérée. Il existait à notre connaissance que deux outils incorporant un module d'assemblage et spécialement dédiés à ce type de variants, SOAPindel (Li et al., 2013) et NovelSeq (Hajirasouliha et al., 2010) mais qui ne permettaient de détecter chacun qu'un sous-ensemble bien particulier d'insertions : les grandes insertions nouvelles pour NovelSeq et les insertions de taille petite à moyenne (max 200 pb) pour SOAPindel. Ces limitations sont le résultat de l'étape d'assemblage qui se limite à un sous-échantillon des lectures. Pour SOAPindel, seules les lectures dont le *mate* est mappé dans la région du site d'insertion sont utilisées pour l'assemblage, ce qui limite la taille de l'insertion à moins de deux fois la taille des fragments de séquençage. Pour NovelSeq, seules les lectures non mappées sur le génome de référence sont assemblées, limitant la détection aux insertions entièrement nouvelles.

Ces choix de sous-échantillonnage étaient en partie dictés par le problème de représentation en mémoire des graphes d'assemblage construits sur l'ensemble des lectures du génome re-séquencé. En effet, en 2012, l'assembleur *de novo* ABYSS (Simpson et al., 2009) nécessitait plus de 300 Go de mémoire pour assembler un génome humain (Chikhi and Rizk, 2013). Si de telles ressources étaient envisageables pour une tâche ponctuelle d'assemblage d'un génome de référence, ce n'était pas raisonnable pour la détection de variants dans de nombreux génomes re-séquencés. Or, en 2012, Rayan Chikhi et Guillaume Rizk, deux doctorants de notre équipe, ont proposé de nouvelles méthodes et structures de données permettant de calculer et représenter en très peu de mémoire le graphe de de Bruijn d'un ensemble de lectures courtes. Le graphe de de Bruijn d'un ensemble de séquences est un graphe dirigé dont les sommets sont l'ensemble des $k$-mers, ou mots de taille $k$, contenus dans les séquences et les arcs sont les chevauchements exacts suffixe-préfixe de taille $k-1$ (Figure 2.2). Si on élimine les $k$-mers contenant des erreurs de séquençage et qu'on choisit une valeur de $k$ suffisamment grande (généralement $> 30$), le génome d'où les lectures ont été échantillonnées est représenté par un ensemble de chemins dans ce graphe. Rayan Chikhi et Guillaume Rizk ont proposé d'une part un algorithme de comptage des $k$-mers sur disque (DSK) qui permet de filtrer les $k$-mers avec des erreurs de séquençage sur la base de leur abondance dans le jeu de lectures (Rizk et al., 2013), et d'autre part une structure de données à base d'un filtre de Bloom pour représenter en mémoire l'ensemble des sommets du graphe (Chikhi and Rizk, 2013). Ces deux innovations permettent de calculer, représenter et traverser le graphe de de Bruijn de grands ensembles de lectures courtes avec très peu de mémoire et en temps raisonnable. Par exemple, l'assemblage d'un génome humain nécessite moins de 6 Go de mémoire vive avec Minia et peut donc être réalisé sur un ordinateur portable ou de bureau. Ces travaux ont été ensuite ré-implémentés et diffusés à la communauté dans la librairie C++ GATB qui facilite le développement d'outils basés sur cette implémentation du graphe de de Bruijn (Drezen et al., 2014).

Ces innovations et leurs implémentations ont démocratisé l'utilisation du graphe de de Bruijn pour bien d'autres tâches que celle à laquelle elle était cantonnée jusqu'à présent, l'assemblage *de novo* d'un génome de référence. À partir de cette implémentation, nous avons notamment développé un logiciel de correction de lectures courtes (Bloocoo) (Benoit et al., 2014), un compresseur de fichier de séquençage (Leon) (Benoit et al., 2015), un outil de découverte de petits variants sans génome de référence (discoSnp) (Uricaru et al., 2015), et, ce qui nous occupe dans ce manuscrit, une méthode d'assemblage local pour la détection des insertions (MindTheGap).

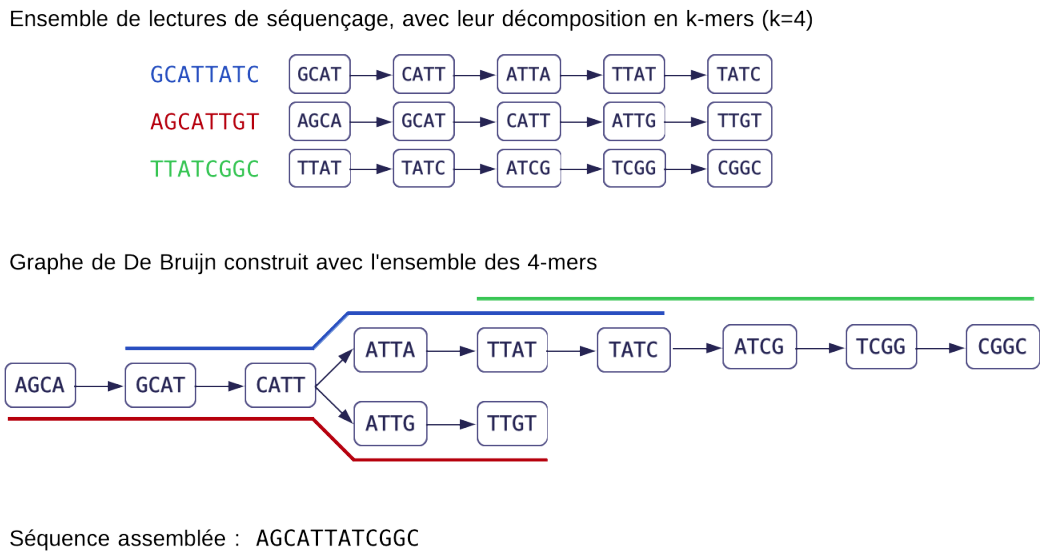


FIGURE 2.2 – Exemple d'un graphe de de Bruijn pour $k$=4, construit à partir de 3 lectures (en haut). Après avoir décomposé les lectures en $k$-mers, ces derniers sont comptés (étape non représentée ici) pour éliminer les $k$-mers rares. L'ensemble des $k$-mers distincts forment l'ensemble des sommets du graphe, les arcs correspondent aux chevauchements exacts suffixe-préfixe de taille exactement $k-1$. Utilisé pour l'assemblage *de novo* de séquences, on cherche ensuite des chemins avec des propriétés particulières, en bas est représentée la séquence assemblée issue du chemin le plus long.

## 2.2 La méthode MindTheGap

Grâce à cette représentation compacte du graphe de de Bruijn, la méthode proposée peut se permettre d'utiliser l'ensemble des lectures du génome re-séquencé pour assembler les séquences des variants d'insertion. Cela lui permet en théorie de pouvoir assembler n'importe quelle taille et n'importe quel type d'insertions, qu'elles soient nouvelles ou duplicatives. C'est une des originalités et des forces de notre méthode, mais cela ne constitue qu'une de ses deux principales étapes. Car avant d'assembler spécifiquement la séquence de l'insertion, il faut d'abord détecter l'évènement d'insertion et son site d'insertion dans le génome pour ancrer l'assemblage local.

### 2.2.1 Deux étapes originales

La méthode MindThegap est ainsi composée de deux étapes principales, représentées dans la Figure 2.3. La première, appelée Find, identifie les sites potentiels d'insertions sur le génome. La deuxième, appelée Fill, effectue un assemblage local autour de ces sites d'insertion. Plus précisément, l'étape Find renvoie des paires de $k$-mers situés de part et d'autre de chaque site d'insertion sur le génome de référence. Ces paires de $k$-mers sont données en entrée de l'étape Fill qui recherche des chemins dans le graphe de de Bruijn reliant ces deux $k$-mers.

Si l'assemblage local effectué avec l'ensemble des lectures constitue la principale motivation de développement de cette méthode, et sa principale valeur ajoutée par rapport à l'état de l'art, l'étape de détection des sites d'insertion (Find) n'en est pas moins originale.

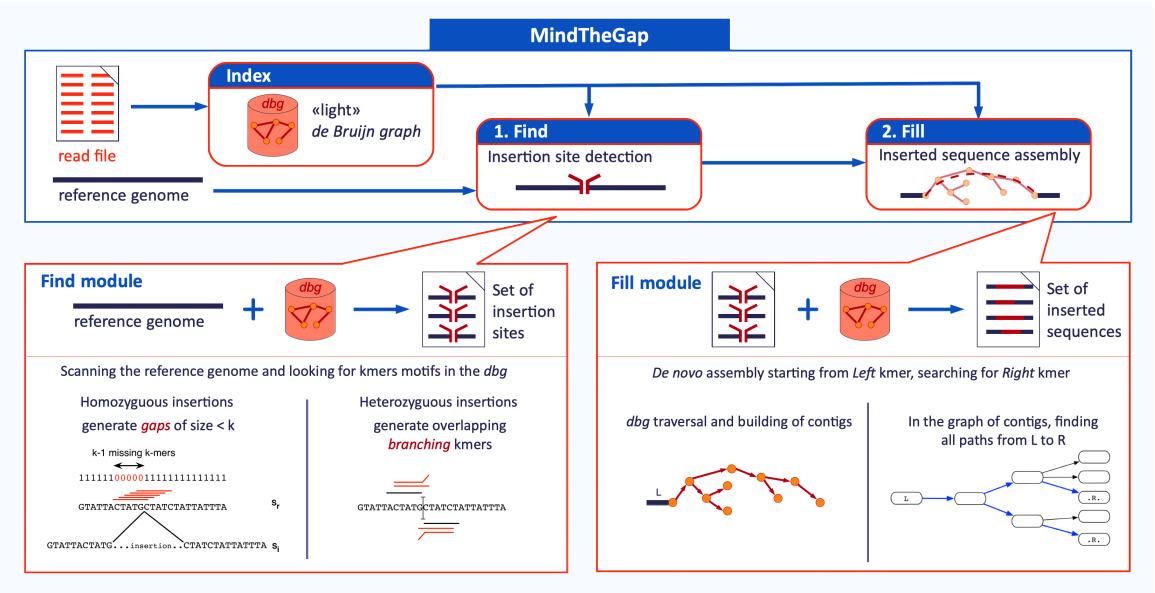


FIGURE 2.3 – Les différentes étapes de la méthode MindTheGap.

En effet, contrairement à la quasi totalité des outils de détection des variants de structure qui prennent en entrée le résultat d'une étape de mapping des lectures sur le génome de référence, MindTheGap en est indépendant. La détection des sites d'insertion ou points de cassure se fait là encore en exploitant cette représentation efficace des données par leur graphe de de Bruijn, en comparant celui-ci avec le génome de référence, sur la base de leurs $k$-mers communs et spécifiques.

Plus précisément, on recherche le long du génome de référence des motifs particuliers d'absence de $k$-mers (insertions homozygotes) ou de $k$-mers ayant plusieurs voisins (insertions hétérozygotes) qui sont générés par les insertions. Dans le cas d'une insertion homozygote, la séquence du génome de référence au site d'insertion est absente dans le génome rere-séquenséquencé. En termes de présence-absence de $k$-mers, cela se traduit dans la plupart des cas par $k-1$ $k$-mers consécutifs sur le génome de référence qui sont absents dans le graphe de de Bruijn du génome re-séquencé (Figure 2.3 à gauche dans l'encart du module Find). Nous appelons ces stretchs de $k$-mers absents des *gaps*. D'autres variants homozygotes génèrent également des gaps, mais pour la plupart leur taille est différente : elle est d'exactement $k$ pour un SNP et strictement supérieure à $k$ pour une délétion. D'autres variants, comme les inversions ou translocations, génèrent des gaps de mêmes caractéristiques que les insertions, mais ils sont beaucoup moins abondants dans les génomes que les SNPs et les délétions. Ainsi, ce motif de gap de taille $k-1$ est relativement spécifique à des variants d'insertions et est surtout très facile à énumérer en parcourant les $k$-mers du génome de référence. Il faut noter cependant que la présence d'une séquence répétée entre une extrémité de la séquence insérée et les séquences flanquant le site d'insertion peut altérer ce motif, réduisant la taille du gap de la taille de la répétition (voir un exemple dans la Figure 1 du papier en Annexe A.1). Un paramètre permet à l'utilisateur de prendre plus ou moins en compte ces répétitions potentielles, qui sont appelées parfois micro-homologies ou homologies jonctionnelles et nous verrons dans le chapitre 3 qu'il a de l'importance pour certaines données.

Dans le cas d'une insertion hétérozygote, le site de l'insertion est présent dans un des haplotypes du génome re-séquencé, on n'observe donc pas d'absences de $k$-mers, mais on

observe un motif particulier en termes de $k$-mers branchants (des $k$-mers ayant plusieurs voisins dans le graphe) : un $k$-mer branchant à droite suivi à une distance de $k$ par un $k$-mer branchant à gauche (Figure 2.3 à droite dans l'encart du module Find). De la même manière que pour les gaps, c'est la distance de $k$ entre les deux $k$-mers branchants qui différencie le type de variant entre une insertion, un SNP ou une délétion, et qui peut être réduite en cas de répétition entre la séquence insérée et le site d'insertion.

Ce choix d'éviter une étape de mapping a été guidé par plusieurs éléments : i) le gain en temps de calcul puisque le mapping est une tâche longue et que le graphe de de Bruijn est construit une seule fois pour les deux étapes, ii) contrairement aux signaux de mapping anormaux, un unique motif (voire deux en fonction de la zygotie du variant) est recherché quelque soit le type, la taille des insertions et les caractéristiques de taille de fragments des données, iii) le besoin pour l'étape suivante de récupérer des $k$-mers voisins du site d'insertion effectivement présents dans le graphe de de Bruijn et dont la position vis-à-vis de l'insertion est maîtrisée.

Lors de l'étape Fill, le graphe de de Bruijn est parcouru en partant du $k$-mer à gauche de chaque site d'insertion par un parcours en largeur. Un ensemble de contigs est construit avec l'algorithme de l'assembleur *de novo*, Minia (Chikhi and Rizk, 2013), qui écrase les petites *bulles* ou *tips*, jusqu'à atteindre des paramètres limites d'exploration locale du graphe en termes de nombre de contigs construits ou nombre total de paires de bases assemblées. Puis si le $k$-mer de droite est retrouvé dans un ou plusieurs de ces contigs, l'ensemble des chemins reliant ces deux $k$-mers sont énumérés. Une autre originalité de l'approche est de renvoyer plusieurs solutions si elles existent, plutôt que de s'arrêter à la première solution. Cependant cela entraine une difficulté supplémentaire, car des petits polymorphismes ponctuels peuvent générer un grand nombre de solutions peu différentes. Afin de limiter la redondance dans les résultats due à du polymorphisme ponctuel, les différentes solutions sont alignées et les plus similaires sont représentées par une unique séquence pour renvoyer un ensemble de séquences présentant deux à deux au moins 10 % de divergence.

Pour les détails algorithmiques et d'implémentation nous renvoyons le lecteur vers la publication de la méthode donnée en Annexe A.1.

### 2.2.2 Nouvelles fonctionnalités postérieures à la publication

Depuis sa publication en 2014, le logiciel a évolué pour améliorer son utilisation, ses performances ou encore pour élargir son domaine d'application.

En termes d'interface avec l'utilisateur et de performances, on peut citer entre autres choses, la parallélisation du module Fill, la paramétrisation automatique en fonction des données en entrée du graphe de de Bruijn, l'adaptation à des données de type exome et la sortie au format VCF avec la normalisation à gauche des variants (voir un exemple dans la Figure 1.3).

D'un point de vue algorithmique, nous présentons ci-dessous les deux développements principaux, qui ont permis notamment d'ajouter deux fonctionnalités nouvelles à l'outil : la détection d'autres types de variants dans le module Find et l'adaptation du module Fill pour le problème de gap-filling ou finishing d'un assemblage *de novo* de génome.

#### Détection d'autres types de variants et gestion des variants proches

Les motifs de $k$-mers recherchés dans la version initiale du module Find sont basés sur l'hypothèse que les deux $k$-mers de part et d'autre du site d'insertion sont identiques

dans le génome re-séquencé et dans le génome de référence. Cela empêche la détection de sites d'insertions localisés à proximité (moins de $k$ nucléotides) d'un autre variant, et en particulier de variants souvent simples à détecter par ailleurs et très fréquents : les SNPs. Nous avons donc étendu les définitions des motifs recherchés et implémenté des algorithmes permettant de détecter des SNPs et délétions homozygotes, à proximité ou isolés des sites d'insertions. Nous avons vu dans la section précédente que les variants homozygotes génèrent des stretchs de $k$-mers le long du génome de référence qui sont absents dans le génome re-séquencé (des *gaps*), dont la taille dépend du type de variant. Un site d'insertion isolé génère un gap de taille inférieure à $k-1$. Les SNPs isolés génèrent des gaps de taille exactement $k$ et les délétions des gaps de taille supérieure à $k$, dépendant principalement de la taille du segment supprimé. Lorsque ces variants sont proches (à moins de $k$ nucléotides), les motifs se chevauchent sur le génome et génèrent un plus grand gap. Après avoir identifié ces trop grands gaps, l'idée de la méthode est d'essayer les différentes façon de corriger le gap de chaque côté par la recherche de $k$-mers proches dans le graphe de de Bruijn (ou micro-assemblage). Cette méthode de correction des gaps dans le génome est notamment inspirée d'une méthode de correction *de novo* des erreurs de séquençage dans les lectures courtes, que nous avions proposée auparavant, appelée Bloocoo (Benoit et al., 2014).

Ces développements permettent de détecter d'autres types de variants (délétions et SNPs) en plus des insertions, avec comme objectif de résoudre le maximum de *gaps* ou différences de $k$-mers entre le génome re-séquencé et le génome de référence. Mais surtout, cela a permis d'améliorer le rappel de la détection des variants de type insertion en présence d'autres polymorphismes dans les données de séquençage.

**MindTheGap pour finir un assemblage de génome**

La deuxième fonctionnalité ajoutée à MindTheGap depuis sa publication fait intervenir le module d'assemblage local de MindTheGap. Le module Fill de MindTheGap effectue un assemblage local entre 2 séquences, d'une *source* vers une *cible*. Ce type d'assemblage ciblé peut servir à d'autres applications que la détection d'insertions, et en particulier pour *boucher* les trous d'un assemblage *de novo* de génome, étape appelée en anglais le *gap-filling*. Nous avons développé deux pipelines qui impliquent le module Fill de MindTheGap pour ce type d'application : MTG-link, pour le gap-filling de génome en utilisant des données de séquençage de type *linked-reads* et MinYS pour l'assemblage de génomes guidé par une référence dans des données métagénomiques. Dans le premier cas, MindTheGap est utilisé tel quel, tandis que pour MinYS des développements ont du être apportés à MindTheGap pour répondre à cette nouvelle application.

MinYS, pour *Mine Your Symbiont*, est un logiciel qui permet d'assembler un génome à partir d'un séquençage métagénomique en utilisant un génome de référence comme guide. Le cas d'application typique est celui de l'assemblage d'un génome bactérien ou viral présent dans le microbiote d'un hôte eucaryote. Dans le cas courant où l'organisme hôte est séquencé en entier, les génomes issus de son microbiote sont représentés par une faible proportion des lectures de séquençage (1 à 5 %). Dans de nombreux cas, un symbiote en particulier est visé par le biologiste et un génome plus ou moins proche phylogénétiquement est déjà connu. Pour éviter d'assembler l'ensemble des génomes du microbote et celui de l'hôte, le génome de référence peut être utilisé pour focaliser l'assemblage sur le génome d'intérêt. L'approche classique est de ne considérer que les lectures qui s'alignent sur le génome de référence. Le défaut majeur de cette approche est de manquer des parties du génome d'intérêt qui seraient absentes ou trop divergentes du génome de référence utilisé. Pour palier à ce défaut,

le module Fill de MindTheGap intervient en deuxième étape de la méthode MinYS, pour reconstruire les séquences manquantes ou trop divergentes du génome de référence.

La première étape construit un ensemble de contigs à partir des lectures alignées sur le génome de référence. Dans la deuxième étape, les extrémités de ces contigs sont donnés en entrée de MindTheGap, qui effectue des assemblages locaux avec le graphe de de Bruijn construit à partir de la totalité des lectures de séquençage métagénomiques, et non plus les seules lectures alignées sur le génome de référence. Afin d'être le moins possible influencé par la structure du génome de référence, on effectue une recherche exhaustive entre toutes les paires possibles d'extrémités de contigs, sans tenir compte de l'ordre ni de l'orientation de ces contigs dans le génome de référence. C'est sur ce point, que de nouveaux développements sur le module Fill ont été apportés. L'utilisation naïve de ce module impliquerait d'effectuer $2 \times N \times 2 \times (N-1)$ explorations locales du graphe de de Bruijn, si $N$ est le nombre de contigs initial. En effet, pour chaque extrémité de contig (source), il y a $2 \times (N-1)$ cibles possibles. L'algorithme a été modifié pour permettre de rechercher à partir d'une séquence source tous les chemins possibles vers un ensemble de séquences cibles (et non plus une seule), en une seule exploration locale du graphe, tout en évitant de trouver des solutions redondantes qui ré-assembleraient, par exemple, des contigs initiaux.

Dans cette approche, nous avons ainsi tiré partie de deux spécificités de MindTheGap : la capacité à passer à l'échelle de gros jeux de données (avec la structure de données compacte du graphe de de Bruijn) et le fait que plusieurs solutions sont renvoyées si elles existent. Ce dernier point permet notamment d'identifier des variants de structure ou des souches avec des structures de génome différentes qui co-existent dans le microbiote séquencé. Sur ces deux points MinYS se démarque nettement de l'état de l'art. Ce travail a fait l'objet d'une partie de la thèse de Cervin Guyomar que j'ai co-encadré et d'un papier publié récemment dans la revue *NAR Genomics and Bioinformatics* (Guyomar et al., 2020). Le logiciel, implémenté en Python, est diffusé sur github [1] et comme un paquet Bioconda [2].

## 2.3 Validation et applications de MindTheGap

### 2.3.1 Validation de l'approche de détection d'insertions

Dans la publication de 2014, la méthode a été validée dans un premier temps sur des données simulées avec des insertions de différentes tailles et dans des génomes de différentes complexité : du plus simple avec la bactérie *E. coli*, en passant par le génome de la levure *S. cerevisiae*, au plus complexe avec un chromosome humain. Des insertions nouvelles ont été simulées en simulant des lectures de séquençage sur le génome de référence, mais en utilisant comme génome de référence pour la détection un génome muté avec des délétions. Les séquences insérées à découvrir sont les séquences délétées dans le génome de référence, ainsi, elles ne sont pas aléatoires et reflètent la complexité des génomes utilisés.

Sur ces données simulées, MindTheGap a obtenu de très bons résultats en termes de sensibilité et de précision de détection des insertions. Sur un petit génome, avec peu de répétitions, comme celui d'*E. coli*, MindTheGap détecte plus de 97 % des insertions simulées quelques soit leur taille (de 10 pb à 1Kb) avec une précision presque parfaite (moins de 1 % de faux positifs). Comme attendu, pour des génomes plus grands et contenant plus de répétitions, la sensibilité décroit avec la taille des séquences à assembler (jusqu'à 65%

1. `https://github.com/cguyomar/MinYS`
2. `https://anaconda.org/bioconda/minys`

pour des insertions de 1 Kb dans un chromosome humain). Les sites d'insertion sont bien détectés quelque soit la taille de l'insertion, mais c'est bien l'étape d'assemblage local qui est impactée par la présence de régions dans le graphe de de Bruijn plus complexes, très denses en branchements, fréquemment générées par des séquences répétées en un grand nombre de copies comme les éléments transposables. Dans ce cas, les limites d'exploration locale du graphe sont atteintes avant d'avoir atteint la séquence cible. Les performances sont également un peu moins bonnes pour les insertions qui sont à l'état hétérozygote dans les données de séquençage par rapport aux insertions homozygotes (perte de 10 à 30 % de sensibilité). Cela était attendu car, d'une part, le motif de $k$-mers généré par les insertions hétérozygotes est moins spécifique que celui des insertions homozygotes (il peut notamment être confondu avec des régions répétées), et d'autre part, la profondeur de séquençage est deux fois plus faible pour assembler la séquence insérée.

Ces résultats obtenus sur des données simulées assez simples ont constitué une preuve de concept que l'approche fonctionnait. Les tests sur données simulées ont permis d'analyser finement les raisons des erreurs et/ou manques de détection (faux positifs et faux négatifs), d'analyser l'influence des différents paramètres sur les résultats et d'optimiser la méthode et ses paramètres. Les données simulées ont également permis de comparer notre approche à d'autres logiciels. Cependant, au moment de la publication, seuls deux logiciels permettaient d'assembler des variants d'insertion, SOAPindel (Li et al., 2013) et NovelSeq (Hajirasouliha et al., 2010). Nous n'avons pas réussi à faire fonctionner Novelseq, qui n'est plus maintenu depuis 2012 et SOAPindel, comme attendu, s'est révélé limité aux insertions de petites tailles. Ainsi, cette comparaison a essentiellement mis en évidence que MindTheGap était le seul outil en 2014 à pouvoir détecter et assembler des variants d'insertions plus grands que la taille des fragments séquencés.

Si les données simulées présentent de nombreux avantages pour l'évaluation et la comparaison des méthodes, elles ont cependant une limite majeure de ne pas toujours bien représenter la réalité biologique et de sous-estimer la difficulté du problème. Dans ce cas particulier, nos données simulées présentent deux limitations importantes : l'absence d'autres polymorphismes dans les données de séquençage (SNPs et petits indels qui sont beaucoup plus fréquents en pratique que les grandes insertions), et le fait que les insertions ont été simulées à des localisations génomiques aléatoires (probabilité uniforme le long du génome), ce qui ne reflète pas la réelle complexité des insertions et de leurs points de cassure. Or, en 2014, il existait très peu de données avec des longs variants d'insertions dont la séquence insérée était résolue et validée (voir Chapitre 3, page 32). Ainsi, la validation de MindTheGap sur des données réelles n'a été effectuée que sur une vingtaine d'insertions identifiées au préalable chez un individu humain par du séquençage de fosmids. Cette expérimentation a notamment permis de montrer que MindTheGap était capable d'assembler avec une grande précision certaines de ces insertions, notamment de très longues ($>$ 4 Kb), avec des ressources de calcul très raisonnables pour un jeu de données contenant plusieurs milliards de lectures. Elle a également permis de mettre en évidence plusieurs pistes d'amélioration, comme la gestion d'autres polymorphismes (SNPs et délétions) à proximité des points de cassure qui a été résolue par la suite (voir Section 2.2.2).

### 2.3.2 Applications sur des données réelles

Suite à sa publication, MindTheGap a été utilisé pour produire des résultats biologiques, aussi bien pour découvrir que pour valider des variants d'insertions, mais aussi comme gap-filler pour produire des assemblages de génomes. Les types de données et d'organismes

étudiés sont très divers, allant de génomes viraux, de mitochondries, de bactéries à des génomes plus complexes d'insectes, de plantes et des exomes humains. Ces résultats ont été obtenus notamment lors de collaborations avec des biologistes, mais également sans notre collaboration suite à la publication du logiciel comme l'attestent au moins cinq publications (Burioli et al., 2017; Carrier et al., 2018; Daval et al., 2019; Feldman et al., 2019; Fuentes et al., 2019). Parmi ces publications, on peut notamment citer celle de Fuentes et al. (2019) parue dans la revue à fort impact *Genome Research* en 2019, qui étudie le polymorphisme de structure de 3 000 génomes du riz, dans laquelle MindTheGap a été l'outil choisi pour détecter les grandes insertions.

Dans le cadre d'une collaboration à long terme avec l'INRAE sur le génome du puceron du pois, nous avons utilisé MindTheGap à plusieurs reprises : (i) pour rechercher des variants d'insertion qui pourraient expliquer la variabilité du nombre de lectures non mappées sur le génome de référence entre différentes populations et races d'hôtes de puceron (Gouin et al., 2015), (ii) pour obtenir le génome de référence d'un symbiote encore peu caractérisé du puceron du pois, *Rickettsia sp* (Guyomar et al., 2018) et (iii) pour répertorier les variants structuraux d'un phage responsable de traits phénotypiques importants (phage APSE de la bactérie *Hamiltonella defensa*, publication en cours de préparation). Dans le cadre de collaborations encore en cours sur des génomes de papillons, nous utilisons MindTheGap comme gap-filler pour l'assemblage de génomes mitochondriaux ou de régions d'intérêt. C'est le cas par exemple, pour le *locus Supergene* chez le papillon mimétique *Heliconius numata*, qui présente un polymorphisme d'inversion associé au patron de coloration des ailes des papillons. Nous utilisons le pipeline MTG-link pour reconstruire cette région de 1,3 Mb dans une douzaine d'individus re-séquencés avec des lectures liées et pouvoir ainsi étudier plus précisément leurs différences structurales.

Enfin, plus récemment, dans le cadre de la thèse de Wesley Delage, que j'ai co-encadré avec Julien Thévenon, médecin-généticien spécialiste des maladies rares, MindTheGap a été utilisé pour des applications de santé humaine, en collaboration avec l'Inserm de Grenoble et le Centre Hospitalier Universitaire de Dijon. MindTheGap a notamment été appliqué sur des données de séquençage d'exomes de patients atteints de maladies rares, pour lesquelles certaines variations génomiques impliquées sont connues. Les essais réalisés sur une trentaine de séquençages d'exomes n'ont pas permis de retrouver les variations attendues, mais ont produit des ensembles de prédictions qui nécessitent d'être analysés plus en détails. En effet, les quelques variations connues à l'heure actuelle étaient des CNV dont les localisations ne sont pas précisemment connues, ce qui rend difficile une comparaison avec MindTheGap, et les grandes insertions ($> 50$ pb) qui sont la cible principale de MindTheGap sont peu fréquentes dans les exons et encore peu caractérisées dans le diagnostic clinique car aucune méthode d'analyse standardisée n'existe. Cependant, une réussite de cette expérience a été l'utilisation de MindTheGap sur des données réelles par des utilisateurs du domaine médical et d'avoir des retours sur l'outil.

### 2.3.3 Passage à l'échelle sur un génome humain entier

Après avoir appliqué MindTheGap sur des données d'exomes humains, l'étape suivante était de l'appliquer sur des données de re-séquençages de génomes entiers humains. La confrontation à des données réelles de séquençage de génomes humains a cependant apporté une mauvaise surprise en terme de passage à l'échelle.

Appliqué à des données simulées, l'outil montre des bonnes performances en temps de calcul et mémoire vive utilisée. Par exemple, sur des données simulées sur le chromosome

3 humain, MindTheGap met environ 15 minutes à détecter et assembler les 200 insertions simulées. Sur des vraies données de séquençage du génome de *C. elegans*, on obtient un temps de calcul très similaire. Mais les temps de calcul observés sur un génome humain étaient d'un tout autre ordre de grandeur et montraient une augmentation non linéaire avec la taille des données en entrée. En effet, au lieu d'une multiplication par 10 attendue du fait de la taille du génome, soit quelques heures, on obtenait des temps déraisonnables de plusieurs centaines d'heures, dus essentiellement au module Fill. Sans faire une analyse très détaillée de la complexité des algorithmes, on peut facilement estimer que la complexité du module Find de détection des sites d'insertion est linéaire avec la taille du génome, et celle du module Fill est linéaire avec le nombre de sites d'insertions détectés à l'étape du Find. On aurait pu penser que ce nombre de sites d'insertion augmenterait de manière linéaire avec la taille du génome. Cela est sûrement vrai pour une séquence aléatoire ou un génome simple. Mais dans le cas du génome humain, qui est un génome complexe, représenté à plus de la moitié par des séquences répétées, on observe que ce nombre de sites n'augmente pas linéairement : la proportion de sites faux positifs est plus importante dans un génome complet qu'avec un seul chromosome. De plus, le temps de calcul moyen pour assembler chaque site est plus de 20 fois plus grand avec le graphe construit sur le génome complet qu'avec un seul chromosome (par exemple, près de 20 secondes versus moins d'1 seconde par site), alors que les paramètres de limite d'exploration du graphe ne dépendent pas de la taille de celui-ci.

Ainsi, la taille et la complexité du génome humain font augmenter le nombre de sites à assembler et le temps moyen pour les assembler. Mais, il existe encore un autre facteur qui aggrave encore le temps de calcul, et qui est souvent négligé dans les données simulées, c'est le vrai polymorphisme contenu dans les données réelles de séquençage. La très grande majorité de ce polymorphisme est sous la forme de SNPs et de petits indels (1 à 2 pb). En particulier, ces petits indels (les insertions, mais aussi une partie des petites délétions) génèrent le même motif de $k$-mers que les grandes insertions visées par MindTheGap. Ce qui semblait être un avantage de MindTheGap de ne pas dépendre de la taille des insertions, devient alors un énorme fardeau. Ainsi en appliquant le module Find à un vrai séquençage plein génome du génome humain, on peut obtenir plus de 250 000 sites potentiels d'insertions à assembler, soit une estimation du temps d'assemblage à plus d'un mois (sans parallélisation) !

Nous avons proposé plusieurs solutions, qui sans résoudre complètement ce problème sur données réelles humaines, en diminuent l'impact (division par 2,5 voire 3 du nombre de sites à assembler) :

- l'imputation des SNPs homozygotes de l'individu re-séquencé dans le génome de référence avant de lancer le module Find. Cette étape rajoute souvent peu de temps de calcul, puisque dans de nombreuses analyses, les SNPs sont les premiers polymorphismes à être détectés et étudiés et ce type de données est donc très souvent disponible.
- l'assemblage dès le module Find des petites insertions. L'approche est similaire à celle utilisée pour les variants proches, des tentatives d'assemblage local de 1 à 2 pb sont effectuées pour chaque site potentiel d'insertion. Cela évite d'explorer dans le module Fill une partie du graphe beaucoup plus importante que nécessaire pour ces petites insertions qui sont généralement triviales à assembler.
- un filtre sur les caractéristiques de branchement des $k$-mers voisins au site d'insertion est appliqué pour limiter le nombre de sites faux positifs. Les sites faux positifs sont le plus souvent détectés avec le motif hétérozygote, dans des régions répétées qui se caractérisent par une grande densité de $k$-mers branchants.

## 2.4 Conclusion

Nous avons présenté dans ce chapitre une méthode originale pour détecter et assembler des variants d'insertions avec des données de lectures courtes. Cette méthode était l'une des premières quand elle a été publiée à faire intervenir une structure de données et des algorithmes d'assemblage de séquence pour la détection de variants de structure. Cette stratégie a montré son efficacité pour les variants de type insertion, mais elle est maintenant de plus en plus courante dans la nouvelle génération d'outils de détection génériques pour tous les types de variants de structure (Chen et al., 2016; Cameron et al., 2017; Wala et al., 2018).

Le développement du logiciel MindTheGap a débuté dans les années 2012-2013, et a impliqué, à des degrés d'implications variables, plusieurs personnes que j'ai encadrées ou avec qui j'ai collaboré (Guillaume Rizk, Anaïs Gouin, Pierre Marijon, Cervin Guyomar, et Wesley Delage). Il se poursuit encore à l'heure actuelle pour améliorer les performances de la méthode ou ajouter de nouvelles fonctionnalités. Le logiciel, implémenté en C++, est diffusé librement sur github [3] et comme un paquet bioconda [4]. Il dispose d'une communauté d'utilisateurs comme en témoignent le nombre de téléchargements [5] et les retours d'utilisateurs que nous recevons. Ainsi, il a été appliqué avec succès pour produire des résultats biologiques, principalement sur des petits génomes ou des exomes. Son intégration dans des pipelines d'assemblage ou d'amélioration d'assemblage de génomes est également un succès récent (outils MTG-link et MinYS (Guyomar et al., 2020)).

Un domaine d'application que nous avons commencé à étudier ces trois dernières années mais qui mériterait d'être plus développé est celui de la santé humaine. Dans ce contexte, nous avons rencontré deux difficultés principales : le passage à l'échelle de la méthode sur des données plein génome du génome humain et l'évaluation des résultats obtenus dans un contexte où peu de grandes insertions avec un intérêt clinique ont été caractérisées.

Enfin, nous verrons dans le chapitre suivant que l'arrivée des données de séquençage de troisième génération nous permet d'évaluer différemment ces outils et remet en question certains résultats et choix méthodologiques.

3. `https://github.com/GATB/MindTheGap`
4. `https://anaconda.org/bioconda/mindthegap`
5. Chiffres donnés par Bioconda : près de 4 000 sur les 2 dernières années.

# Chapitre 3

# Vers une meilleure compréhension des faibles performances des outils avec des lectures courtes

Dans ce chapitre, nous présentons des travaux de caractérisation de données d'insertions humaines qui ont été obtenus dans le cadre de la thèse de Wesley Delage (2017-2020), et qui ont été publiés en 2020 dans la revue *BMC Genomics* (Delage et al., 2020) (publication en Annexe A.2).

## 3.1 Motivations et contexte

### 3.1.1 Apport des lectures longues pour la détection des variants de structure

Nous avons vu que les lectures courtes sont peu adaptées pour la détection des variants de structure, et en particulier pour les variants de type insertions. Cela se concrétise/est visible dans la très faible représentation de ce type de variant dans les bases de données, les catalogues de variants et les études des variants de structure dans les populations. Par exemple, les premières publications des résultats du projet 1 000 génomes sont presque exclusivement consacrés aux délétions (1000 Genomes Project Consortium et al., 2010, 2012). Les outils de détection de variants de structure sont presque exclusivement évalués sur des variants de type délétion. Enfin, l'exemple le plus frappant est celui de la base de données dbVar, qui est la base de données de référence répertoriant les variants de structure identifiés chez l'homme (Lappalainen et al., 2012). En 2020, ce sont plus de 6 millions de variants de structure qui sont stockés dans cette base de données provenant de 196 études. On observe que les variants de type insertion sont largement sous-représentés par rapport aux délétions : seulement 28% des variants contenus dans dbVar correspondent à des insertions, le reste étant très majoritairement des délétions (70 %). Cela fait un ratio de 2,5 fois alors qu'on s'attendrait à avoir un ratio équilibré entre ces deux types de variants, qui sont le symétrique l'un de l'autre. Mais surtout, au sein de ces 28 % d'insertions, seulement 1,5% sont précisément caractérisées et possèdent une séquence nucléique associée.

L'arrivée des technologies de séquençage en lectures longues, telles que ONT et PacBio, promettait de changer les choses pour la détection et l'analyse des variants de structure. En effet, la grande taille des lectures facilite la détection des grands variants puisque de

nombreux variants de structure peuvent être entièrement inclus dans une même lecture et que le mapping des lectures ou morceaux de lectures (split-mapping) sur le génome de référence est beaucoup plus spécifique. Ainsi, en 2017, les premiers logiciels de découverte des variants de structure avec des lectures longues ont été développés (voir Section 1.3.5 p. 17), mettant en évidence de nombreux nouveaux variants de structure par rapport aux ensembles détectés seulement avec des lectures courtes (Sedlazeck et al., 2018b; Mahmoud et al., 2019). L'assemblage *de novo* de génomes est également facilité avec ces technologies et la stratégie de découverte de variants de structure par comparaison d'assemblages est devenu possible, même si elle reste plus coûteuse en ressources de calcul que l'approche par mapping.

### 3.1.2 2019-2020 : les premiers catalogues exhaustifs et précis chez l'homme

Avec ces nouveaux moyens de détection, il est alors devenu envisageable d'obtenir enfin des ensembles de variants de structure représentatifs, fiables et bien caractérisés pour une espèce donnée. Plusieurs consortiums internationaux se sont lancés dans ce type de projet notamment pour le génome humain. C'est le cas du "Human Genome Structural Variation Consortium" (HGSV) [1], faisant suite au groupe de travail sur les variants de structure du projet "1 000 génomes" et du consortium "Genome in a Bottle" (GiaB) [2]. Ces deux consortiums ont rendu publique des ensembles de variants de structure sur le génome humain sans précédent, à peu près à la même période, en 2019 et 2020 respectivement (Chaisson et al., 2019; Zook et al., 2020)). Ces deux études se sont chacune appuyées sur de grandes quantités de données (équivalent à plus de 500 X de couverture du génome humain chacune), produites par jusqu'à 10 technologies de séquençage différentes, et également sur des moyens de calculs et humains conséquents pour obtenir ces résultats (publications avec 90 et 50 auteurs respectivement, dont certains communs). Cependant, les objectifs et les méthodologies diffèrent. Pour la première, l'objectif principal était d'obtenir un catalogue le plus exhaustif possible des variants de structure afin de mieux connaitre et caractériser le paysage de ces variants chez l'homme. Le second objectif était de comparer et d'évaluer les performances des différentes technologies de séquençage pour la découverte des variants de structure. Pour Genome in a Bottle, qui est un consortium dédié à la production de données et d'outils d'évaluations pour la recherche clinique, l'objectif était de fournir un ensemble de variants le plus fiable possible pour permettre l'évaluation des stratégies et outils de détection des variants de structure. D'un point de vue méthodologique, l'étude du HGSV repose sur une stratégie d'assemblage des haplotypes alors que l'étude du GiaB sur de l'alignement contre le génome de référence. Un travail important et très chronophage dans les deux études consiste ensuite à combiner en un seul ensemble non redondant et fiable (appelé en anglais *callset* ou ici catalogue) les variants de structure prédits par les multiples outils et types de données utilisés.

Ces catalogues de variants de structure sont une ressource essentielle pour les développeurs d'outil de détection de variants de structure, car ils permettent d'évaluer les outils avec des données réelles et surtout sur des variants réels, dont les caractéristiques peuvent être mal représentées dans les simulations de variants. En particulier pour les variants de type insertions, ces catalogues sont inédits : ce sont les premiers ensembles d'insertions de plus de 50 pb à l'échelle du génome entier dont le site d'insertion et la séquence insérée sont parfaitement caractérisés. Ainsi, dès leur parution, nous avons testé notre outil MindTheGap

1. `https://www.internationalgenome.org/human-genome-structural-variation-consortium/`
2. `https://www.nist.gov/programs-projects/genome-bottle`

sur ces jeux de données. Les résultats ont produit un choc et une grande déception : MindTheGap n'était capable de détecter qu'une infime fraction de ces variants réels, moins de 5 %. Notre outil n'était pas le seul en cause, d'autres outils concurrents testés n'atteignaient pas plus de 10 % et les publications du HGSV et de GiaB avaient montré le faible rappel des détecteurs de variants de structure basés sur les lectures courtes et en particulier pour les variants de type insertion. Cependant, l'écart de rappel avec nos résultats sur données simulées (entre 75 et 95 % contre moins de 5%) était frappant et témoignait d'une faible connaissance des caractéristiques des vraies insertions.

C'est pourquoi, dans ce travail, nous nous sommes attachés à caractériser le plus finement possible l'ensemble des variants d'insertions catalogués par ces deux études. Puis, nous avons cherché à identifier parmi les caractéristiques répertoriées, lesquelles étaient liées à une perte de rappel des outils de détection d'insertions basés sur des lectures courtes. Ces analyses ont été publiées dans le journal BMC genomics (Delage et al., 2020), nous présentons les principaux résultats obtenus dans les deux sections suivantes et renvoyons la lectrice et le lecteur à la publication produite en Annexe A.2 pour les détails des résultats, les figures et les méthodes.

## 3.2 Caractéristiques des vraies insertions chez l'homme

### 3.2.1 Quatre niveaux de caractérisation des insertions

Le degré de difficulté de détection d'un variant d'insertion peut dépendre de plusieurs facteurs comme par exemple le contexte génomique du variant ou bien la nature de la séquence qu'il faut assembler. Nous avons défini quatre niveaux de caractérisation des variants d'insertion :

1. la nature de la séquence insérée, qui définit le type de l'insertion,
2. la taille de la séquence insérée,
3. le contexte génomique du site d'insertion, en termes de séquences codantes, et de natures des séquences répétées,
4. la complexité des points de cassure, mesurée par la taille des homologies jonctionnelles (voir ci-dessous).

Si les catégorisations par taille et par contexte génomique sont assez classiques et assez immédiates, la distinction des différents types d'insertion (premier niveau) et l'analyse fine des points de cassure constituent une originalité de notre travail, et ne sont possibles qu'avec des insertions dont la séquence est résolue.

**Annotation du type d'insertion** Nous avons vu qu'il existait deux grands types : les insertions nouvelles dont la séquence est absente du génome de référence et les insertions duplicatives (ou duplications) dont la séquence existe dans le génome de référence. Parmi les duplications, on peut distinguer des sous-types très différents, en fonction de la localisation et du nombre de copies de la séquence dupliquée dans le génome. Par exemple, on distingue l'insertion d'un élément transposable, qui possède plusieurs centaines voire milliers de copies très similaires dans le génome, d'une grande séquence dupliquée en tandem, ou encore de l'expansion d'un motif court répété en tandem un grand nombre de fois dans la séquence insérée. On peut s'attendre à ce que ces différents types génèrent des signaux différents de mapping des lectures et des difficultés différentes pour l'assemblage. Ainsi, pour analyser

les causes de perte de rappel des outils de détection, il est important de savoir annoter et distinguer ces différents types. Ce travail d'annotation n'avait été fait que partiellement et de manière hétérogène entre les deux catalogues de variants de structure du HGSV et du GiaB, ne permettant pas de comparer les deux études, ni de représenter tous les types existants.

Nous avons alors proposé une annotation en cinq types, qui permet de couvrir l'ensemble des différents types et sous-types de la base de données dbVar [3] :

— insertion *de novo* : absence de la séquence inserée dans le génome de référence ;
— insertion d'un élément mobile : séquence similaire à des éléments mobiles connus ;
— expansion d'une 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 site d'insertion ;
— duplication dispersée : séquence dont une copie est présente dans le génome de référence à un *locus* différent du site d'insertion.

Nous avons alors proposé une méthodologie d'annotation qui permet d'assigner un type de manière non ambigüe à une grande majorité des variants d'insertion. Cette méthodologie est basée principalement sur l'alignement de la séquence insérée contre le génome de référence, les régions adjacentes au site d'insertion et une banque d'éléments transposables. Cependant, dans de très nombreux cas, les alignements obtenus ne couvrent pas entièrement la séquence insérée ou bien peuvent relever de différents types. Nous avons donc fait intervenir un seuil de couverture de la séquence insérée qui est paramétrable et un arbre de décision pour annoter les insertions qui pourraient relever de plusieurs types (voir la Figure 1 de l'article en Annexe A.2).

**Homologie jonctionnelle au point de cassure** La présence de petites séquences répétées aux points de cassure des variants de structure peut impacter l'alignement des lectures et ainsi gêner la détection précise et correcte des variants. Or, il est connu que certains mécanismes moléculaires de réparation de l'ADN qui sont responsables de la formation de réarrangements peuvent générer de telles micro-duplications, souvent appelées micro-homologies ou homologies jonctionnelles, au niveau des points de cassure (Ottaviani et al., 2014). Du fait du nombre limité de catalogues de variants de structure dont la séquence est résolue, ces homologies ont été très peu caractérisées à grande échelle en taille et en fréquence (la plus grosse étude porte sur environ 2000 variants humains, contenant moins de 400 insertions (Kidd et al., 2010)). Nous avons donc cherché à quantifier ce phénomène dans ces nouveaux catalogues d'insertions humaines. Sans *a priori* sur leur taille, nous avons appelé une homologie jonctionnelle toute répétition de plus d'une paire de base entre la séquence adjacente à gauche (resp. à droite) du site d'insertion et la fin (resp. le début) de la séquence insérée (voir aussi l'exemple d'homologie jonctionnelle de la Figure 1.3). Comme la précision de la localisation et/ou de la séquence peut ne pas être parfaite dans les catalogues, nous avons autorisé une certaine flexibilité dans la localisation et/ou le pourcentage d'identité de la duplication.

### 3.2.2 Résultats : la majorité des insertions sont *difficiles*

Nous avons caractérisé quatre catalogues de variants d'insertion, correspondant à quatre individus humains différents, trois provenant de l'étude du HGSV et un de l'étude de GiaB.

3. `https://www.ncbi.nlm.nih.gov/dbvar/content/help/#types`

Chaque catalogue contient environ 14 000 insertions, avec plus de la moitié des insertions d'un catalogue qui lui sont spécifiques. Malgré le faible nombre d'insertions partagées entre catalogues, les distributions des différentes caractéristiques des insertions sont très similaires d'un individu à l'autre. En particulier, la répartition dans les différents types d'insertion est stable même entre les deux études qui ont utilisé des méthodologies différentes pour constituer leur catalogue. Les distributions des caractéristiques des insertions sont représentées dans la Figure 3.1.

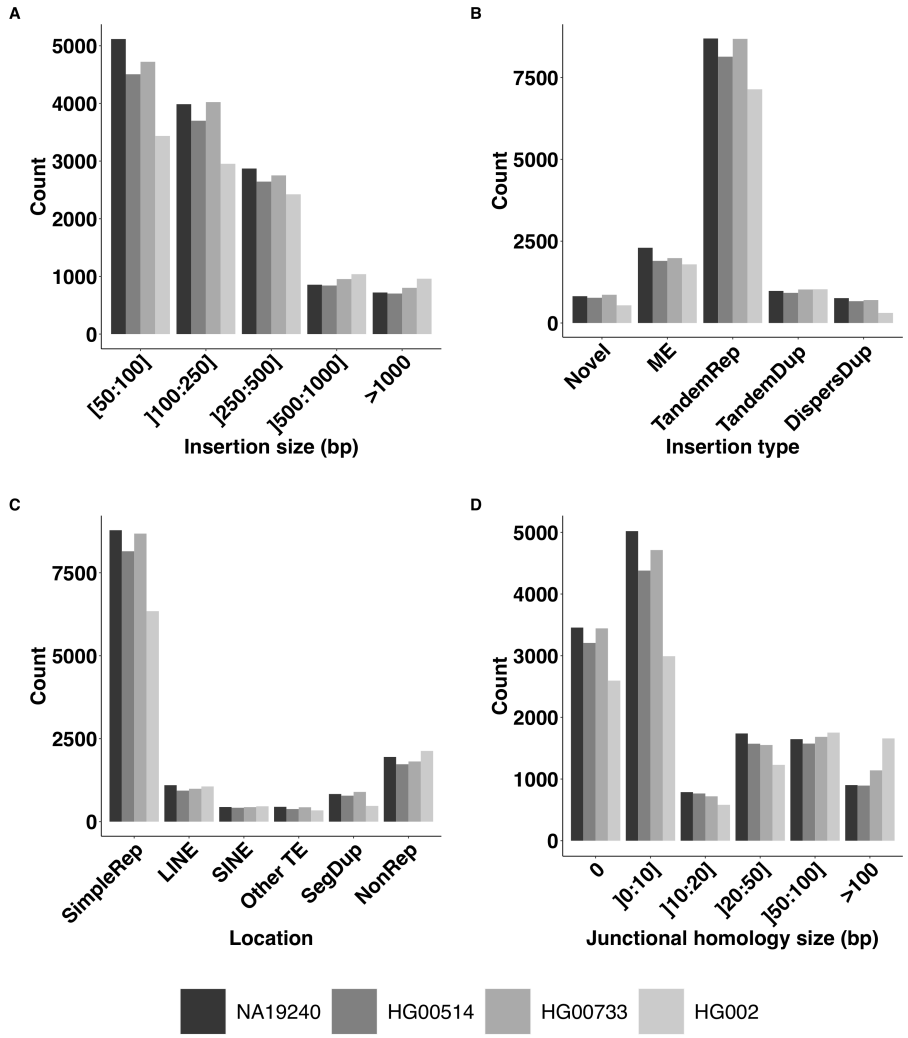


FIGURE 3.1 – Distributions des caractéristiques des variants d'insertion dans plusieurs catalogues humains. Distributions (a) de la taille des insertions, (b) du type d'insertion, (c) du contexte génomique d'insertion et (d) de la taille des homologies jonctionnelles. Abréviations : SimpleRep pour répétition simple, ME ou TE pour élément mobile/transposable, TandemRep pour répétition en tandem, TandemDup pour duplication en tandem, DispersDup pour duplication dispersée. Figure extraite de (Delage et al., 2020).

Le point remarquable de cette analyse est que la grande majorité des insertions peuvent être qualifiées de *difficiles*, c'est-à-dire qu'elles présentent des caractéristiques qui sont susceptibles de rendre leur détection difficile avec des lectures courtes. En quelques chiffres clés,

on observe que 63 % des insertions sont des expansions d'un court motif répété en tandem (répétition en tandem), le deuxième type le plus représenté étant les insertions d'éléments mobiles (16%). En terme de localisation génomique, la répartition des insertions n'est pas uniforme le long du génome. On observe un fort biais vers les régions répétées et difficilement mappables du génome, avec plus de 70% des insertions qui sont localisées dans des régions répétées dites simples (court motif répété en tandem) alors que ces régions ne représentent que 3 % du génome. Enfin, concernant la complexité des points de cassure, plus de 40 % des insertions possèdent une homologie jonctionnelle de taille supérieure à 10 pb. Là encore, c'est plus qu'attendu par chance, sur 2 000 insertions simulées uniformément le long du génome et avec une séquence aléatoire, la taille médiane d'homologie jonctionnelle mesurée est de 0 et la plus grande observée est de 7 pb.

Les différents niveaux de caractérisation, c'est-à-dire le type, la taille, le contexte génomique et l'homologie jonctionnelle, ne sont pas indépendants les uns des autres. Ainsi, on observe des distributions de taille, de localisation génomique et d'homologie jonctionnelle différentes en fonction du type d'insertion. Par exemple, la très grande majorité des insertions de type répétition en tandem sont localisées dans des répétitions en tandem. Ces insertions en particulier cumulent donc plusieurs niveaux de difficulté pour les outils de détection avec des lectures courtes et on peut se demander si la perte de rappel est plutôt due au type de la séquence insérée ou bien à son contexte génomique d'insertion.

## 3.3 Identification des causes de la perte de rappel des outils de découverte

Après avoir identifié les différentes caractéristiques des insertions humaines, nous avons voulu quantifier l'impact de chacune sur la détection des insertions par des méthodes utilisant les lectures courtes. Dans les études du HGSV et du GiaB, chaque insertion est annotée avec le type de méthode qui a permis de la détecter. On peut ainsi compter globalement que 17 et 28 % des insertions ont été détectées par au moins une méthode avec lectures courtes dans les deux études respectivement. Cependant, il est difficile d'affirmer que dans ces cas les lectures courtes à elles seules ont permis de précisément identifier et assembler l'insertion, puisque le résultat est issue de nombreuses étapes de combinaison et validation de prédictions obtenues avec plusieurs méthodes. Nous avons donc réalisé notre propre benchmark dans le but d'identifier quelles caractéristiques des insertions les rendent si difficiles à détecter avec des lecture courtes.

### 3.3.1 Conception d'un benchmark à base de simulations

Comme nous l'avons vu précédemment, les différents niveaux de caractérisation ne sont pas indépendants les uns des autres dans les catalogues d'insertions réelles. Ainsi, sur ces jeux de données, il est difficile d'évaluer l'impact du type de la séquence insérée s'il est très corrélé au contexte génomique d'insertion. C'est le cas par exemple pour les répétitions en tandem, dont la localisation génomique est très biaisée vers les régions de répétitions simples. Pour pouvoir évaluer l'impact d'une caractéristique en particulier, indépendamment des autres, nous avons proposé une stratégie basée sur la simulation de données de séquençage contenant des insertions dont les caractéristiques sont maitrisées. Nous avons proposé un benchmark comprenant un ensemble d'une vingtaine de jeux de données simulées qui se déclinent en plusieurs scénarios. Chaque jeu de données simule le séquençage à 40x avec

des lectures de 2x250 pb sur le chromosome 3 humain qui est altéré par 200 insertions dont les caractéristiques de taille, de type de séquence insérée, de localisation génomique et d'homologie jonctionnelle au point de cassure sont contrôlées.

Un premier jeu de données est appelé *référence* et correspond aux insertions les plus faciles théoriquement à détecter : des insertions de taille intermédiaire (250 pb, la taille des lectures), de type insertion nouvelle (la séquence insérée est absente du chromosome 3 humain, elle contient pas ou peu de répétitions puisqu'elle est tirée d'une exon d'un autre génome éloigné, *Sacharomyces cerevisiae*), localisée dans un contexte génomique non répété (les exons) et il n'y a pas d'homologie jonctionnelle au site d'insertion. Ensuite, nous avons décliné quatre scénarios de simulation où un seul des quatre niveaux de caractérisation est modifié par rapport à la simulation de référence :

- scénario 1 : la taille des insertions. Il contient trois jeux de données simulées avec trois tailles différentes d'insertion : 50, 500 et 1 000 pb. Pour chacun de ces jeux de données, les séquences insérées de la simulation de référence ont simplement été agrandies ou rétrécies sans changer leur localisation.
- scénario 2 : le type d'insertion. Cinq jeux de données ont été simulés avec les types suivants : duplication dispersée, duplication en tandem, insertion d'éléments mobiles (type SINE), répétitions en tandem avec deux tailles de motif (6 et 25 pb).
- scénario 3 : l'homologie jonctionnelle, avec cinq tailles testées de 10 à 150 pb. Pour simuler l'homologie jonctionnelle d'une taille $X$ donnée, le premier $X$-mer de chaque séquence insérée de la simulation de référence est remplacé par le $X$-mer flanquant à droite le site d'insertion correspondant.
- scénario 4 : le contexte génomique d'insertion. En se basant sur les annotations de RepeatMasker des séquences répétées du chromosome 3, nous avons distingué cinq contextes en plus de celui de la simulation de référence (les exons) : les régions sans répétition, les répétitions simples courtes (<300 pb) et longues (>300 pb), les éléments mobiles de type SINEs et LINEs.

Sur cette vingtaine de jeux de données simulées, nous avons appliqué quatre logiciels permettant de détecter des insertions. Nous avons choisi trois logiciels de détection de variants de structure génériques, parmi les plus récents et les plus utilisés par la communauté, qui détectent tous types de variants de structure et pas seulement les insertions, mais qui sont théoriquement les mieux adaptés aux variants d'insertions dans cette catégorie puisqu'ils utilisent tous des techniques d'assemblage local. Il s'agit de Manta (Chen et al., 2016), SVaba (Wala et al., 2018) et GRIDSS (Cameron et al., 2017). Le quatrième outil testé est naturellement l'outil que nous avons développé qui lui est dédié à ce type de variant, MindTheGap (voir Chapitre 2). Pour chaque outil et sur chaque jeu de données, nous avons mesuré le nombre d'insertions correctement prédites par rapport à la vérité simulée, soit le rappel. Il faut noter que nous avons distingué deux types de rappel en fonction de la précision des prédictions : le *rappel du site d'insertion* où seule la localisation de l'insertion est évaluée et le *rappel avec séquence résolue* où pour être considérée comme un vrai positif une prédiction doit avoir une séquence assemblée correctement (avec au moins 90 % d'identité de séquence avec la séquence simulée).

### 3.3.2 Résultats

Les résultats détaillés de ce benchmark sont présentés dans les Tableaux 1 et 2 de l'article en Annexe A.2, le deuxième tableau est repris de manière simplifiée ci-après dans la Table 3.1. Nous résumons ici quelques points remarquables. Tout d'abord, comme attendu, les ou-

tils n'ont aucun problème, avec des rappels de 100 %, si les insertions simulées sont simples comme dans la simulation de référence (de taille intermédiaire (250 pb) sans répétition dans la séquence insérée, ni dans le contexte génomique, ni au point de cassure). Cependant, dès qu'on introduit des caractéristiques plus difficiles, les rappels des outils peuvent chuter jusqu'à moins de 5 % et les valeurs sont extrêmement variables en fonction de la caractéristique simulée mais également de l'outil en question. Parmi les caractéristiques qui ont le plus d'impact, on notera que le type de la séquence insérée est plus problématique que le contexte génomique dans lequel elle s'insère, même lorsqu'on ne s'intéresse qu'au site d'insertion. Ainsi, pour répondre à la question donnée en exemple, c'est le caractère de motif répété en tandem présent dans la séquence à assembler qui semble limiter le plus la détection des répétitions en tandem plutôt que leur contexte d'insertion lui aussi composé de motifs répétés en tandem. Cependant, vraisemblablement que la combinaison de ces deux difficultés rend leur détection encore plus difficile avec des lectures courtes.

Un résultat surprenant concerne l'homologie jonctionnelle. Le rappel de tous les outils chute fortement en présence d'homologie de grande taille (>50 pb), mais même une petite taille de 10 à 20 pb peut impacter le rappel de certains outils. Cela s'explique par le fait que ces petites répétitions altèrent les signatures de mapping, comme elles altèrent les motifs de $k$-mers dans MindTheGap (voir Chapitre 2 page 23). Or, notre étude montre que ces répétitions ne sont pas si anecdotiques qu'on pouvait le penser, puisque près de 40 % des insertions de ces catalogues possèdent de telles répétitions. Les algorithmes de détection de variants de structure bénéficieraient donc de tenir compte de cette caractéristique des variants.

Le résultat le plus marquant de ce benchmark est l'absence de résolution de la séquence insérée pour la plupart des outils, même quand le site d'insertion est correctement prédit (voir Table 3.1). Ainsi, à l'exception de MindTheGap, les outils ne fournissent la séquence insérée que si elle est entièrement nouvelle et que sa taille est limitée à la taille des lectures. Dans les autres cas, le rappel obtenu est très souvent de 0 %. MindTheGap, quant à lui, n'est pas limité par la taille de la séquence et peut assembler certaines insertions duplicatives. Enfin, aucun des quatre outils testés n'est capable d'assembler les insertions de type répétitions en tandem, qui sont pourtant le type majoritaire dans les jeux de données réelles.

### 3.3.3 Quelques leçons pour MindTheGap

Concernant notre outil MindTheGap, ces résultats mettent en lumière certaines forces ainsi que des faiblesses sur lesquelles on peut travailler. Les points forts de MindTheGap se situent au niveau de la résolution de séquences des grandes insertions nouvelles et des duplications dispersées. Globalement, les rappels de MindTheGap sur le site d'insertion sont souvent plus faibles que ses concurrents mais cela est du au fait que MindTheGap ne renvoie pas les sites d'insertion qu'il n'a pas réussi à assembler. Certains types d'insertions sont complètement absents des résultats de MindTheGap : ce sont les duplications en tandem, les expansions de répétitions en tandem et les insertions avec des grandes homologies jonctionnelles. Pour les expansions de motifs courts en tandem, c'est l'assemblage qui pose problème car ces séquences forment des cycles dans le graphe de de Bruijn et l'algorithme actuel ne permet pas de parcourir plusieurs fois un même sommet du graphe. Dans les autres cas, la perte de rappel se fait dès le module Find, car le motif de $k$-mers est altéré par le type d'insertion. C'était notamment attendu pour les homologies jonctionnelles qui ne sont prises en compte par défaut que si leur taille est inférieure à 5 pb. La flexibilité du motif vis-à-vis de telles répétitions est paramétrable (paramètre `-max-repeat`) mais elle se limite

| | | Rappel avec séquence résolue | | | |
|---|---|---|---|---|---|
| | | **GRIDSS** | **Manta** | **SvABA** | **MindTheGap** |
| **Simulation de référence** | | 81 | 100 | 96 | 100 |
| **Scenario 1 Taille d'insertion** | 50 pb | 56 | 100 | 100 | 100 |
| | 500 pb | 0 | 0 | 0 | 99 |
| | 1 000 pb | 0 | 0 | 0 | 98 |
| **Scenario 2 Type d'insertion** | Dup. dispersée | 0 | 0 | 16 | 96 |
| | Dup. tandem | 0 | 0 | 0 | 0 |
| | Element transp. | 0 | 0 | 61 | 58 |
| | Rep. tandem (motif de 6 pb) | 0 | 0 | 1 | 0 |
| | Rep. tandem (motif de 25 pb) | 0 | 0 | 0 | 0 |
| **Scenario 3 Homologie jonct.** | 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 génomique** | Non répété | 77 | 97 | 93 | 83 |
| | Rep. tandem (<300 pb) | 77 | 98 | 97 | 73 |
| | Rep. tandem (>300 pb) | 77 | 93 | 90 | 58 |
| | SINE | 77 | 99 | 94 | 53 |
| | LINE | 76 | 97 | 95 | 89 |

TABLE 3.1 – Rappel avec séquence résolue de plusieurs logiciels de détection d'insertions avec des lectures courtes en fonction des différents scénarios de simulation. Les cellules du tableau 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 gris montrent peu de différence ou une amélioration du rappel. Tableau repris de (Delage et al., 2020)

bien évidemment à la valeur de $k$ et entraine une perte de spécificité du motif, ce qui peut générer un grand nombre de faux positifs. Enfin, une duplication en tandem d'une séquence à l'identique est un cas extrême d'une très grande homologie jonctionnelle, ce qui explique l'absence du motif de $k$-mers pour ce type d'insertion. Ainsi, ces résultats suggèrent que l'approche par $k$-mers du module Find de MindTheGap atteint ses limites pour détecter certains types de sites d'insertions, et ceux-ci sont malheureusement les plus fréquents dans les catalogues d'insertions humaines.

## 3.4 Conclusion

Ce travail représente la première caractérisation des variants de type insertion à grande échelle chez l'homme. Il a été permis grâce à l'exhaustivité, la précision et la résolution de séquence de catalogues de variants de structure obtenus seulement récemment grâce aux nouvelles technologies de séquençage en lectures longues. Le résultat marquant de cette caractérisation est la très grande diversité des insertions, tant du point de vue de leurs caractéristiques génomiques, que du point de vue de leur capacité à être détectées avec les lectures courtes.

Si l'inadéquation des lectures courtes pour la découverte de variants de structure en général était déjà bien reconnue, notre étude met en lumière l'ampleur du problème pour les variants de type insertion en particulier et plaide pour une meilleure évaluation des outils. Les faibles valeurs de rappel obtenues dans notre benchmark contrastent avec les bonnes performances montrées dans les publications des outils ou dans les précédents benchmarks publiés pour évaluer les outils de détection des variants de structure. Cela provient d'un manque de réalisme des données simulées et/ou d'un biais de représentativité des vrais variants utilisés pour la validation. Ainsi, en ne considérant que certains types d'insertions, et en particulier les plus *faciles*, les évaluations précédentes ont largement sur-estimé les capacités des outils actuels.

Le tableau n'est pourtant pas si noir, car les performances des outils ne sont pas si catastrophiques pour tous les types d'insertion et les insertions les plus difficiles ne sont pas forcément celles qui sont le plus recherchées dans les applications biologiques. Cependant, une évaluation plus réaliste, qui tient compte des différents types d'insertion notamment, permettra à l'utilisateur de faire un choix d'outil plus éclairé en fonction de ses besoins.

Cette étude a permis également d'identifier plus précisément les facteurs responsables d'une telle perte de rappel et de suggérer des pistes pour améliorer les algorithmes de détection des insertions avec des lectures courtes, et en particulier pour notre outil MindTheGap. De plus, la variabilité des rappels entre les outils montre que bien souvent le signal du variant est présent même avec des lectures courtes et qu'il peut être détecté, mais que les différents outils n'appliquent pas les mêmes filtres. Cela donne donc de l'espoir de pouvoir faire mieux, soit en améliorant les algorithmes grâce à l'analyse fine de ces résultats, soit en cherchant une combinaison des différents outils qui prendrait le meilleur de chacun et maximiserait le rappel. À l'heure actuelle, les méthodes dites *meta* ou de *consensus* qui combinent plusieurs outils ont pour principal objectif de réduire le nombre de faux positifs et donc effectuent simplement des intersections entre les ensembles de prédictions. Effectuer des unions permettrait d'améliorer la sensibilité mais augmenterait considérablement le nombre de faux positifs. Une étude, similaire à celle effectuée ici, non pas sur le rappel mais sur la précision des outils permettrait probablement d'identifier des combinaisons d'outils plus intelligentes et de réduire le nombre de faux positifs. Cependant, pour cela, il faudrait caractériser les prédictions et non plus les vraies insertions et cela n'est possible que si les séquences insérées ont été assemblées. Il faut donc travailler dans un premier temps à améliorer la résolution de séquence des prédictions. Pour cela, notre benchmark montre que l'outil MindTheGap, et en particulier son module d'assemblage Fill, est bien placé dans la compétition et pourrait avantageusement s'intégrer dans une méthode qui combine plusieurs outils.

Toutefois, compte tenu de ces résultats, on peut se demander s'il est rentable de consacrer des efforts sur ces méthodes alors que les lectures longues se généralisent. C'est une question que nous discuterons dans le Chapitre 5 Section 5.1.

Ce travail représente une partie du travail de thèse de Wesley Delage que j'ai co-encadré. Les résultats ont été publiés récemment dans BMC Genomics (Delage et al., 2020) et les outils développés pour l'annotation des insertions et l'évaluation des outils sont diffusés à la communauté sur github [4].

4. `https://github.com/WesDe/SVAn` pour l'annotation des insertions et `https://github.com/WesDe/InserSim` pour la simulation des différents scénarios

# Chapitre 4

# Génotypage des Variants de Structure avec des lectures longues

Dans ce chapitre, nous présentons une méthode de génotypage des variants de structure avec des lectures longues, qui a été développée dans le cadre de la thèse de Lolita Lecompte (2017-2020) et qui a été publiée en 2020 dans la revue *Bioinformatics* (Lecompte et al., 2020) (publication en Annexe A.3).

## 4.1 Motivations et contexte

Une fois les variants de structure détectés et caractérisés pour un ou plusieurs individus, une question se pose naturellement lorsque d'autres individus sont séquencés plus tard : ces variants de structure sont-ils présents dans ces nouveaux génomes et si oui dans quel état (homozygote ou hétérozygote pour des individus diploïdes). C'est le problème du *génotypage*. Cette question est particulièrement prégnante lorsqu'on recherche des associations significatives entre des variations génétiques et des phénotypes avec de nombreux individus (GWAS), ou dans des études de génétique des populations. Ces deux types d'études prennent généralement en entrée des données sous la forme d'une matrice avec en ligne les différents variants identifiés et en colonne, pour chaque individu re-séquencé, la fréquence allélique ou le génotype de chaque variant.

### 4.1.1 Le problème du génotypage

D'un point de vue méthodologique, on distingue donc deux problèmes : la découverte des variants et le génotypage des variants. Dans le premier, on dispose d'un génome de référence et d'un ensemble de lectures de séquençage représentant le génome d'un autre individu. Le problème est d'identifier tous les variants d'un certain type, ici les variants de structure, qui différencient le génome re-séquencé du génome de référence. Dans le problème du génotypage, une troisième donnée compose l'entrée du problème : un ensemble de variants connus et bien caractérisés, c'est-à-dire avec leur position dans le génome de référence, et la séquence des allèles alternatifs. Le problème est d'assigner un génotype ou une fréquence allélique à chaque variant, c'est-à-dire estimer la quantité relative de chaque allèle de chaque variant dans l'ensemble des lectures de séquençage. Puisqu'on dispose de plus d'informations *a priori*, le problème de génotypage semble plus facile que celui de la découverte où l'espace des possibles est beaucoup plus grand.

Les deux problèmes sont étroitement liés et sont parfois confondus. La confusion peut provenir du fait que les méthodes de découvertes possèdent souvent un module de génotypage permettant d'enrichir l'information des variants découverts avec le génotype ou la quantification des allèles dans les échantillons ayant permis leur découverte. Cependant, dans cette situation, seuls les variants découverts sont génotypés. Dans l'hypothèse où les outils de découvertes seraient extrêmement sensibles, on pourrait supposer que tous les variants non détectés sont absents et déduire ainsi leur génotype. Cependant, nous avons vu que cette hypothèse est loin d'être valide, il est donc important d'avoir une méthode capable de vérifier la présence mais aussi l'absence d'un variant donné. De même, les méthodes de découvertes produisant de nombreux faux positifs, elles demandent une profondeur de séquençage plus importante pour valider la découverte d'un variant qu'elles n'en auraient besoin pour simplement quantifier la présence d'un variant déjà validé par ailleurs. En effet, dans de nombreuses applications, on effectue la découverte des variants avec un petit ensemble d'individus fortement couverts, puis on les génotype dans des plus grands ensembles d'individus plus faiblement couverts. Enfin, la comparaison de variants de structure prédits indépendamment chez des individus différents est une tâche complexe, car il n'est pas rare qu'un même variant soit prédit et décrit différemment par un même outil pour des jeux de lectures différents. Le génotypage permet de s'affranchir de cette étape de comparaison et d'identification des variants communs entre échantillons et d'obtenir des informations quantitatives pour chaque échantillon basées sur la même représentation des variants pour tous les individus.

### 4.1.2 État de l'art : pas d'outil pour les lectures longues

Naturellement, les développements méthodologiques sur l'étude des variants de structure se sont d'abord portés sur le problème de découverte : avant de pouvoir quantifier des variants, il faut les avoir détectés. Si rapidement, les outils de découverte des variants se sont dotés de modules de génotypage des variants prédits, jusqu'à récemment, il existait peu de méthodes dédiées exclusivement à la tâche de génotypage et qui permettaient de génotyper n'importe quel variant et pas seulement ceux qui sont découverts dans l'échantillon en entrée. Ainsi les premières publications qui présentent des outils de génotypage datent de 2018 (pour SV2 (Antaki et al., 2018) et BayesTyper (Sibbesen et al., 2018)), près de dix ans après les premières publications d'outils de découverte des variants de structure. Avant 2018, il existait néanmoins deux outils de génotypage, DELLY (Rausch et al., 2012) et SVtyper (Chiang et al., 2015). Le premier est en fait un outil de découverte qui s'est doté dans un second temps d'un outil de génotypage, mais ce dernier n'a jamais été décrit dans une publication. SVtyper est publié en 2015, mais n'est pas l'objet principal de la publication qui présente en fait une suite d'outils pour le mapping et la découverte de variants de structure, celle de Lumpy. Ces deux outils permettent en théorie de génotyper n'importe quel variant de structure, mais en pratique, ayant été développés en étroite association avec un outil particulier de découverte, ils sont difficilement utilisables avec des fichiers de variants obtenus avec d'autres outils.

Puis, avec l'augmentation de la quantité et de la qualité des variants de structure prédits et catalogués grâce aux technologies de séquençage de troisième génération, on a assisté dans les années 2019-2020 à une multiplication des méthodes de génotypage pour les lectures courtes. D'un point de vue méthodologique et algorithmique, les premiers outils de génotypage se caractérisent par le fait qu'ils se basent exclusivement sur l'alignement des lectures courtes sur le génome de référence et quantifient les signaux de mapping aberrants classi-

quement utilisés pour la découverte des variants (paires de reads discordantes, split-reads, profondeur de séquençage). En conséquence, la séquence de l'allèle alternatif des variants n'est pas représentée, ni utilisée pour la quantification. Cela induit un biais en faveur de l'allèle de référence, et empêche de génotyper les variants de type insertions nouvelles pour lesquels la séquence du variant alternatif est complètement absente du génome de référence. La plupart des méthodes plus récentes ont pour objectif d'éliminer ce biais et ont bénéficié de l'essor des représentations de génomes par des graphes. Dans un graphe de génomes ou de variations, les sommets sont des séquences et les arêtes des chevauchements ou adjacences de séquences observés dans au moins un allèle ou génome. Chaque allèle ou haplotype est représenté par un chemin dans le graphe. Ainsi, contrairement à un génome de référence qui ne représente qu'un seul allèle, avec ce type de graphe on dispose d'une structure pouvant représenter tous les allèles connus d'un ensemble de variants. Lors du mapping des lectures sur le graphe, il n'y a pas *a priori* de biais de mapping vers la référence. Les génotypeurs les plus récents, tels que Paragraph (Chen et al., 2019) et graphTyper2 (Eggertsson et al., 2019), exploitent ces structures de données. Cependant, la tâche de mapping des lectures est plus complexe et plus coûteuse sur un graphe. Pour ne pas avoir à mapper la totalité des lectures sur le graphe du génome complet, une pré-sélection des lectures d'intérêt est effectuée à partir du mapping sur le génome de référence avant d'effectuer le mapping sur le graphe. Le biais vers la référence n'est donc pas complètement éliminé.

L'ensemble des méthodes présentées précédemment ont été développées spécifiquement pour les lectures courtes. En 2018, lorsque nous avons commencé ce travail, il existait plusieurs méthodes de découvertes des variants de structure dédiées aux lectures longues, mais aucune méthode de génotypage n'avait été décrite dans la littérature. Deux outils permettaient cependant le génotypage de variants de structure avec des lectures longues, mais ils n'ont pas été dédiés à cette tâche : Sniffles (Sedlazeck et al., 2018b) est un outil de découverte de variants de structure qui possède une option de génotypage (-Ivcf) non décrite dans la littérature, et svviz2 (Spies et al., 2015) est un outil de visualisation d'alignements qui estime des génotypes en sous-produit de sortie. Pour ces deux outils, là encore, il existe un biais vers la référence puisque les lectures ne sont alignées que sur le génome de référence pour le premier, et sont pré-sélectionnées avec la référence dans le deuxième.

Dans ce travail, nous avons donc proposé la première méthode dédiée au génotypage des variants de structure avec des lectures longues. Cette méthode est implémentée dans le logiciel SVJedi et a été publiée dans le journal Bioinformatics en 2020 (article en Annexe A.3).

## 4.2 La méthode : SVJedi

### 4.2.1 Principe et originalité

La méthode de génotypage que nous avons proposée est basée sur l'alignement des lectures longues sur des séquences représentatives des deux allèles de chaque variant de structure à génotyper. Pour un variant de structure donné, on définit ses séquences représentatives par ses *points de cassure*, c'est-à-dire des adjacences de séquence qui sont spécifiques de l'un ou l'autre allèle, accompagnés de séquences flanquantes d'une taille paramétrable fixée par défaut à 5 Kb. Les nombres de lectures alignées sur l'un et l'autre allèle sont comparés et permettent d'estimer la fréquence allélique du variant, puis son génotype dans le cas d'un individu diploïde par une méthode statistique classique de maximum de vraisemblance. Les différentes étapes sont schématisées dans la Figure 4.1.

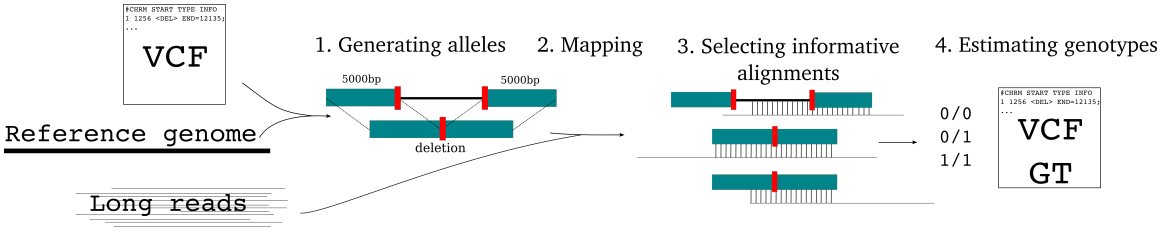


FIGURE 4.1 – Les différentes étapes de la méthode SVJedi. Figure extraite de (Lecompte et al., 2020).

L'originalité de cette méthode est double : d'une part la représentation des allèles qui donne le même poids à chacun des deux allèles de chaque variant de structure et d'autre part l'alignement des lectures sur une petite sous partie du génome au lieu du génome complet. La représentation des allèles permet de s'affranchir complètement du biais vers la référence et la limitation de la séquence de référence aux séquences contenant des différences de structure à génotyper permet de gagner en temps de calcul. Ces choix contrastent notamment avec les méthodes existantes pour les lectures courtes, ils sont en effet largement influencés par le type de lecture de séquençage à aligner : les lectures étant longues de plusieurs Kb, elles s'alignent de manière plus spécifique que les lectures courtes de type Illumina. Ainsi, elles sont moins sujettes aux problèmes de multi-mapping le long du génome de référence (plusieurs positions génomiques équivalentes en terme de score d'alignement) et de sur-mapping. Le sur-mapping est un problème qui apparait si la séquence de référence sur laquelle les lectures sont mappées est incomplète : le mapper cherche à maximiser le nombre de lectures alignées et force l'alignement de lectures provenant de ces régions manquantes à d'autres *loci* représentés dans la séquence de référence. Ces alignements non légitimes issus de multi-mapping ou de sur-mapping sont la principale source d'erreurs de génotypage et doivent donc être évités.

En ne mappant que sur les séquences représentatives des allèles des variants de structure, on gagne en temps de calcul car les lectures longues de part leur taux et leur type d'erreurs de séquençage peuvent être coûteuses à aligner sur le génome entier. En contre partie, on s'expose au problème de sur-mapping, c'est-à-dire d'aligner des lectures sur des séquences représentatives de variant de structure alors qu'elles proviennent d'une autre région du génome qu'on ne souhaite pas génotyper. Une étape importante de la méthode consiste donc à identifier et éliminer ces alignements non légitimes parmi les résultats d'alignement. Pour cela, nous avons implémenté des filtres basés sur le score de qualité de mapping et sur l'analyse des coordonnées des alignements sur les lectures et les séquences de référence. En particulier, un alignement, pour être considéré comme légitime et informatif, doit couvrir au moins un point de cassure et doit être semi-global, c'est-à-dire qu'il implique les extrémités soit de la lecture soit de la séquence de référence (voir les détails dans l'article présenté en annexe).

### 4.2.2 Implémentation

La principale difficulté algorithmique dans cette méthode est l'alignement des lectures longues sur les séquences représentatives des variants de structure. Les alignements recherchés sont des alignements locaux ou semi-globaux permettant de détecter des similarités faibles jusqu'à 70 % d'identité environ avec un nombre important d'insertions et délétions

qui correspondent aux profil des erreurs de séquençage des données de lectures longues, telles que celles obtenues avec les technologies PacificBioscience et Nanopore. Ce problème a fait l'objet de recherche et développement dès l'apparition de ces lectures et est très bien résolu par les outils de mapping actuels dédiés aux lectures longues. Parmi eux, minimap2 (Li, 2018) est l'un des plus populaires et probablement le plus rapide. Il est également facile d'utilisation et d'intégration dans un pipeline. Nous avons donc estimé qu'il n'était pas nécessaire de ré-implémenter un mapper dédié.

SVJedi a ainsi été implémenté en Python comme un pipeline faisant intervenir séquentiellement 3 outils ou modules :

1. l'interprétation du fichier de variants de structure en entrée et la génération des séquences représentatives,
2. le mapping des lectures longues sur les séquences représentatives avec minimap2,
3. l'analyse des alignements pour sélectionner les alignements informatifs et l'estimation des génotypes.

L'implémentation modulaire permet de ne pouvoir effectuer que certaines parties du pipeline et de ré-utiliser des résultats intermédiaires pour d'autres runs. Par exemple, la représentation des variants de structure peut ainsi être ré-utilisée pour différents individus à génotyper. Elle permet également de pouvoir interchanger le mapper de lectures longues.

## 4.3 Résultats

### 4.3.1 Données pour l'évaluation de la méthode

Au moment du développement de la méthode, il n'existait pas de jeux de données réelles, du moins pour un génome complexe tel que le génome humain, permettant d'évaluer précisément la qualité de nos estimations des génotypes. En effet, nous avons besoin de données de séquençage en lectures longues pour au moins un individu pour lequel on connait les génotypes d'un ensemble représentatif et bien caractérisé de variants de structure. Comme nous l'avons vu précédemment, des ensembles exhaustifs, ou au moins représentatifs, et bien caractérisés de variants de structure sont rares ou très récents. Chez l'homme, la base de données dbVar est biaisée vers les variants de structure de type délétions, identifiées principalement avec des lectures courtes, et ce n'est qu'en 2019 qu'ont été rendus disponibles les callsets du HGSV et de GiaB pour quatre individus. Parmi ces différents jeux de données, seul le jeu de données du GiaB dispose de l'information de génotype pour un individu (HG002) et nous l'avons utilisé dans un second temps pour évaluer notre méthode.

Ainsi, dans un premier temps, nous avons donc généré des données simulées pour développer, optimiser et valider la méthode. Le principe de la génération d'un jeu de données simulées est le suivant : on définit un ensemble de variants de structure à génotyper sur un génome donné, on génère deux haplotypes à partir de ce génome en incorporant dans chacun un sous-ensemble de l'ensemble des variants de structure à génotyper et on simule un séquençage de lectures longues sur ce génome diploïde synthétique. Pour un variant de structure donné, son absence ou sa présence dans un ou les deux haplotypes définit son génotype dans l'individu simulé. Nous avons effectué ces simulations sur le chromosome 1 humain et nous avons sélectionné des variants de structure déjà caractérisés dans ce chromosome grâce à la base de données de dbVar lorsque c'était possible, c'est-à-dire pour les variants de structure de type délétion. La majorité des simulations a été effectué avec ce

type de variant de structure qui est également le plus fréquent et le mieux représenté dans les bases de données.

Cette approche de simulation permet non seulement d'évaluer précisément la qualité des estimations des génotypes puisque la vérité est connue, mais également d'évaluer la robustesse de la méthode vis-à-vis de différentes caractéristiques des données que l'ont peut facilement faire varier dans le processus de simulation. Ainsi, nous avons fait varier dans nos simulations le type de variant de structure, la technologie de séquençage, la profondeur de séquençage, le taux d'erreurs de séquençage et la précision de caractérisation des variants. L'évaluation se base sur deux métriques : le taux de génotypage et la précision de génotypage. Le premier est le pourcentage de variants pour lesquels un génotype a été assigné (dans SVJedi, la raison pour laque un variant n'est pas génotypés est un nombre insuffisant de lectures mappées de manière informative sur ses allèles). La précision de génotypage est le pourcentage de variants, parmi les variants génotypés, dont l'estimation du génotype est correcte.

### 4.3.2 SVJedi, une méthode efficace et robuste

Les résultats sur données simulées montrent que SVJedi estime correctement le génotype pour plus de 97 % des variants, quelque soit le type de variant de structure. La précision du génotypage reste très haute, au dessus de 95 %, lorsqu'on altère la qualité ou la quantité des données en entrée, par exemple si on augmente le taux d'erreurs de séquençage, si on diminue la couverture de séquençage ou si les points de cassures sont moins précis. La couverture de séquençage a néanmoins un impact attendu sur le taux de génotypage, c'est-à-dire la proportion de variants pour lesquels un génotype est estimé. Si la couverture est trop faible, la méthode préfère renvoyer une valeur manquante que prendre le risque de se tromper. Une couverture de 10X est cependant suffisante pour génotyper plus de 90 % des variants avec plus de 95 % de précision.

Sur les données réelles de l'individu humain HG002 (données de Genome in a Bottle), le taux de génotypage est plus faible. Avec 30X de données PacBio, 90 % des 12 745 délétions et insertions sont génotypées par SVJedi. Parmi ces 90 %, 92 % obtiennent le même génotype que celui estimé par Genome in a Bottle. Même si les génotypes données par Genome in a Bottle sont également des prédictions d'outils bioinformatiques, on peut les considérer comme vérité étant donné le grand nombre d'outils utilisés et combinés. On obtient donc une précision également plus faible de 92 % avec les vraies données. Ces différences de résultats entre les données réelles et simulées s'expliquent principalement par les différences de caractéristiques des variants de structure à génotyper. Comme nous l'avons vu dans le chapitre précédent, ces nouveaux jeux de variants de structure obtenus avec des technologies longues lectures sont plus exhaustifs et ont révélé des variants d'une plus grande complexité par rapport aux données précédentes dans dbVar. En effet, on observe que la très grande majorité des variants de structure qui ne sont pas génotypés ou mal génotypés par SVJedi correspondent à des catégories particulières de variants de structure : les variants de petite taille (<100 pb) et ceux localisés dans un contexte de répétition en tandem. Pour ces différents types de variants, la précision de SVjedi peut baisser jusqu'à 81 % et le taux de génotypage jusqu'à 70 %. Sur les grands variants de structure situés en dehors des répétitions en tandem, SVjedi obtient un taux et une précision de génotypage similaires à ceux obtenus sur les données simulées (99 et 98 % respectivement pour le taux et la précision).

### 4.3.3 Comparaison avec d'autres approches

Nous avons comparé ces résultats avec ceux obtenus par d'autres approches, et dans un premier temps les deux outils directement concurrents utilisant des lectures longues : l'outil de découverte de variants de structure Sniffles (Sedlazeck et al., 2018b) avec l'option `-Ivcf`, et l'outil de visualisation des variants de structure svviz2 (Spies et al., 2015). Sur le jeu de données réelles du GiaB, ces deux outils estiment un génotype pour presque tous les variants (taux de génotypage de 100 % ou presque), mais au prix d'une précision bien plus faible que SVJedi : 82 et 66 % de précision pour Sniffles et svviz2 respectivement, comparé à 92 % pour SVJedi.

SVJedi se démarque aussi nettement de ses concurrents concernant le temps de calcul : pour génotyper les 12 745 variants de structure de GiaB avec un re-séquençage PacBio à 30X d'un génome humain, le temps de calcul est seulement de 2h25 avec 40 cœurs. La très grande majorité du temps est pris par l'alignement des lectures avec minimap2 (2h15). Surtout, SVjedi est 7 fois plus rapide que Sniffles (17h15) et 50 fois plus rapide que svviz2 (plus de 5 jours, mais il ne dispose pas d'une version parallélisée).

Pour ce jeu de données, il existe également des données de lectures courtes Illumina. La comparaison de SVJedi avec une approche de génotypage avec des lectures courtes montre clairement l'apport des lectures longues pour la qualité du génotypage, puisque l'un des meilleurs génotypeurs de lectures courtes, SVtyper (Chiang et al., 2015), a assigné un mauvais génotype à plus de la moitié des variants (précision de 46 %).

Enfin, la dernière comparaison a pour objectif d'évaluer l'apport d'une approche dédiée de génotypage par rapport à une approche de découverte qui génotype les variants détectés. Dans cette approche, seuls les variants correctement découverts peuvent être génotypés et le taux de génotypage correspond donc au rappel de détection des méthodes. Pour ce jeu de données, en moyenne seulement la moitié des variants de structure ont pu être correctement prédits par Sniffles ou PBsv. Plus surprenant, parmi les variants détectés la précision de génotypage obtenue est plutôt faible, 44 et 78 % respectivement pour Sniffles et PBsv. Cette expérience montre donc que la tâche de génotypage est bien différente de celle de la découverte et qu'elle mérite d'avoir ses méthodes dédiées.

L'ensemble de ces résultats sont détaillés par type de variant de structure dans la Table 4.1.

## 4.4 Vers l'utilisation de graphes pour représenter les variants

La méthode de génotypage présentée ici repose sur une bonne représentation des allèles des variants de structure. Une limitation actuelle de la méthode réside dans le fait que chaque variant est représenté et génotypé indépendamment des autres. Cela ne pose pas de problème lorsque les variants sont bien espacés les uns des autres sur le génome (typiquement au moins 1 Kb de distance) ou lorsque le catalogue de variants est bien *propre* et que chaque variant est bien représenté qu'une seule fois (pas de redondance). C'était le cas notamment du catalogue de l'individu HG002 du GiaB qui est issu de longs efforts de curation et dont certains variants trop proches ont été volontairement éliminés par manque de certitude sur leur qualité. Cela ne reflète donc pas tous les cas d'utilisation. En particulier, la distance entre deux variants de structure successifs peut être plus petite dans la réalité et est amenée à se réduire encore lorsque le catalogue ne représente plus un seul individu mais toute une population.

Lorsque des variants sont proches, voire chevauchants, nous observons une forte baisse

| | Délétions | | Insertions | | |
|---|---|---|---|---|---|
| Outil | précision | taux | précision | taux | temps |
| SVJedi | **91,7** | 85,8 | **92,5** | 93,6 | **2h25m** |
| Sniffles-`Ivcf` | 82,5 | **99,9** | 81,7 | **99,8** | 17h16m |
| svviz2 | 72,5 | **100** | 61,0 | **100** | 5 jours* |
| SVtyper (Illumina) | 46,5 | 99,2 | - | - | 5h32m |
| Sniffles (découverte) | 48,7 | 52,4 | 39,8 | 44,8 | 18h04m |
| pbsv | 90,1 | 72,7 | 68,8 | 59,8 | 5h29m |

TABLE 4.1 – Comparaison de plusieurs outils et approches pour le génotypage des 12 745 délétions et insertions du catalogue du GiaB chez l'individu HG002. Trois approches sont comparées : des outils de génotypage pour lectures longues (trois premiers outils), un outil de génotypage pour lectures courtes (SVTyper) et des outils de découverte pour lectures longues (deux derniers outils). À l'exception de SVtyper qui utilise un jeu de données de séquençage Illumina 30X, tous les autres outils ont été exécutés avec un jeu de données de lectures longues PacBio 30X. Les temps d'exécution ont été mesurés sur un nœud de calcul de 40 CPU. * svviz2 n'est pas parallélisé. Table reprise de (Lecompte et al., 2020).

du taux de génotypage de SVJedi, avec 20 à 30 % de variants de structure non prédits si la distance est inférieure à 50 pb ou si les variants se chevauchent. Ainsi, si on ajoute dans le catalogue de variants des variants proches des variants initiaux, même s'ils sont absents dans l'individu re-séquencé, ils gênent le génotypage des variants initiaux qui étaient bien génotypés seuls. Cela s'explique par le fait que plusieurs variants partagent alors des séquences communes dans leurs représentations et de nombreuses lectures ne sont alors plus considérées pour le génotypage du fait qu'elles s'alignent correctement sur deux variants différents (multi-mapping).

Pour palier ce problème, l'idée est de construire une unique représentation des allèles pour une même région contenant des variants proches ou chevauchants. Pour cela, et pour pouvoir représenter l'ensemble des haplotypes possibles, nous utilisons une représentation sous la forme d'un graphe de séquences à la place de représentations linéaires indépendantes. Plusieurs étapes du pipeline original de SVJedi doivent alors être adaptées à cette nouvelle représentation : l'étape de mapping des lectures longues et l'interprétation des alignements obtenus. Une première implémentation a été effectuée par Sandra Romain cette année dans le cadre de son stage de master 2, dans un nouveau pipeline appelé SVJedi-graph. Il fait notamment intervenir les outils VG-toolkit (Garrison et al., 2018) et GraphAligner (Rautiainen and Marschall, 2020). Les premiers résultats obtenus sur des données simulées sont très prometteurs puisqu'ils permettent de retrouver les performances de SVJedi obtenues sur des variants isolés quelque soit la distance ou le chevauchement des variants. Cette approche permettra très certainement d'améliorer les performances de SVJedi sur des jeux de données plus réalistes.

## 4.5 Conclusion

Alors que la découverte des variants de structure est nettement améliorée par les technologies de séquençage en lectures longues, les catalogues de variants de structure s'enrichissent pour de nombreux organismes. Le génotypage de ces variants dans des nouveaux séquençages individuels est devenu une problématique importante et nous avons assisté au développe-

ment de nombreuses méthodes bioinformatiques ces dernières années pour effectuer cette tâche. Cependant, aucune méthode utilisant des données de lectures longues n'existait et nous avons ainsi proposé la première, appelée SVJedi.

Cette méthode est dédiée aux lectures longues et exploite leur spécificité : la grande longueur des lectures rend leur mapping bien plus spécifique que les lectures courtes et nous permet de s'affranchir du mapping sur le génome complet. À la place, les différents allèles de chaque variant sont explicitement représentés, ce qui permet d'éviter le biais vers la référence qui est un défaut classique des méthodes utilisant les lectures courtes. Les résultats obtenus sur des données simulées et réelles humaines montrent que cette nouvelle approche de génotypage est efficace et rapide. Ce travail constitue le travail de thèse de Lolita Lecompte, il a été publié récemment dans la revue Bioinformatics (Lecompte et al., 2020) (voir Annexe A.3). Le logiciel est diffusé sous licence libre sur github [1] et distribué dans Bioconda [2]. Les améliorations avec la représentation en graphes de séquences sont en cours de validation et de diffusion avec l'outil SVJedi. Une des perspectives de ce travail que j'envisage à court terme exploitera encore la représentation par graphe de séquences pour estimer non plus les génotypes, mais les haplotypes de variants de structure proches, c'est-à-dire comment les différents allèles des variants proches sont liés sur les chromosomes homologues.

1. `https://github.com/llecompte/SVJedi`
2. `https://anaconda.org/bioconda/svjedi`

# Chapitre 5

# Discussion et perspectives

Dans ce document, j'ai choisi de présenter parmi mes différentes contributions en bioinformatique des séquences, celles qui portent sur une thématique particulière : l'étude des variants de structure et ses problématiques méthodologiques. C'est une thématique qui me motive particulièrement depuis de nombreuses années, et sur laquelle je veux poursuivre mes recherches futures.

C'est un domaine de recherche qui est en plein essor en ce moment car, d'une part les technologies de séquençage sont plus adaptées pour détecter les variants de structure, et d'autre part, après avoir étudié en profondeur les variations ponctuelles, les biologistes se tournent désormais vers des variants plus complexes pour expliquer la variabilité phénotypique restante. Ainsi, de plus en plus d'études montrent l'impact de ces variants sur des traits phénotypiques. Un exemple récent d'une telle étude a été publié récemment sur la tomate (Alonge et al., 2020), cette étude identifie notamment de nombreux variants de structure qui modifient des traits d'intérêt agronomique tels que la saveur du fruit, sa taille ou la productivité de la plante. La recherche méthodologique pour l'étude de ces variants est par conséquent un domaine très compétitif, et qui évolue très vite du fait de l'évolution des technologies de séquençage. Ainsi, dans ce chapitre, je discuterai dans un premier temps l'évolution de l'utilisation des différentes technologies de séquençage pour ces questions avant de présenter les perspectives de mes travaux et mes axes de recherche actuels et futurs.

## 5.1 Est-ce la fin des lectures courtes pour étudier les variants de structure ?

La détection des variants de structure avec des données de séquençage est un problème complexe et non résolu pour tous les types de variants et tous les types de données de séquençage. Depuis plusieurs années, nous avons observé et quantifié l'inadéquation des lectures courtes pour détecter et analyser ce type de variations génomiques. Le chapitre 3 en particulier, montre bien la faible sensibilité de ces données (et de leurs méthodes associées) pour une majorité des variants de type insertion. L'amélioration des technologies de séquençage permettant d'obtenir des lectures longues de plusieurs dizaines de Kilobases voire des Mégabases a permis de considérablement augmenter la sensibilité de détection et réduire le nombre de fausses prédictions. La réduction actuelle des taux d'erreur de ces lectures promet d'améliorer encore les performances de détection des variants de structure avec ces lectures. Cependant, ces technologies restent encore très coûteuses et pour de nombreuses applications, les technologies moins chères seront encore probablement largement utilisées.

Par exemple, les études de recherche d'association avec des phénotypes et les études de génomique des populations nécessitent d'analyser des grands nombres d'individus, typiquement plusieurs centaines. À part pour des projets sur quelques espèces modèles ou des espèces à fort intérêt économique, le coût des technologies ONT et PacBio (et encore plus pour les données PacBio HiFi) reste pour le moment prohibitif pour de telles études (Coster et al., 2021). Ainsi, une stratégie moins coûteuse pour des projets à grands nombres d'échantillons est de partitionner les échantillons en deux ensembles. Le premier est composé d'un petit effectif d'individus, le plus représentatif possible de la diversité génomique à étudier, ces individus seront séquencés avec des lectures longues à forte couverture pour découvrir de nouveaux variants de structure et constituer un catalogue de qualité. Dans le second ensemble, de taille plus grande, les individus sont séquencés avec une technologie moins chère, les lectures courtes. Ces lectures permettent de quantifier dans des grandes populations le sous-ensemble de variants de structure découverts dans le premier ensemble. Ainsi, de grandes quantités de lectures courtes vont continuer à être générées. Dans un tel plan expérimental, *a priori* les lectures courtes ne sont pas utilisées pour découvrir de nouveaux variants de structure, seulement pour le génotypage. Il est donc important d'améliorer au moins les méthodes de génotypage avec ces données, qui comme nous l'avons montré dans le Chapitre 4 sont encore peu efficaces. Il est également probable que pour certaines applications la découverte de variants de structure avec ces lectures s'avère utile. Par exemple, une étape de découverte avec toutes les lectures courtes en amont du séquençage en lectures longues peut être envisagée pour optimiser la sélection les individus du premier ensemble (c'est le cas dans l'étude effectuée sur 900 échantillons de tomates de Alonge et al. (2020), dont 100 échantillons ont été sélectionnés grâce à l'outil SVcollector (Sedlazeck et al., 2018a) pour maximiser la diversité structurale).

Une autre alternative pour ce type de projet est d'utiliser des données de séquençage "linked-reads", qui permettent d'associer une information longue distance à des données de type lectures courtes. Les premières technologies à produire ce type de données restaient encore trop chères (par exemple 10X chromium genomics) pour des projets à grands nombres d'échantillons. Mais récemment, d'autres technologies ont été développées, telles que TELL-seq (Chen et al., 2020), stLFR (Wang et al., 2019) et Haplotagging (Meier et al., 2021). En particulier, Haplotagging est un nouveau protocole libre qui permet de séquencer avec un sur-coût très faible par rapport à un séquençage Illumina classique des centaines d'individus avec une information longue distance. Dans la publication, ils démontrent qu'avec des centaines d'individus séquencés à seulement 2 à 3 x chacun, ils peuvent reconstruire de grands haplotypes et identifier des variants de structure pour des analyses poussées de génomique des populations. C'est une alternative très intéressante dans le domaine de la génomique environnementale où l'intérêt économique est plus faible et les quantités d'ADN sont plus limitées.

Dans le domaine médical, le diagnostic de nombreuses maladies par séquençage plein génome est en développement notamment en France grâce au Plan France Médecine Génomique, mais la question du coût est cruciale pour adopter ces tests en routine. Ainsi, c'est actuellement et probablement pour encore quelques années la technologie Illumina de lectures courtes qui est préconisée. Ainsi, dans ce domaine, malgré les limitations intrinsèques des courtes lectures, il y a un vrai besoin d'améliorer les méthodes de détection avec des lectures courtes.

Même si les lectures longues ont un avantage certain par rapport aux lectures courtes (et cet avantage ne va faire que croître avec l'amélioration de leur qualité et la diminution du coût), je pense que les méthodes développées pour les lectures courtes seront encore

beaucoup utilisées et méritent d'être encore améliorées. Ainsi, je propose par la suite quelques pistes d'amélioration et développements sur ces données et en particulier les données linked-reads. Concernant les lectures longues, elles vont bien entendu prendre une part de plus en plus importante dans les études des variants de structure et donc naturellement dans mes problématiques de recherche méthodologique. Les méthodes actuelles sont encore jeunes et ont encore des limitations sur lesquelles du travail reste à faire.

## 5.2 Améliorer les méthodes de détection des variants de structure avec diverses données de séquençage

### 5.2.1 Perspectives pour l'outil MindTheGap

Comme nous l'avons vu à la fin du Chapitre 2, nous travaillons actuellement à améliorer le passage à l'échelle de MindTheGap sur des données plein génome humaines. Cela passe par la réduction le nombre de Faux Positifs du module Find et le micro-assemblage plus rapide des petits variants. Cependant, les résultats du Chapitre 3 ont montré que cette approche de détection basée sur les $k$-mers a atteint ses limites pour un certains nombre d'insertions et que le point fort de MindTheGap par rapport aux autres méthodes réside plutôt dans son module d'assemblage local. Ainsi, une perspective immédiate que j'aimerais tester est de développer un pipeline qui mettrait le module d'assemblage de MindTheGap en sortie d'outils plus performants pour la détection des sites d'insertion, tels que Manta ou GRIDSS. Outre les aspects de gestion des entrées/sorties des différents outils et de leurs formats, une difficulté de ce pipeline concerne l'ancrage de l'assemblage local de MindTheGap, c'est-à-dire le choix des $k$-mers source et cible en fonction des sorties de l'outil de détection. En effet, le succès de l'assemblage repose sur le fait que ces deux $k$-mers sont présents dans le graphe de de Bruijn représentant les données de séquençage. Le module Find de MindTheGap étant basé sur l'analyse du graphe de de Bruijn, cette condition était par construction respectée. Ce n'est plus le cas avec les autres outils de l'état de l'art qui sont basés sur le mapping de lectures et renvoient le plus souvent les informations des points de cassure uniquement sur la base du génome de référence.

Si cette stratégie s'avère efficace pour améliorer le rappel des insertions complexes humaines, l'étape suivante sera d'annoter automatiquement les insertions prédites et d'évaluer leur précision en fonction des différentes caractéristiques génomiques. Ainsi, grâce aux outils développés dans l'étude du Chapitre 3, nous pourrons enrichir les sorties de MindTheGap et améliorer sa précision. Même s'il est probable que certains types d'insertions resteront difficiles à détecter avec des lectures courtes, cette stratégie pourra être utile pour les applications en santé humaine, comme le diagnostic de maladies génétiques.

### 5.2.2 Developper des outils pour les données linked-reads

Les données de type linked-reads sont une alternative aux données de lectures longues, qui offrent des prix attractifs et des caractéristiques très utiles pour les questions autour des variants de structure. D'un point de vue méthodologique, elles offrent l'avantage de pouvoir ré-utiliser et adapter nos travaux précédents sur les lectures courtes puisqu'il s'agit du même type de données mais avec des informations longue-distances supplémentaires.

L'état de l'art des méthodes dédiés aux données linked-reads est assez restreint. Quelques outils ont été développés suite à l'essor de la technologie 10X Chromium Genomics en 2016 et 2017. Suite à l'arrêt de la commercialisation de ce type de données par 10X Chromium

Genomics pour des problèmes de propriété intellectuelle, l'engouement pour cette technologie a largement décru et ces outils ont arrêté d'être maintenus. Ce n'est que récemment que d'autres technologies ont repris le flambeau et suscitent de nouveau l'intérêt notamment pour des projets de génomique des populations de génomes non modèles. Or, les outils actuels sont peu adaptés à des génomes non modèles, qui peuvent présenter une diversité génétique et un taux d'hétérozygotie accrus, et ont probablement été sur-paramétrés pour des données humaines de type 10X. D'autre part, ils sont très gourmands en ressources de calcul et notamment en utilisation de la mémoire vive, ce qui les rend parfois inutilisables.

Ainsi un premier axe de recherche porte sur ces problèmes de performances en temps et mémoire. Nous travaillons sur la représentation de ces données en mémoire et notamment des informations spécifiques à ces données : les barcodes. En effet, de nombreuses applications demandent d'extraire ou de comparer des ensembles de barcodes en fonction des régions génomiques où sont mappées leurs lectures, ou bien de récupérer l'ensemble de lectures contenant un barcode donné. Nous développons LRez, une librairie C++ et une suite d'outils, qui permet d'indexer les lectures par barcodes et de faire des opérations simples mais rapides sur les lectures en fonction de leurs barcodes. LRez est disponible sur github et bioconda et est en cours de publication (Morisse et al., 2021b). C'est une brique de base essentielle pour ensuite développer d'autres méthodes utilisant efficacement ses données.

Parmi ces méthodes, je travaille actuellement sur deux applications : l'assemblage local pour boucher les trous d'un assemblage non fini ou pour reconstruire la séquence de points de cassure, et la découverte de variants de structure. Dans le premier cas, l'information des barcodes est utilisée pour sélectionner, pour chaque séquence à assembler, un sous-ensemble de lectures susceptibles de provenir du *locus* génomique en question, en fonction des ensembles de barcodes observés aux extrémités de la séquence. La réduction du jeu de lectures permet de réduire la complexité du graphe d'assemblage. Le choix de la structure de données pour le graphe d'assemblage se pose alors : un graphe de de Bruijn en ré-utilisant directement MindTheGap, ou bien un graphe de chevauchements de lectures qui est plus classiquement utilisé pour des lectures longues puisque la taille des données le permet ici. La première solution pose des problèmes de paramétrage du graphe et notamment de la taille des $k$-mers, car les données de séquençage en entrée ont une profondeur qui dépend de l'efficacité de la sélection des barcodes. Alors que la deuxième solution permet des chevauchements de lectures de taille variable, ce qui s'adapte mieux aux variabilités de couverture. Nous étudions et comparons ces deux approches dans l'outil MTG-link [1] qui est actuellement en cours de développement.

Dans le problème de la découverte des variants de structure avec ces données, l'information longue distance portée par les barcodes fournit un signal spécifique et une valeur ajoutée par rapport aux lectures courtes seules notamment pour les grands variants de structure, et en particulier les grandes inversions. Le signal de barcodes classiquement recherché est une paire de régions distantes sur le génome qui partagent un nombre de barcodes communs plus grand qu'attendu. L'identification de ce signal soulève alors deux problèmes, l'un algorithmique sur le comptage efficace des barcodes partagés, et l'autre statistique sur la modélisation et l'évaluation statistique de ces comptages. Le premier se résout grâce aux structures d'indexation développées dans la librairie LRez (actuellement implémenté dans le prototype LEVIATHAN (Morisse et al., 2021a)), et le second est encore à explorer. Des développements sont également encore nécessaires pour l'étape suivante, qui est la caractérisation des variants de structure et de leurs points de cassure dans les régions identifiées

1. `https://github.com/anne-gcd/MTG-Link`

puisque le signal de barcodes seul donne une information fiable mais peu précise, et également pour la découverte des variants de structure qui ne possèdent pas exactement deux points de cassure, comme les insertions (un seul point de cassure) et les transpositions (trois points de cassure).

Enfin, les données issues de la technologie Haplotagging comportent une difficulté supplémentaire. Chaque individu étant séquencé à faible couverture (2-3 X), la découverte de variants de structure se fait sur l'ensemble *poolé* des individus. Les méthodes devront certainement être adaptées pour gérer ce polymorphisme dans les données qui peut gêner notamment les assemblages locaux. L'étape suivante est alors de revenir à l'information individuelle pour génotyper les variants de structure découverts pour chaque individu. Cette problématique rejoint celle adressée dans le chapitre 4 et pourra bénéficier des travaux effectués sur la représentation des variants de structure et de leurs allèles. Dans ce cas, l'utilisation des informations de barcode sera capitale car la couverture physique en molécules est bien plus importante que la couverture en lectures courtes.

### 5.2.3 Affiner les points de cassure avec des données de lectures longues

Concernant l'utilisation des lectures longues, certains types de variants restent encore mal prédits par les outils actuels ou bien leur description n'est pas suffisamment précise. Notre benchmark sur divers types d'insertions, présenté dans le Chapitre 3, montre par exemple que l'outil Sniffles renvoie des séquences insérées de mauvaise qualité avec plus de 20 % de divergence avec la séquence simulée et n'est pas capable de détecter les homologies jonctionnelles de grande taille ou les duplications en tandem (voir le matériel supplémentaire [2] de (Delage et al., 2020)). De même, les inversions bordées par des répétitions inversées (ce qui est très fréquent car cela fait partie du mécanisme de génération de ces inversions) sont encore très mal prédites avec des lectures longues (Mahmoud et al., 2019). Dans ces deux cas, le simple mapping des lectures longues sur le génome de référence ne suffit pas pour correctement prédire ces variants. Plusieurs axes d'amélioration sont possibles comme faire de l'assemblage local des lectures ou construire des séquences consensus pour les séquences insérées et les points de cassure des variants, ou encore rechercher de manière explicite la présence d'homologie jonctionnelle puisque nous avons vu que c'est une caractéristique présente dans plus de 40 % des insertions.

Le problème de détection des inversions m'intéresse particulièrement, car ce sont des variants qui jouent un rôle important dans l'adaptation, l'évolution et la spéciation des espèces, notamment dans des espèces d'insectes sur lesquelles j'entretiens des collaborations depuis plusieurs années. Dans le cas des inversions avec des répétitions inversées, la détection précise des divergences de séquence si elles existent entre les deux copies répétées et avec les lectures pourrait permettre de mieux assigner les lectures et d'identifier précisément le point de recombinaison dans la séquence répétée. Cela fait notamment écho à mes travaux de thèse de génomique comparée sur l'affinement des points de cassure de réarrangements évolutifs. J'avais développé une méthode appelée Cassis, basée sur la segmentation du signal des alignements des séquences flanquant les points de cassure (Lemaitre et al., 2008; Baudet et al., 2010). Par la suite, j'ai ré-utilisé et adapté cette méthode pour des données de séquençage dans le cadre d'un projet d'analyse de la biologie de virus intégrés dans le génome d'une guêpe parasitoïde. C'est un système où des répétitions directes du génome viral permettent à celui-ci de s'exciser du génome hôte pour former des cercles viraux indépendants qui sont ensuite exportés pour défendre les œufs de la guêpe contre le système

2. `https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-020-07125-5#Sec34`

immunitaire de l'organisme qu'elle parasite. L'utilisation de cette approche de segmentation sur le signal des mutations ponctuelles a permis l'identification précise des points d'excision et l'analyse de leur répartition pour mieux comprendre ce mécanisme moléculaire (Legeai et al., 2020). Dans le cas de lectures longues, la problématique algorithmique sera d'obtenir des alignements de séquences précis malgré les forts taux d'erreurs de séquençage de ces données et de distinguer ces dernières du polymorphisme des génomes re-séquencés.

Ces pistes d'amélioration visent à préciser la description des variants et affiner la position du ou des points de cassure. C'est une problématique cruciale lorsqu'on veut ensuite comparer des prédictions obtenues avec des outils différents ou chez des individus différents ou encore pour le génotypage des variants dans des populations.

## 5.3 Représentation et quantification des Variants de Structure dans les graphes de génome

Ces dernières années, les représentations de génomes sous forme de graphes de séquences sont de plus en plus développées et plébiscitées pour remplacer la représentation linéaire du génome de référence (voir les articles de revue de The Computational Pan-Genomics Consortium (2018) et de Sherman and Salzberg (2020)). Ainsi, au lieu de représenter classiquement le génome d'une espèce par une unique séquence linéaire et arbitraire, la génomique se tourne vers des modèles de représentation plus complexes, tels que les graphes de séquences, graphes de variations, ou pan-génomes, permettant de représenter également l'ensemble des variants connus pour une espèce ou plusieurs espèces proches (on peut notamment citer la suite très connue VG-toolkit (Garrison et al., 2018), les outils GRAF de SevenBridges (Rakocevic et al., 2019) ou plus récemment encore minigraph (Li et al., 2020)). Ces représentations, couplées à de nouveaux outils d'alignement et de détection de variants sur graphe, permettent notamment d'améliorer la sensibilité des méthodes de découverte des variants en s'affranchissant du biais dû au choix d'un individu de référence (Garrison et al., 2018; Rakocevic et al., 2019).

Au cours de mes travaux sur les variants de structure, j'ai utilisé et exploité ce type de représentation à plusieurs reprises : comme sortie de la méthode d'assemblage guidé par référence MinYS (Chapitre 2, Section 2.2.2, p. 26) et pour le génotypage de variants de structure (Chapitre 4, Section 4.4, p. 48). Dans le premier cas, cette représentation permet de représenter la co-existence, dans des données de séquençage, de souches bactériennes présentant des différences de structure de génome. Mais une problématique méthodologique importante a été de convertir une sortie d'assemblage en un graphe de génome avec le moins de redondance possible. Cette problématique n'est pas complètement résolue et une perspective immédiate est d'identifier et de caractériser les variants de structure présents dans une telle sortie d'assemblage métagénomique. Dans le deuxième cas, lorsque les variants de structure sont déjà bien caractérisés, cette représentation s'est révélée extrêmement efficace pour représenter l'ensemble des haplotypes possibles et ne pas biaiser le génotypage de chaque variant de structure par ses voisins. Après le génotypage, une problématique que je compte aborder pour poursuivre cet axe de recherche est l'haplotypage (ou aussi appelé le phasing des variants). Cela consiste à énumérer et quantifier l'abondance de chemins dans ces sous-graphes.

De manière plus générale, si les méthodes actuelles de construction de graphes de séquences sont performantes concernant les variations ponctuelles ou de petite taille, la représentation et la détection de variants de structure plus grands et plus complexes restent

des défis méthodologiques. Les délétions et insertions sont les variants de structure les plus faciles à représenter, ils forment des *bulles* avec deux chemins alternatifs dans le graphe. Les variants équilibrés, comme les inversions, les translocations ou les transpositions sont plus difficiles à implémenter, notamment car ils peuvent créer des cycles. Par exemple, la représentation GRAF de l'entreprise SevenBridges est par nature un graphe acyclique (DAG) et les segments inversés sont dupliqués dans le graphe pour éviter les cycles (Rakocevic et al., 2019). L'outil de génotypage de VG-toolkit montre également des performances nettement moins bonnes pour les inversions que pour les insertions ou délétions (Hickey et al., 2020). À cela s'ajoutent également des problématiques d'explosion combinatoire du nombre de chemins et de passage à l'échelle lorsque le nombre d'individus ou de variants augmentent dans le graphe. Ainsi, cette thématique offre de nombreux axes de recherche pour l'étude des variants de structure, avec des défis méthodologiques importants, que je voudrais explorer sur le plus long terme.

## 5.4 Applications et questions biologiques

En parallèle de ces problématiques purement méthodologiques, je voudrais continuer à m'impliquer dans des travaux d'analyses de données, en collaboration avec des biologistes. Depuis ma thèse et jusqu'à aujourd'hui, ces deux aspects ont toujours co-habité dans mes travaux de recherche. Ils se nourrissent l'un de l'autre. La collaboration avec des biologistes sur des données et une question biologique particulière permet d'identifier précisément les besoins méthodologiques et de développer des outils réellement utiles. Grâce aux nouvelles méthodes développées, certaines questions biologiques trouvent des réponses, et le plus souvent de nouvelles questions sont posées. Ces données avec l'expertise des biologistes permettent également d'évaluer les méthodes développées dans un contexte réel et de les améliorer. Ainsi, depuis mon arrivée à Rennes en 2010, j'entretiens des collaborations notamment dans le domaine de la génomique des insectes, des ravageurs de culture avec l'INRAE sur des problématiques d'adaptation, et des papillons mimétiques avec le CNRS et le MNHN sur des question de modes d'évolution et de spéciation. Dans ce cadre, j'ai notamment encadré plusieurs jeunes chercheurs et ingénieurs sur des tâches essentiellement de bio-analyse. Ce travail d'analyse de données en collaboration est peu visible dans ce manuscrit car j'ai choisi dans ce document un angle méthodologique pour présenter mes travaux mais également car jusqu'à maintenant peu de mes collaborations portaient directement sur cette thématique des variants de structure. En effet, sur ces génomes d'organismes non modèles, il y avait déjà beaucoup à faire pour l'assemblage de génomes de référence avec des données ayant un fort taux d'hétérozygotie ou la détection de variations ponctuelles.

Cela change actuellement avec l'accessibilité des données de lectures longues à ce type de projets. Ainsi, je suis impliquée actuellement dans deux projets dont la question biologique centrale porte sur les variants de structure et sur l'évolution de la structure des génomes. Chez le papillon mimétique *Heliconius numata*, plusieurs grandes inversions dans un *locus* particulier appelé *Supergene* sont étroitement liés à la diversité des patrons de coloration des ailes et influencent l'évolution et l'écologie des populations naturelles. Je travaille donc à la caractérisation fine de ces inversions et de leurs points de cassure et à leur génotypage dans des centaines d'individus dans un contexte de variants complexes, imbriqués les uns dans les autres. Dans le deuxième projet, ce sont les phénomènes d'adaptation d'espèces de papillons alpins à leur environnement, et notamment à l'altitude, que nous étudierons par la comparaison des génomes et de leur structure. Deux espèces notamment sont issues

d'hybridations entre espèces parentales différentes, rendant le choix d'un génome de référence difficile et risqué. L'approche que nous privilégierons sera celle de la représentation des génomes par un graphe de séquences. Des questions méthodologiques sur la découverte et le génotypage des variants de structure avec des lectures longues et des données linked-reads font naturellement partie de ce projet.

De manière plus générale, les données et les méthodes commencent tout juste à être efficaces pour détecter les variants de structures dans les génomes et de nombreuses questions biologiques peuvent enfin être abordées. Parmi elles, les mécanismes d'apparition et de maintien de tels variants, leurs impacts fonctionnels sur la biologie des organismes, ainsi que leurs impacts sur l'évolution des génomes et des populations, sont des problématiques biologiques qui m'intéressent. J'aimerais les aborder par des recherches méthodologiques (et les problèmes ne manquent pas comme nous l'avons vu précédemment) mais également par l'analyse des données qui sont et seront produites en toujours plus grande quantité.

# Bibliographie

1000 Genomes Project Consortium, R. M. Durbin, G. R. Abecasis, D. L. Altshuler, A. Auton, L. D. Brooks, R. M. Durbin, R. A. Gibbs, M. E. Hurles, and G. A. McVean. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319) :1061–1073, Oct 2010. doi : 10.1038/nature09534. [lien].

1000 Genomes Project Consortium, G. R. Abecasis, A. Auton, L. D. Brooks, M. A. DePristo, R. M. Durbin, R. E. Handsaker, H. M. Kang, G. T. Marth, and G. A. McVean. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422) :56–65, Nov 2012. doi : 10.1038/nature11632. [lien].

A. Abyzov, A. E. Urban, M. Snyder, and M. Gerstein. Cnvnator : An approach to discover, genotype, and characterize typical and atypical cnvs from family and population genome sequencing. *Genome Res*, 21(6) :974–984, Jun 2011. doi : 10.1101/gr.114876.110. [lien].

M. Alonge, X. Wang, M. Benoit, S. Soyk, L. Pereira, L. Zhang, H. Suresh, S. Ramakrishnan, F. Maumus, D. Ciren, et al. Major impacts of widespread structural variation on gene expression and crop improvement in tomato. *Cell*, jun 2020. doi : 10.1016/j.cell.2020.05.021.

D. Antaki, W. M. Brandler, and J. Sebat. SV2 : accurate structural variation genotyping and de novo mutation detection from whole genomes. *Bioinformatics*, 34(10) :1774–1777, May 2018. doi : 10.1093/bioinformatics/btx813.

C. Baudet, C. Lemaitre, Z. Dias, C. Gautier, E. Tannier, and M.-F. Sagot. Cassis : Detection of genomic rearrangement breakpoints. *Bioinformatics*, 26(15) :1897–1898, Jun 2010. doi : 10.1093/bioinformatics/btq301. [lien].

G. Benoit, D. Lavenier, C. Lemaitre, and G. Rizk. Bloocoo, a memory efficient read corrector. European Conference on Computational Biology (ECCB), Sept. 2014. [lien]. Poster.

G. Benoit, C. Lemaitre, D. Lavenier, E. Drezen, T. Dayris, R. Uricaru, and G. Rizk. Reference-free compression of high throughput sequencing data with a probabilistic de bruijn graph. *BMC Bioinformatics*, 16(1) :288, 2015. doi : 10.1186/s12859-015-0709-7. [lien].

E. Burioli, M. Prearo, and M. Houssin. Complete genome sequence of ostreid herpesvirus type 1 $\mu$var isolated during mortality events in the pacific oyster crassostrea gigas in france and ireland. *Virology*, 509 :239–251, sep 2017. doi : 10.1016/j.virol.2017.06.027.

D. L. Cameron, J. Schröder, J. S. Penington, H. Do, R. Molania, A. Dobrovic, T. P. Speed, and A. T. Papenfuss. Gridss : sensitive and specific genomic rearrangement detection using

positional de bruijn graph assembly. *Genome Research*, 2017. doi : 10.1101/gr.222109.117. [lien].

D. L. Cameron, L. D. Stefano, and A. T. Papenfuss. Comprehensive evaluation and characterisation of short read general-purpose structural variant calling software. *Nature Communications*, 10 :3240, jul 2019. doi : 10.1038/s41467-019-11146-4.

G. Carrier, C. Baroukh, C. Rouxel, L. Duboscq-Bidot, N. Schreiber, and G. Bougaran. Draft genomes and phenotypic characterization of tisochrysis lutea strains. toward the production of domesticated strains with high added value. *Algal Research*, 29 :1–11, jan 2018. doi : 10.1016/j.algal.2017.10.017.

M. J. Chaisson, A. D. Sanders, X. Zhao, A. Malhotra, D. Porubsky, T. Rausch, E. J. Gardner, O. Rodriguez, L. Guo, R. L. Collins, et al. Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nature Communications*, 10 :1784, Apr. 2019. doi : 10.1038/s41467-018-08148-z. [lien].

K. Chen, J. W. Wallis, M. D. McLellan, D. E. Larson, J. M. Kalicki, C. S. Pohl, S. D. McGrath, M. C. Wendl, Q. Zhang, D. P. Locke, X. Shi, R. S. Fulton, T. J. Ley, R. K. Wilson, L. Ding, and E. R. Mardis. Breakdancer : an algorithm for high-resolution mapping of genomic structural variation. *Nat Methods*, 6(9) :677–681, Sep 2009. doi : 10.1038/nmeth.1363. [lien].

S. Chen, P. Krusche, E. Dolzhenko, R. M. Sherman, R. Petrovski, F. Schlesinger, M. Kirsche, D. R. Bentley, M. C. Schatz, F. J. Sedlazeck, and M. A. Eberle. Paragraph : a graph-based structural variant genotyper for short-read sequence data. *Genome biology*, 20 :291, Dec. 2019. ISSN 1474-760X. doi : 10.1186/s13059-019-1909-7.

X. Chen, O. Schulz-Trieglaff, R. Shaw, B. Barnes, F. Schlesinger, M. Källberg, A. J. Cox, S. Kruglyak, and C. T. Saunders. Manta : rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics (Oxford, England)*, 32 : 1220–1222, Apr. 2016. ISSN 1367-4811. doi : 10.1093/bioinformatics/btv710.

Z. Chen, L. Pham, T.-C. Wu, G. Mo, Y. Xia, P. L. Chang, D. Porter, T. Phan, H. Che, H. Tran, V. Bansal, J. Shaffer, P. Belda-Ferre, G. Humphrey, R. Knight, P. Pevzner, S. Pham, Y. Wang, and M. Lei. Ultralow-input single-tube linked-read library method enables short-read second-generation sequencing systems to routinely generate highly accurate and economical long-range sequencing information. *Genome Research*, 30(6) : 898–909, jun 2020. doi : 10.1101/gr.260380.119.

C. Chiang, R. M. Layer, G. G. Faust, M. R. Lindberg, D. B. Rose, E. P. Garrison, G. T. Marth, A. R. Quinlan, and I. M. Hall. Speedseq : ultra-fast personal genome analysis and interpretation. *Nature methods*, 12 :966–968, Oct. 2015. ISSN 1548-7105. doi : 10.1038/nmeth.3505.

R. Chikhi and G. Rizk. Space-efficient and exact de bruijn graph representation based on a bloom filter. *Algorithms Mol Biol*, 8(1) :22, 2013. doi : 10.1186/1748-7188-8-22. [lien].

W. D. Coster, M. H. Weissensteiner, and F. J. Sedlazeck. Towards population-scale long-read sequencing. *Nature Reviews Genetics*, may 2021. doi : 10.1038/s41576-021-00367-3.

S. Daval, A. Belcour, K. Gazengel, L. Legrand, J. Gouzy, L. Cottret, L. Lebreton, Y. Aigu, C. Mougel, and M. J. Manzanares-Dauleux. Computational analysis of the plasmodiophora brassicae genome : mitochondrial sequence description and metabolic pathway database design. *Genomics*, 111(6) :1629–1640, dec 2019. doi : 10.1016/j.ygeno.2018.11.013.

W. J. Delage, J. Thevenon, and C. Lemaitre. Towards a better understanding of the low recall of insertion variants with short-read based variant callers. *BMC Genomics*, 21(1), nov 2020. doi : 10.1186/s12864-020-07125-5.

E. Drezen, G. Rizk, R. Chikhi, C. Deltel, C. Lemaitre, P. Peterlongo, and D. Lavenier. Gatb : Genome assembly & analysis tool box. *Bioinformatics*, 30(20) :2959–2961, Oct 2014. doi : 10.1093/bioinformatics/btu406. [lien].

H. P. Eggertsson, S. Kristmundsdottir, D. Beyter, H. Jonsson, A. Skuladottir, M. T. Hardarson, D. F. Gudbjartsson, K. Stefansson, B. V. Halldorsson, and P. Melsted. GraphTyper2 enables population-scale genotyping of structural variation using pangenome graphs. *Nature Communications*, 10(1), nov 2019. doi : 10.1038/s41467-019-13341-9.

A. C. English, W. J. Salerno, O. A. Hampton, C. Gonzaga-Jauregui, S. Ambreth, D. I. Ritter, C. R. Beck, C. F. Davis, M. Dahdouli, et al. Assessing structural variation in a personal genome-towards a human reference diploid genome. *BMC Genomics*, 16(1) :286, Apr 2015. doi : 10.1186/s12864-015-1479-3. [lien].

D. Feldman, D. J. Kowbel, A. Cohen, N. L. Glass, Y. Hadar, and O. Yarden. Identification and manipulation of neurospora crassa genes involved in sensitivity to furfural. *Biotechnology for Biofuels*, 12(1), sep 2019. doi : 10.1186/s13068-019-1550-4.

R. R. Fuentes, D. Chebotarov, J. Duitama, S. Smith, J. F. D. la Hoz, M. Mohiyuddin, R. A. Wing, K. L. McNally, T. Tatarinova, A. Grigoriev, R. Mauleon, and N. Alexandrov. Structural variants in 3000 rice genomes. *Genome Research*, 29(5) :870–880, apr 2019. doi : 10.1101/gr.241240.118.

I. Gabur, H. S. Chawla, R. J. Snowdon, and I. A. P. Parkin. Connecting genome structural variation with complex traits in crop plants. *Theoretical and Applied Genetics*, 132(3) : 733–750, Mar. 2019. doi : 10.1007/s00122-018-3233-0.

E. Garrison, J. Sirén, A. M. Novak, G. Hickey, J. M. Eizenga, E. T. Dawson, W. Jones, S. Garg, C. Markello, M. F. Lin, B. Paten, and R. Durbin. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature biotechnology*, 36 : 875–879, Oct. 2018. ISSN 1546-1696. doi : 10.1038/nbt.4227.

A. Gouin, F. Legeai, P. Nouhaud, A. Whibley, J.-C. Simon, and C. Lemaitre. Whole-genome re-sequencing of non-model organisms : lessons from unmapped reads. *Heredity*, 114(5) : 494–501, May 2015. doi : 10.1038/hdy.2014.85. [lien].

C. Guyomar, F. Legeai, E. Jousselin, C. Mougel, C. Lemaitre, and J.-C. Simon. Multiscale characterization of symbiont diversity in the pea aphid complex through metagenomic approaches. *Microbiome*, 6(1) :181, Oct 2018. ISSN 2049-2618. doi : 10.1186/s40168-018-0562-9. [lien].

C. Guyomar, W. Delage, F. Legeai, C. Mougel, J.-C. Simon, and C. Lemaitre. MinYS : mine your symbiont by targeted genome assembly in symbiotic communities. *NAR Genomics and Bioinformatics*, 2(3), jul 2020. doi : 10.1093/nargab/lqaa047.

I. Hajirasouliha, F. Hormozdiari, C. Alkan, J. M. Kidd, I. Birol, E. E. Eichler, and S. C. Sahinalp. Detection and characterization of novel sequence insertions using paired-end next-generation sequencing. *Bioinformatics*, 26(10) :1277–1283, May 2010. doi : 10.1093/bioinformatics/btq152. [lien].

D. Heller and M. Vingron. Svim : Structural variant identification using mapped long reads. *Bioinformatics (Oxford, England)*, Jan. 2019. ISSN 1367-4811. doi : 10.1093/bioinformatics/btz041.

G. Hickey, D. Heller, J. Monlong, J. A. Sibbesen, J. Sirén, J. Eizenga, E. T. Dawson, E. Garrison, A. M. Novak, and B. Paten. Genotyping structural variants in pangenome graphs using the vg toolkit. *Genome Biology*, 21(1), feb 2020. doi : 10.1186/s13059-020-1941-7.

J. M. Kidd, T. Graves, T. L. Newman, R. Fulton, H. S. Hayden, M. Malig, J. Kallicki, R. Kaul, R. K. Wilson, and E. E. Eichler. A human genome structural variation sequencing resource reveals insights into mutational mechanisms. *Cell*, 143(5) :837–847, nov 2010. doi : 10.1016/j.cell.2010.10.027.

M. Kirkpatrick. How and why chromosome inversions evolve. *PLoS Biology*, 8(9) :e1000501, sep 2010. doi : 10.1371/journal.pbio.1000501.

M. Kirsche, G. Prabhu, R. Sherman, B. Ni, S. Aganezov, and M. C. Schatz. Jasmine : Population-scale structural variant comparison and analysis. *bioRxiv*, may 2021. doi : 10.1101/2021.05.27.445886.

J. Korbel, A. Abyzov, X. Mu, N. Carriero, P. Cayting, Z. Zhang, M. Snyder, and M. Gerstein. Pemer : a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol*, 10 (2) :R23, Feb 2009. doi : 10.1186/gb-2009-10-2-r23. [lien].

S. Kosugi, Y. Momozawa, X. Liu, C. Terao, M. Kubo, and Y. Kamatani. Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. *Genome Biology*, 20 :117, jun 2019. doi : 10.1186/s13059-019-1720-5. [lien].

E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822) :860–921, feb 2001. doi : 10.1038/35057062.

I. Lappalainen, J. Lopez, L. Skipper, T. Hefferon, J. D. Spalding, J. Garner, C. Chen, M. Maguire, M. Corbett, G. Zhou, J. Paschall, V. Ananiev, P. Flicek, and D. M. Church. dbVar and DGVa : public archives for genomic structural variation. *Nucleic Acids Research*, 41 (D1) :D936–D941, nov 2012. doi : 10.1093/nar/gks1213.

R. M. Layer, C. Chiang, A. R. Quinlan, and I. M. Hall. Lumpy : a probabilistic framework for structural variant discovery. *Genome biology*, 15 :R84, June 2014. ISSN 1474-760X. doi : 10.1186/gb-2014-15-6-r84.

L. Lecompte, P. Peterlongo, D. Lavenier, and C. Lemaitre. SVJedi : genotyping structural variations with long reads. *Bioinformatics*, 36(17) :4568–4575, may 2020. doi : 10.1093/bioinformatics/btaa527.

H. Lee and M. C. Schatz. Genomic dark matter : the reliability of short read mapping illustrated by the genome mappability score. *Bioinformatics*, 28(16) :2097–2105, jul 2012. doi : 10.1093/bioinformatics/bts330.

F. Legeai, B. F. Santos, S. Robin, A. Bretaudeau, R. B. Dikow, C. Lemaitre, V. Jouan, M. Ravallec, J.-M. Drezen, D. Tagu, F. Baudat, G. Gyapay, X. Zhou, S. Liu, B. A. Webb, S. G. Brady, and A.-N. Volkoff. Genomic architecture of endogenous ichnoviruses reveals distinct evolutionary pathways leading to virus domestication in parasitic wasps. *BMC Biology*, 18(1), jul 2020. doi : 10.1186/s12915-020-00822-3. [lien].

C. Lemaitre, E. Tannier, C. Gautier, and M.-F. Sagot. Precise detection of rearrangement breakpoints in mammalian chromosomes. *BMC Bioinformatics*, 9(1) :286, Jun 2008. doi : 10.1186/1471-2105-9-286. [lien].

H. Li. Minimap2 : pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18) : 3094–3100, may 2018. doi : 10.1093/bioinformatics/bty191.

H. Li, X. Feng, and C. Chu. The design and construction of reference pangenome graphs with minigraph. *Genome Biology*, 21(1), oct 2020. doi : 10.1186/s13059-020-02168-z.

S. Li, R. Li, H. Li, J. Lu, Y. Li, L. Bolund, M. H. Schierup, and J. Wang. Soapindel : efficient identification of indels from short paired reads. *Genome Res*, 23(1) :195–200, Jan 2013. doi : 10.1101/gr.132480.111. [lien].

G. A. Logsdon, M. R. Vollger, and E. E. Eichler. Long-read human genome sequencing and its applications. *Nature Reviews Genetics*, 21(10) :597–614, jun 2020. doi : 10.1038/s41576-020-0236-x.

M. Mahmoud, N. Gobet, D. I. Cruz-Dávalos, N. Mounier, C. Dessimoz, and F. J. Sedlazeck. Structural variant calling : the long and the short of it. *Genome Biology*, 20(1), nov 2019. doi : 10.1186/s13059-019-1828-7.

J. I. Meier, P. A. Salazar, M. Kučka, R. W. Davies, A. Dréau, I. Aldás, O. B. Power, N. J. Nadeau, J. R. Bridle, C. Rolian, N. H. Barton, W. O. McMillan, C. D. Jiggins, and Y. F. Chan. Haplotype tagging reveals parallel formation of hybrid races in two butterfly species. *Proceedings of the National Academy of Sciences*, 118(25) :e2015005118, jun 2021. doi : 10.1073/pnas.2015005118.

M. Mohiyuddin, J. C. Mu, J. Li, N. B. Asadi, M. B. Gerstein, A. Abyzov, W. H. Wong, and H. Y. Lam. MetaSV : an accurate and integrative structural-variant caller for next generation sequencing. *Bioinformatics*, 31(16) :2741–2744, apr 2015. doi : 10.1093/bioinformatics/btv204.

P. Morisse, F. Legeai, and C. Lemaitre. LEVIATHAN : efficient discovery of large structural variants by leveraging long-range information from linked-reads data. *bioRxiv*, mar 2021a. doi : 10.1101/2021.03.25.437002.

P. Morisse, C. Lemaitre, and F. Legeai. Lrez : C++ api and toolkit for analyzing and managing linked-reads data. *arXiv*, Mar. 2021b. [lien].

S. Nurk, S. Koren, A. Rhie, M. Rautiainen, A. V. Bzikadze, A. Mikheenko, M. R. Vollger, N. Altemose, L. Uralsky, A. Gershman, et al. The complete sequence of a human genome. *bioRxiv*, may 2021. doi : 10.1101/2021.05.26.445798.

D. Ottaviani, M. LeCain, and D. Sheer. The role of microhomology in genomic structural variation. *Trends in Genetics*, 30(3) :85–94, mar 2014. doi : 10.1016/j.tig.2014.01.001.

A. W. Pang, J. R. MacDonald, D. Pinto, J. Wei, M. A. Rafiq, D. F. Conrad, H. Park, M. E. Hurles, C. Lee, J. C. Venter, E. F. Kirkness, S. Levy, L. Feuk, and S. W. Scherer. Towards a comprehensive structural variation map of an individual human genome. *Genome Biology*, 11(5) :R52, 2010. doi : 10.1186/gb-2010-11-5-r52.

G. Rakocevic, V. Semenyuk, W.-P. Lee, J. Spencer, J. Browning, I. J. Johnson, V. Arsenijevic, J. Nadj, K. Ghose, M. C. Suciu, S.-G. Ji, G. Demir, L. Li, B. Toptaş, A. Dolgoborodov, B. Pollex, I. Spulber, I. Glotova, P. Kómár, A. L. Stachyra, Y. Li, M. Popovic, M. Källberg, A. Jain, and D. Kural. Fast and accurate genomic analyses using genome graphs. *Nature genetics*, 51 :354–362, Feb. 2019. ISSN 1546-1718. doi : 10.1038/s41588-018-0316-4.

T. Rausch, T. Zichner, A. Schlattl, A. M. Stütz, V. Benes, and J. O. Korbel. Delly : structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, 28(18) : i333–i339, Sep 2012. doi : 10.1093/bioinformatics/bts378. [lien].

M. Rautiainen and T. Marschall. GraphAligner : rapid and versatile sequence-to-graph alignment. *Genome Biology*, 21(1), sep 2020. doi : 10.1186/s13059-020-02157-2.

G. Rizk, D. Lavenier, and R. Chikhi. Dsk : k-mer counting with very low memory usage. *Bioinformatics*, 29(5) :652–653, Feb 2013. doi : 10.1093/bioinformatics/btt020. [lien].

G. Rizk, A. Gouin, R. Chikhi, and C. Lemaitre. Mindthegap : integrated detection and assembly of short and long insertions. *Bioinformatics*, 30(24) :3451–3457, Dec 2014. doi : 10.1093/bioinformatics/btu545. [lien].

F. J. Sedlazeck, Z. Lemmon, S. Soyk, W. J. Salerno, Z. Lippman, and M. C. Schatz. SVCollector : Optimized sample selection for validating and long-read resequencing of structural variants. *bioRxiv*, jun 2018a. doi : 10.1101/342386.

F. J. Sedlazeck, P. Rescheneder, M. Smolka, H. Fang, M. Nattestad, A. von Haeseler, and M. C. Schatz. Accurate detection of complex structural variations using single-molecule sequencing. *Nature Methods*, 15(6) :461–468, apr 2018b. doi : 10.1038/s41592-018-0001-7.

R. M. Sherman and S. L. Salzberg. Pan-genomics in the human genome era. *Nature Reviews Genetics*, 21(4) :243–254, feb 2020. doi : 10.1038/s41576-020-0210-7.

J. A. Sibbesen, , L. Maretty, and A. Krogh. Accurate genotyping across variant classes and lengths using variant graphs. *Nature Genetics*, 50(7) :1054–1059, jun 2018. doi : 10.1038/s41588-018-0145-5.

J. T. Simpson, K. Wong, S. D. Jackman, J. E. Schein, S. J. M. Jones, and I. Birol. Abyss : a parallel assembler for short read sequence data. *Genome Res*, 19(6) :1117–1123, Jun 2009. doi : 10.1101/gr.089532.108. [lien].

N. Spies, J. M. Zook, M. Salit, and A. Sidow. svviz : a read viewer for validating structural variants. *Bioinformatics*, page btv478, aug 2015. doi : 10.1093/bioinformatics/btv478.

M. C. Stancu, M. J. van Roosmalen, I. Renkens, M. M. Nieboer, S. Middelkamp, J. de Ligt, G. Pregno, D. Giachino, G. Mandrile, J. E. Valle-Inclan, J. Korzelius, E. de Bruijn, E. Cuppen, M. E. Talkowski, T. Marschall, J. de Ridder, and W. P. Kloosterman. Mapping and

phasing of structural variation in patient genomes using nanopore sequencing. *Nature Communications*, 8(1), nov 2017. doi : 10.1038/s41467-017-01343-4.

The Computational Pan-Genomics Consortium. Computational pan-genomics : status, promises and challenges. *Briefings in Bioinformatics*, 19(1) :118—-135, Jan. 2018. doi : 10.1093/bib/bbw089.

R. Uricaru, G. Rizk, V. Lacroix, E. Quillery, O. Plantard, R. Chikhi, C. Lemaitre, and P. Peterlongo. Reference-free detection of isolated snps. *Nucleic Acids Res*, 43(2) :e11, Jan 2015. doi : 10.1093/nar/gku1187. [lien].

J. A. Wala, P. Bandopadhayay, N. Greenwald, R. ORourke, T. Sharpe, C. Stewart, S. Schumacher, Y. Li, J. Weischenfeldt, X. Yao, C. Nusbaum, P. Campbell, G. Getz, M. Meyerson, C.-Z. Zhang, M. Imielinski, and R. Beroukhim. SvABA : genome-wide detection of structural variants and indels by local assembly. *Genome Research*, mar 2018. doi : 10.1101/gr.221028.117.

O. Wang, R. Chin, X. Cheng, M. K. Y. Wu, Q. Mao, J. Tang, Y. Sun, E. Anderson, H. K. Lam, D. Chen, et al. Efficient and unique cobarcoding of second-generation sequencing reads from long DNA molecules enabling cost-effective and accurate sequencing, haplotyping, and de novo assembly. *Genome Research*, 29(5) :798–808, apr 2019. doi : 10.1101/gr.245126.118.

J. Weischenfeldt, O. Symmons, F. Spitz, and J. O. Korbel. Phenotypic impact of genomic structural variation : insights from and for human disease. *Nature Reviews Genetics*, 14 (2) :125–138, jan 2013. doi : 10.1038/nrg3373.

M. Wellenreuther, C. Mérot, E. Berdan, and L. Bernatchez. Going beyond SNPs : The role of structural genomic variants in adaptive evolution and species diversification. *Molecular Ecology*, 28(6) :1203–1209, mar 2019. doi : 10.1111/mec.15066.

S. M. Yan, R. M. Sherman, D. J. Taylor, D. R. Nair, A. N. Bortvin, M. C. Schatz, and R. C. McCoy. Local adaptation and archaic introgression shape global diversity at human structural variant loci. *bioRxiv*, jan 2021. doi : 10.1101/2021.01.26.428314. [lien].

K. Ye, M. H. Schulz, Q. Long, R. Apweiler, and Z. Ning. Pindel : a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, 25(21) :2865–2871, Nov 2009. doi : 10.1093/bioinformatics/btp394. [lien].

J. M. Zook, N. F. Hansen, N. D. Olson, L. Chapman, J. C. Mullikin, C. Xiao, S. Sherry, S. Koren, A. M. Phillippy, P. C. Boutros, et al. A robust benchmark for detection of germline large deletions and insertions. *Nature Biotechnology*, jun 2020. doi : 10.1038/s41587-020-0538-8.

# Annexe A

# Principales publications associées

## A.1 Publication 1

**MindTheGap : integrated detection and assembly of short and long insertions**

Guillaume Rizk, Anaïs Gouin, Rayan Chikhi, **Claire Lemaitre**.
*Bioinformatics* 2014 30(24) :3451-3457.

BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 24 2014, pages 3451–3457
doi:10.1093/bioinformatics/btu545

Genome analysis

Advance Access publication August 14, 2014

# MindTheGap: integrated detection and assembly of short and long insertions

Guillaume Rizk[1,*], Anaïs Gouin[2], Rayan Chikhi[3] and Claire Lemaitre[1,*]

[1]Inria/IRISA GenScale, Campus de Beaulieu, 35042 Rennes cedex, France, [2]INRA, UMR 1349 Institut de Génétique, Environnement et Protection des Plantes, Domaine de la Motte - 35653 Le Rheu Cedex, France and [3]Department of Computer Science and Engineering, Pennsylvania State University, PA, USA

Associate Editor: Michael Brudno

**ABSTRACT**

**Motivation:** Insertions play an important role in genome evolution. However, such variants are difficult to detect from short-read sequencing data, especially when they exceed the paired-end insert size. Many approaches have been proposed to call short insertion variants based on paired-end mapping. However, there remains a lack of practical methods to detect and assemble long variants.

**Results:** We propose here an original method, called MindTheGap, for the integrated detection and assembly of insertion variants from re-sequencing data. Importantly, it is designed to call insertions of any size, whether they are novel or duplicated, homozygous or heterozygous in the donor genome. MindTheGap uses an efficient *k*-mer-based method to detect insertion sites in a reference genome, and subsequently assemble them from the donor reads. MindTheGap showed high recall and precision on simulated datasets of various genome complexities. When applied to real *Caenorhabditis elegans* and human NA12878 datasets, MindTheGap detected and correctly assembled insertions >1 kb, using at most 14 GB of memory.

**Availability and implementation:** http://mindthegap.genouest.org

**Contact:** guillaume.rizk@inria.fr or claire.lemaitre@inria.fr

Received and revised on April 6, 2014; accepted on August 4, 2014

## 1 INTRODUCTION

Structural variants (SVs) are large-scale structural changes in the genome. They have been typically defined in opposition to point mutations, which are single nucleotide polymorphisms (SNPs) and short insertions or deletions (indels). SVs therefore include insertions, deletions and inversions of genomic sequences. Recent research has shown that they play an important role in evolution and diseases (1000 Genomes Project Consortium *et al.*, 2010; Stewart *et al.*, 2011). However, SVs are challenging to discover using present-day sequencing approaches, as they generally span genomic regions that are longer than the reads. Computational methods have been designed to extract evidence of SVs from sequencing data using two types of analyses: paired-end mapping of reads to a reference genome and copy number estimation using read depth (Alkan *et al.*, 2011; Medvedev *et al.*, 2009).

*To whom correspondence should be addressed.

### 1.1 Definition of insertion variants

In this work, we will focus on *insertion* variants: sequences that are present at one site (position) in the donor genome but are absent from the reference genome at this site. We divide insertions into three mutually exclusive types: (i) *novel insertions* in the donor genome that have no match in the reference, (ii) *duplicated insertions*, which are found at two or more sites in the donor and a strict subset of those in the reference and (iii) *transpositions*, which are sequences in the reference that moved to a different site in the donor. Duplicated insertions include mobile element insertions (MEI), for which databases of known sequences have been created to facilitate discovery (Stewart *et al.*, 2011).

All three types of insertions are difficult to detect using short reads. Different techniques are used to detect insertions that are *short* (shorter than the reads), *medium* (of size between read length and insert size) or *long* (of size exceeding insert size). In the next two sections, we review techniques used to identify insertion sites, and techniques used to reconstruct insertion sequences.

### 1.2 Identification of insertion sites

As short insertions are likely to be fully contained in several reads, mapping donor reads to a reference genome enables simultaneous discovery of the sites and contents of insertions (Albers *et al.*, 2011; DePristo *et al.*, 2011; Li *et al.*, 2009; Ye *et al.*, 2009). In this context, results are sensitive to mapping parameters and may be degraded in low-coverage or low-complexity regions of the reference. Although the discovery of short indels has been an extensively studied problem, a recent article has observed considerable differences between the results of popular tools (Pabinger *et al.*, 2013).

Sites of medium-sized insertions can be detected by analyzing mapping positions of paired reads. General SV calling tools call insertions sites by clustering neighboring read pairs that have a shorter insert size than expected, e.g. BreakDancer and GASV (Chen *et al.*, 2009; Sindi *et al.*, 2009). NovelSeq (Hajirasouliha *et al.*, 2010) and SOAPindel (Li *et al.*, 2013) detect sites of long, novel insertions by clustering paired reads for which one mate is unmapped.

Alternatively, tools based on read coverage can detect duplicated insertions of any length by finding reference segments that have higher read depth than expected. While insertion sites cannot be determined by this method alone, the Reprever (Kim *et al.*, 2013) software identifies low-copy duplicated

© The Author 2014. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

Downloaded from http://bioinformatics.oxfordjournals.org/ at INRIA Rennes on December 8, 2014

insertions by combining paired-end mapping with read depth analysis. Finally, several methods detect sites of mobile element insertions using collections of known transposable element sequences, by searching for read pairs where one mate is mapped to a known element and the other to a unique part of the reference genome (Ewing and Kazazian, 2011; Hormozdiari *et al.*, 2010; Stewart *et al.*, 2011).

### 1.3 Reconstruction of inserted sequences

While short insertions are easy to reconstruct (as seen in Section 1.2), to the best of our knowledge, only a few methods are capable of handling medium or long insertions. They are based on global or local *de novo* assembly of reads that are potentially involved in an insertion.

SOAPindel (Li *et al.*, 2013), Scalpel (Narzisi *et al.*, 2013) and TIGRA (Chen *et al.*, 2014) select paired reads for which one of the mates maps nearby an insertion site. The other mates are used to assemble separately each inserted sequence. This approach can only reconstruct insertions that are shorter than twice the insert size. NovelSeq (Hajirasouliha *et al.*, 2010) reconstructs novel insertions (of any size) by assembling all unmapped reads, and then aligning the extremities of assembled sequences to all predicted insertion sites. Parrish *et al.* (2011) proposed to extend this approach to duplicated insertions by performing a global assembly of all reads that are either unmapped, discordantly paired or mapped to high-coverage regions.

Cortex_var (Iqbal *et al.*, 2012) builds a colored de Bruijn graph from the reference genome and all donor reads. Insertions appear in the graph as *bubbles* (sets of paths between two nodes), where one short path corresponds to the reference genome, and longer paths correspond to inserted sequences. Theoretically, this approach enables the discovery of insertions regardless of their size and type. However, because of practical limitations, Cortex_var only finds a restricted class of bubbles: those that (i) contain exactly two paths and (ii) all intermediate nodes having exactly one in-neighbor and one out-neighbor.

To summarize, available tools are highly specialized and lack the versatility to detect and assemble insertions of any size and any type. SOAPindel, Scalpel and Cortex_var are practically limited to short insertions, Reprever is limited to low-copy duplicated sequences and Novelseq is limited to novel insertions.

### 1.4 Our contribution

We propose a new tool, MindTheGap, for detecting and assembling insertions. MindTheGap has several novel features that are not found in other tools. First, a mapping-free site detection algorithm has been designed to detect insertions of any size. Second, an improved method for insertion assembly enables the reconstruction of long insertions of all three types. Third, a memory-efficient data structure enables high scalability.

We evaluated MindTheGap on simulated and real Illumina sequencing data. Among 1 kbp simulated homozygous insertions, a large fraction were found and correctly assembled (recall values between 65–98.4%, precision >97%). Simulated heterozygous 1 kbp insertions proved to be more challenging to assemble (60% recall for *Caenorhabditis elegans*, 35% for human chromosome 22); however, precision remained high (93% and 89%, respectively). We assembled long insertions using MindTheGap on an actual whole-genome human dataset, which required only 14 GB of memory.

## 2 METHODS

The input of MindTheGap is a set of reads and a reference genome. The software performs three steps: (i) construction of the de Bruijn graph of the reads, (ii) detection of insertion breakpoints on the reference genome (*find* module) and (iii) local assembly of inserted sequences (*fill* module). Both the detection step and the assembly step rely solely on the constructed graph.

The output of the second step is a set of putative insertion positions on the reference genome, whereas the output of the last step is, for each insertion site, one or several assembled sequences.

### 2.1 de Bruijn graph construction

The de Bruijn graph is a directed graph over all distinct $k$-mers in the reads. An edge is present when two $k$-mers share an exact $(k-1)$-overlap. The graph is constructed using the algorithms implemented in the Minia assembler (Chikhi and Rizk, 2013; Salikhov *et al.*, 2013). Minia encodes the graph using a Bloom filter and an additional hash table to suppress false-positive results. The data structure supports two operations: (i) membership queries for $k$-mers that are neighbors of existing $k$-mers in the graph, and (ii) traversal of the graph from an existing $k$-mer. These operations are respectively used in Section 2.2 (insertion site detection) and Section 2.3 (local assembly).

### 2.2 Find module: detection of insertion sites

MindTheGap detects insertion sites by scanning the reference genome and testing membership of reference $k$-mers in the de Bruijn graph. Homozygous and heterozygous insertions are handled using two different methods.

*2.2.1 Homozygous insertions* The general case for detecting homozygous insertions can be modeled as follows. Let Sr be a sequence (the reference). For a position $j$ in the reference, the $k$-mer at position $(j-k+1)$ (resp. $j+1$) is called the left (resp. right) flanking $k$-mer. Let Sd (the donor) be a copy of Sr where a sequence $I$ has been inserted between the nucleotides at position $i$ and $i+1$ (the insertion site, see Fig. 1). For each position in the reference genome, a binary character records whether the $k$-mer starting at this position is present ('1') or absent ('0') in the donor reads. Depending on the context, the reference genome will correspond to the string of nucleotides or to the string of binary characters. Let a gap be a substring in the reference genome equal to $0^n$ (formed by repeating '0' $n$ times), for $n > 0$, that is immediately flanked by '1' characters. In most cases, a homozygous insertion site at position $i$ has a gap of size $k-1$ starting at position $i-k+2$ (all $k-1$ $k$-mers overlapping the insertion site are absent in $S_d$). We refer to this situation as a canonical insertion site (see Fig. 1A).

A gap may be shorter than $k-1$, for instance, when the prefix of the inserted sequence $I$ exactly matches the prefix of the sequence to the right of the insertion site (see Fig. 1B). More generally, if the longest common prefix of $I$ and $S_r[i+1\ldots]$ is of size $r_1$ and the longest common suffix of $I$ and $S_r[\ldots i]$ is of size $r_2$, then the size of the gap is $k-r_1-r_2-1$. It is important to note that when $r_1$ or $r_2$ is greater than 0, with only sequences $S_d$ and $S_r$ at hand, it is not possible to localize precisely the insertion site, as it can be at any location in $[i-r_2\ldots i+r_1]$. We refer to such sites as *fuzzy sites*. Homozygous insertion sites are called when gaps of size in the range $[k-1-r, k-1]$ are detected, with $r$ being a user-defined parameter indicating the largest allowed repeat at the insertion.

The size of the gap is an important criterion to detect homozygous insertion sites, as other types of variants also yield gaps. SNPs create gaps of size exactly $k$ and deletions of length $d$ yield gaps of size $k+d-1$.



Downloaded from http://bioinformatics.oxfordjournals.org/ at INRIA Rennes on December 8, 2014

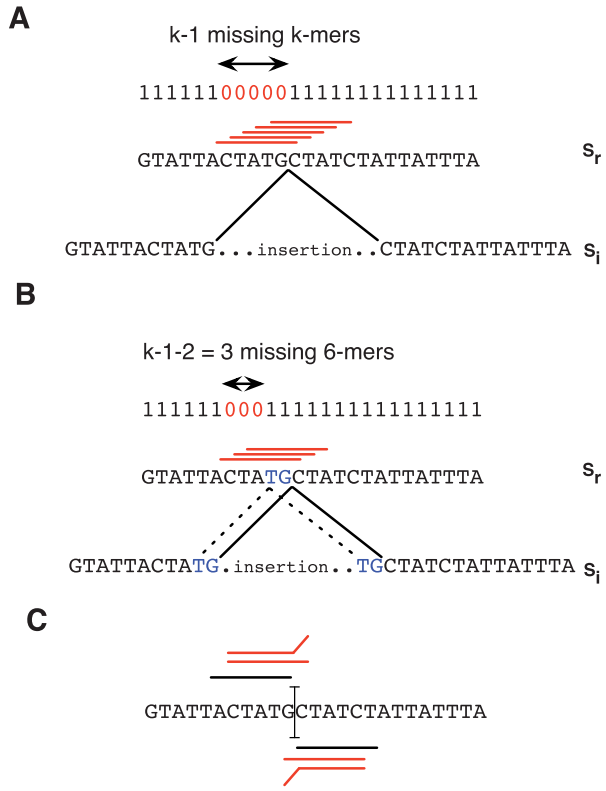


**Fig. 1.** (**A**) Canonical insertion site. The site is detected by its specific signature: the $(k-1)$ $k$-mers spanning the insertion site are missing in the sequence with the insertion (here $k = 6$), these $k$-mers are represented as red segments. (**B**) Fuzzy insertion site. Insertion ends with the same nucleotides (TG) present on the left of the site. In dashed lines, an alternative insertion site. (**C**) Heterozygous insertion site. Flanking $k$-mers (in black) surrounding a heterozygous site respectively have two right branching $k$-mers (for the $k$-mer on the left of the site) and two left-branching $k$-mers (for the right $k$-mer)

Variants that are separated by less than $k$ nucleotides yield longer gaps. In fact, only new junctions between existing sequences can yield gaps of size $<k$, which is the case for insertion events, but also for inversion or translocation sites. Finally, gaps of various sizes may also appear due to insufficient read coverage or non-uniqueness of $k$-mers inside the reference genome. These effects are controlled by the value of $k$, which is a parameter of our method.

*2.2.2 Heterozygous insertions* While heterozygous insertions sites do not yield gaps, flanking $k$-mers at these sites still exhibit features that can be detected. The left flanking $k$-mer of a heterozygous insertion site has at least two out-neighbors in the de Bruijn graph: one neighbor in the reference sequence and at least one other neighbor that is a prefix of the inserted sequence. Similarly, the right flanking $k$-mer has at least two in-neighbors with similar properties (see Fig. 1C). As in the homozygous case, small repetitions at the extremities of inserted sequences slightly alter the pattern. The left flanking $k$-mer may overlap the right flanking $k$-mer in the reference genome. MindTheGap detects heterozygous insertion sites by scanning the reference genome and testing neighborhoods of putative left and right flanking $k$-mers whose distance from one another is comprised between $k - r$ and $k$, $r$ being the same user-defined parameter as for homozygous insertions, indicating the largest allowed repeat at the insertion.

Heterozygous SNPs and deletions yield similar patterns, but the left and right flanking $k$-mers are further separated from each other ($k + 1$ nucleotides apart for SNPs and 1-bp deletions, $k+d-1$ nucleotides apart for deletions of size $d$). However, heterozygous inversions and translocations do exhibit identical patterns. Also, inexact repetitions in the reference genome create branching $k$-mers, which may yield by chance the same pattern as a heterozygous insertion. To reduce this effect, we apply an additional filter: the $k-1$ suffix (resp. prefix) of the right (resp. left) flanking $k$-mer must have less than $h$ occurrences in the reference genome. When $h$ is set to 1, this prevents the detection of patterns that may be generated by repetitions in the reference genome alone, in absence of any sequence variants.

## 2.3 Fill module: assembly of inserted sequences

The third step of MindTheGap is called the *fill* module. Starting from a known insertion site represented by flanking $k$-mers ($L$, $R$), the module performs *de novo* assembly to attempt to reconstruct the inserted sequence between $L$ and $R$. In a nutshell, a graph of contigs is constructed by performing breadth-first traversal of $k$-mers, starting from $L$. The traversal is halted when graph becomes too complex. Then, all the contigs in the graph are searched for the presence of $R$. All paths between $L$ and the contigs containing $R$ are enumerated, and one or more putative insertion sequences are returned (see Fig. 2).

More specifically, insertions are assembled using the algorithm of Minia (Chikhi and Rizk, 2013; Salikhov *et al.*, 2013). Assembly is performed by traversing the graph from a given starting $k$-mer in a breadth-first fashion. A consensus sequence (contig) is generated by skipping over certain motifs, such as bubbles (putative short variants) and tips (putative errors). This Minia assembly procedure stops whenever a contig cannot be unambiguously extended.

A graph of contigs is constructed for each insertion site ($L$, $R$) as follows. First, an initial contig $c_L$ is constructed by calling the Minia assembly procedure from the $L$ $k$-mer. Given a contig $c$ (initially $c = c_L$), the four putative neighbors of the last $k$-mer of $c$ are examined. If no neighbor is present, indicating that $c$ could not be extended, then no further action is performed for this contig. Otherwise, if two or more neighbors are present in the data structure, new contigs will be constructed starting from each of these neighbors. Directed edges will be inserted from $c$ to these new contigs. This process goes on to construct the contig graph in breadth-first order until a maximum number of contigs (parameter $n$, usually set to 100) is reached, or a maximal depth (parameter $i$, usually set to 10 kb and computed by counting nucleotides in contigs) is reached.

An exhaustive search is performed to find occurrences of $R$ within all contigs in graph, as an exact match (default behavior) or up to a constant number of mismatches. All possible paths between $L$ and $R$ are exhaustively enumerated (i.e. putative insertions). If all paths spell pair-wise identical sequences (minimum identity of 80%), then one of them is returned. Otherwise, the insertion site is considered to be unsuccessfully assembled and all paths are returned. The *fill* module is performed bi-directionally, i.e. should the ($L$, $R$) insertion site yield no path, then the module attemps to assemble the (rc($R$), rc($L$)) insertion, where rc() denotes the reverse-complement operation.

## 2.4 Evaluation protocol

*2.4.1 Simulated datasets* To evaluate MindTheGap, we generated artificial read datasets and reference genomes based on real genomes. First, we simulated sequencing reads for a real genome (the donor). Then, another genome (the reference) was obtained by simulating non-overlapping deletions from a copy of the donor genome. Deletion locations were sampled uniformly along the sequence. These deletions correspond to homozygous insertions in the donor. To simulate heterozygous insertions, reads were sampled in equal numbers from the donor

Downloaded from http://bioinformatics.oxfordjournals.org/ at INRIA Rennes on December 8, 2014

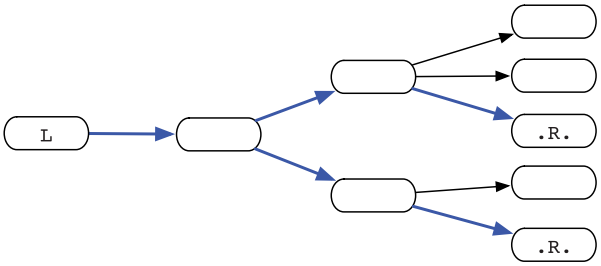


**Fig. 2.** Fill module. A graph of contig is constructed from the left flanking kmer $L$, in a breadth-first search order. Construction stops when a maximum number of nodes is reached, or when a branch becomes too deep. The right flanking kmer $R$ is searched within all nodes, finally all paths (in blue) between $L$ and $R$ are outputted as putative insertions

and the reference genomes. Sequencing was simulated with Wgsim from the Samtools package (Li *et al.*, 2009) using the following parameters: paired-end mode with $2 \times 100$ bp reads, an insert size of 300 bp (std = 50) and a base error rate of 0.01. Coverage was set to 40× for homozygous datasets and 60× for heterozygous datasets.

Three different genomes were used: *Escherichia coli K12* (4.6 Mb), *C.elegans* (100.3 Mb) and the human chromosome 22 (35 Mb without N bases). For each of them, simulated datasets were generated with homozygous or heterozygous deletions of varying sizes.

*2.4.2 Assessment of results* Positions of found breakpoints are compared with positions of introduced deletions in the genome. A breakpoint is considered as true positive (TP) if its location is at most 10 bp from a generated deletion position. This margin is meant to take into account *fuzzy sites*, for which breakpoints are not necessarily found at the exact position of the corresponding deletions (see Section 2). For each TP breakpoint, a global alignment between the assembled inserted sequence and the real sequence of the deletion is then performed with *needle* from the EMBOSS tool suite. We consider the filled sequence as TP if the alignment shows >90% of identity. Finally, the recall is the number of TP filled sequences over the number of simulated insertions, and the precision is the number of TP filled sequences over the number of filled insertions.

*2.4.3 Real sequencing data* Paired-end sequencing data from *C.elegans* strain N2 were downloaded from SRA (accession SRX026594). This dataset is composed of 33.8 M Illumina $2 \times 100$ bp read pairs (insert size of 350 bp), representing roughly 70× of coverage on the 100.3 Mb reference sequence of *C.elegans* (downloaded from NCBI version WBcel235). As we did not find any validated dataset of known large insertions for this genome, we simulated insertion variants following the protocol of simulated data: 1000 regions of a given size (here 1–100 bp or 1 kb) were deleted in the reference genome, corresponding to homozygous insertion variants in the N2 donor genome.

Sequencing data of human individual NA12878 from DePristo *et al.*, 2011, consisting of 2.8 G Illumina $2 \times 101$ bp read pairs, was downloaded from EBI (ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/technical/working/20101201_cg_NA12878/NA12878.hiseq.wgs.bwa.raw.bam). The human genome reference assembly (NCBI36 hg18) was downloaded from UCSC Genome Browser. A set of predicted or validated large insertions were obtained from Supplementary Material of Kim *et al.*, 2013. This set contained 30 validated insertions from the study of Kidd *et al.* (2010) and 44 which were predicted by Kim *et al.* (2013) based on the alignments of 40 kb sequenced fosmids from this individual to the hg18 reference genome. These are long insertions (median size of 5 kb); 68 are >1 kb and 4 are >10 kb. Among these, 61 are predicted as novel insertions, 7 as duplicated and the remaining ones have an unknown status.

## 3 RESULTS

### 3.1 Results on simulated datasets

MindTheGap was applied on several simulated datasets to precisely estimate its recall and precision. This enabled to quantify the impact of different levels of genome complexity, to independently evaluate each module and modes (detection versus assembly, homozygous versus heterozygous) and to analyze the range of insertion sizes MindTheGap is able to detect and assemble.

*3.1.1 High recall and precision in homozygous mode* For insertions of 1 kb, MindTheGap recovered between 65 and 98.4% of the simulated insertions, depending mainly on the complexity of the studied genome (Table 1). Almost all predicted homozygous insertions are true-positive results, resulting in high precision (consistently above 97%). Table 1 shows that almost all insertion sites were detected by the find module in homozygous mode. However, 19–35% of detected insertions could not be assembled by the fill module.

*3.1.2 Varying insertion lengths* Figure 3 shows that MindTheGap can detect and assemble insertions of any size. We observed that the performance of the *find* module is independent of the size of the insertions: recall of the *find* module never fell below 98.5% (data not shown), without any false positive, even for the human chromosome 22 dataset. However, lower recalls are due to the *fill* module failing to assemble longer insertions. For small insertions (<100 bp), MindTheGap obtained high recall and precision for all simulated datasets.

Only 650 over 1000 insertions of 1 kb could be assembled in the chromosome 22, and among these, 646 showed >90% identity with the original deleted sequences. This was likely because of the high repeat content of this chromosome. We observed that the insertions MindTheGap fails to assemble generally correspond to complex graph of contigs, containing many exact repeats longer than $(k - 1)$.

*3.1.3 Heterozygous mode* To evaluate the heterozygous mode of MindTheGap, we simulated datasets with only heterozygous insertions (see Section 2.4). Our analysis in Methods showed that heterozygous insertion sites were likely to be more difficult to detect and distinguish from genomic repetitions than heterozygous insertions sites. Table 2 shows that for the human and *C.elegans* simulated datasets, both recall and precision are significantly below those in homozygous mode. Further investigation showed that the low recall is owing to poor performance of the find module. We found that the results in this module were sensitive to the values of parameters $k$, $r$ (maximal repeat size at fuzzy sites) and $h$ (maximal number of occurrences of flanking kmers in the reference genome). Setting $k$ to a higher value and $r$ and $h$ to smaller values (here: $k = 51$, $r = 2$, $h = 1$) enabled to reach a precision ~97%, at the cost of a noticeably lower recall. However, using a high $k$ is detrimental to the fill module, due to read coverage being halved in heterozygous insertions. Table 2 shows that on these datasets, the fill module assembles significantly more insertions with $k = 31$.

*3.1.4 Comparison with SOAPindel* On insertions of size 1–100 bp, SOAPindel shows similar recall and precision to MindTheGap (Fig. 3). However, SOAPindel is limited in the



Downloaded from http://bioinformatics.oxfordjournals.org/ at INRIA Rennes on December 8, 2014

**Table 1.** Precision and recall results for MindTheGap in homozygous mode on simulated and real datasets

| Dataset | Recall (%) | Precision (%) | N sim. | Find module | | Fill module | |
|---|---|---|---|---|---|---|---|
| | | | | TP | FP | TP | FP |
| *E.coli* simulated dataset | 98.4 | 99.8 | 500 | 499 | 0 | 492 | 1 |
| *C.elegans* simulated dataset | 79.5 | 97.3 | 1000 | 992 | 0 | 795 | 22 |
| *C.elegans* real reads, simulated insertions | 81.1 | – | 1000 | 980 | – | 811 | – |
| Human chromosme 22 simulated dataset | 64.6 | 99.4 | 1000 | 1000 | 0 | 646 | 4 |

*Note*. Simulated insertions of size 1000 (homozygous). The number of deletions simulated in the reference genome appears in the column 'N sim.'

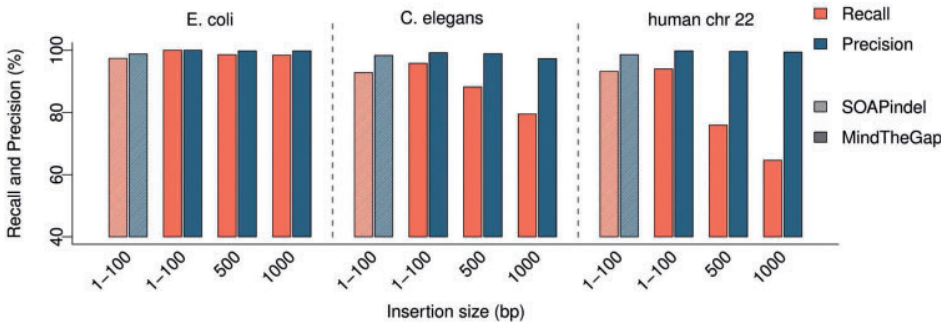


**Fig. 3.** Results of MindTheGap and SOAPindel for several insertion sizes and several genome complexities. SOAPindel results are shown only for insertions of 1–100 bp (first two shaded bars in of each genome section), as it could detect only insertions <189 bp. Best results of SOAPindel were obtained with $k$ parameter set to 31 (shown here) rather than 51. MindTheGap best results were obtained with $k$ set to 31 for *E.coli* datasets and 51 for *C.elegans* and human chromosome 22

size of detectable insertions, depending on the insert size of the reads: given our simulation parameters, we observed that SOAPindel recall decreased for insertions larger than 175 bp, and the largest insertion detected was of length 189 bp. Noticeably, the performance of SOAPindel was independent of the genotype of insertions.

### 3.2 Evaluation on a real sequencing dataset of *C.elegans*

To evaluate the impact of real reads and a real donor genome with some degree of polymorphism in the reference genome, MindTheGap was run on a *C.elegans* strain N2 read dataset against the reference genome containing simulated deletions. This is to simulate homozygous insertion variants in the donor genome. Additional insertions variants are likely to exist *C.elegans* strain. Thus, the number of FP could not be evaluated, as the true set of insertions present in these reads is unknown.

For 1 kb insertion variants, 81.1% were correctly predicted and assembled by MindTheGap (Table 1). Compared with the fully simulated dataset on the same simulated insertions, the *find* module missed more insertion sites, whereas the *fill* module had a better recall of inserted sequences. The first observation could be explained by small polymorphism near the insertion breakpoints that generated longer gaps (see Section 2), whereas the second by a higher read coverage in this dataset.

Additionally, we compared MindTheGap and SOAPindel on this dataset with 1–100 bp simulated insertions. Recall values were similar for both tools: 89% and 91%, respectively.

### 3.3 Application on real insertions of human individual NA12878

To evaluate the ability of MindTheGap to recover real insertions in real data, we executed it on a human individual NA12878 dataset containing 2.8 G 100 bp reads. As the coverage was high, parameter $k$ was empirically set to 63 and $t$ to 5 ($k$-mers with less than five occurrences were discarded). Predictions were then compared with a set of 74 large insertions predicted by alignment of fosmid sequences to the reference hg18 genome (see Section 2.4).

20 insertion sites were recovered by the *find* module. No heterozygous insertions were predicted. We set $r = 15$, which enabled to find twice more sites than with $r = 5$. This suggests that real insertions contain longer repeated sequences at their breakpoints than expected in a random simulation. By analyzing paired-end reads that mapped near each fosmid-predicted breakpoint, we could infer the genotypes: only 23 breakpoints could be confidently assigned to a homozygous genotype (i.e. with less than five read pairs spanning the breakpoint). The *find* module recovered 11 of them. Of the remaining 12 likely homozygous sites, the breakpoints of 8 of them were included in a large gap ($\geq k$) in the reference binary string. This suggests that these sites were close to other form of polymorphism, which would explain why MindTheGap did not detect them.

Among the 20 detected insertions by the *find* module, the *fill* module succeeded in reconstructing correctly two inserted sequences of sizes 4137 bp and 6729 bp, with respectively



Downloaded from http://bioinformatics.oxfordjournals.org/ at INRIA Rennes on December 8, 2014

**Table 2.** Precision and recall results for MindTheGap in heterozygous mode on simulated datasets, containing each 1000 simulated heterozygous insertions of size 1000 bp

| Dataset | Recall (%) | Precision (%) | N sim. | Find module | | Fill module $k = 51$ | | Fill module $k = 31$ | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | TP | FP | TP | FP | TP | FP |
| *C.elegans* dataset | 59.9 | 93.4 | 1000 | 807 | 11 | 310 | 80 | 599 | 42 |
| Human chromosome 22 dataset | 35.5 | 89.0 | 1000 | 816 | 28 | 226 | 8 | 355 | 44 |

*Note*. Parameter $r$ was set to 2, and assembled insertions smaller than 5 bp were filtered out.

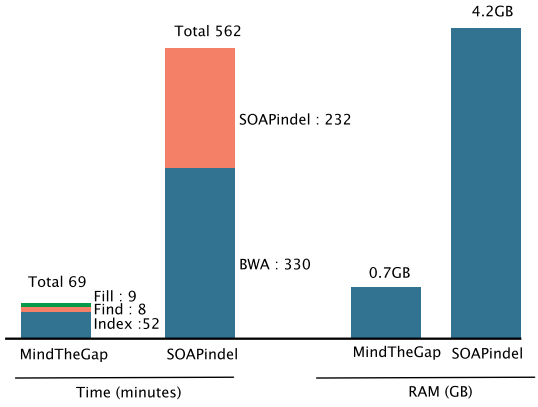


**Fig. 4.** Time (real time in minutes reported by the unix command `time`) and peak of memory used by MindTheGap (with $k = 51$) and SOAPindel (parameter $k = 51$) on the *C.elegans* real sequencing dataset (SRX026594). Peak of memory of SOAPindel approach was reached by the SOAPindel software itself (not `bwa`)

99.9 and 99.8% identity to the fosmid sequences. This corresponds to a recall of 18%, when comparing with the 11 true homozygous insertions that were detected by the *find* module. This recall value is similar to one obtained on simulated insertions of 5 kb with the same read dataset (22%, data not shown).

### 3.4 Time and memory performance

Figure 4 shows the total runtime and maximal memory used by MindTheGap and SOAPindel on the real *C.elegans* dataset. The machine used for all tests is a 12-core Intel E5-2640 @ 2.50 GHz with 192 GB of memory. For MindTheGap, the breakdown of the three steps *index*, *find* and *fill* shows that the major part of running time is spent on the *index* step. For SOAPindel, the time required to map the reads to the reference with `bwa` is included in the total time; however, SOAPindel alone remains slower than MindTheGap as a whole. SOAPindel used eight threads and MindTheGap only one.

Importantly, even though MindTheGap stores in memory the whole de Bruijn graph of the *C.elegans* read dataset, its memory peak (0.7 GB) is six times lower than SOAPindel. On the NA12878 dataset with 2.8 billion reads, MindTheGap also proved to scale efficiently: the *index*/*find*/*fill* steps respectively took 32/6/7 h, with peak memory usage of 6/14/6 GB.

## 4 DISCUSSION

MindTheGap is the first integrated method to detect and assemble insertion variants of any size and any type, using modest computing resources. The *find* module of MindTheGap differs from most other existing methods by not relying on read mapping. Instead, the de Bruijn graph of reads is compared against the reference sequence, which enables fast and low-memory analysis. However, one current limitation of the *find module* is that it fails to detect insertions when other polymorphism occurs near the insertion site. Improvements to waive this limitation are under development, based on a more detailed analysis of gaps longer than $k$. Furthermore, the method could also be used to output SNPs and other types of structural variants.

Long insertion variants are challenging to detect and assemble; thus, there is a shortage of tools to compare MindTheGap with. We compared our results with SOAPindel, which is a popular indel detection software limited to short insertions. The NovelSeq software (Hajirasouliha *et al.*, 2010) is designed to find and assemble large insertions, and therefore would have been another candidate for comparison. However, despite several attempts and reaching out to the author, we were unable to run the software successfully on any of our datasets (the `novelseq_cluster` step ran indefinitely). NovelSeq relies on a complex pipeline, and we conjecture that it may be tailored to specific data types. While most other insertion detection methods require to run external software, MindTheGap is stand-alone and is therefore easy to use. If needed, the modular organization of MindTheGap allows users to replace the *find* module with the results of a classical insertion detection based on paired-end mapping. The *fill* module could also be used as a *de novo* assembly finishing step, i.e. gap-filling between adjacent contigs in scaffolds, although we did not evaluate its performance for this task.

One important design choice for the *fill* module is to perform assembly with all the $k$-mers in the read dataset. This enables to assemble not only novel insertions, but also duplicated insertions and transposition events. Classification of assembled insertions into the different event types is not done by MindTheGap, but can be done by re-mapping insertions to the reference genome. One drawback of considering all reads during insertion assembly is that the de Bruijn graph becomes more complex to analyze. An important future work will be to improve the recall of the *fill* module by using paired-end reads information to guide traversal of contig graphs. As repeated regions are notoriously difficult to assemble, we anticipate that our approach might not be effective



Downloaded from http://bioinformatics.oxfordjournals.org/ at INRIA Rennes on December 8, 2014

for mobile element insertions. However, there exist methods tailored to the assembly of MEI, based on local assembly with recruitment of mate reads.

Our tests on the NA12878 dataset showed there is room for improvement: only two long homozygous insertions were successfully assembled out of 23 predicted ones. We postulate that (i) polymorphism or repetitions near the insertion sites hinder detection by the *find* module, and (ii) the complexity of the human genome makes *de novo* assembly of large contigs difficult. As no other tool was able to assemble long insertions, we could not assess whether our results were owing to weaknesses in our method, or to specificities of this particular dataset (complex insertion sequences or mispredicted insertions).

## ACKNOWLEDGEMENTS

The authors are grateful to Raluca Uricaru for her help and advice.

*Funding*: This work was supported by the ANR (French National Research Agency), ANR-12-BS02-0008 *Colib'read* project, ANR-12-EMMA-0019-01 *GATB* project and ANR-11-BSV7-005-01 *SPECIAPHID*.

*Conflict of interest*: none declared.

## REFERENCES

1000 Genomes Project Consortium, *et al.* (2010) A map of human genome variation from population-scale sequencing. *Nature*, **467**, 1061–1073.

Albers,C.A. *et al.* (2011) Dindel: Accurate indel calls from short-read data. *Genome Res.*, **21**, 961–973.

Alkan,C. *et al.* (2011) Genome structural variation discovery and genotyping. *Nat. Rev. Genet.*, **12**, 363–376.

Chen,K. *et al.* (2009) Breakdancer: an algorithm for high-resolution mapping of genomic structural variation. *Nat. Methods*, **6**, 677–681.

Chen,K. *et al.* (2014) Tigra: a targeted iterative graph routing assembler for breakpoint assembly. *Genome Res.*, **24**, 310–317.

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

DePristo,M.A. *et al.* (2011) A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nat. Genet.*, **43**, 491–498.

Ewing,A.D. and Kazazian,H.H. Jr. (2011) Whole-genome resequencing allows detection of many rare line-1 insertion alleles in humans. *Genome Res.*, **21**, 985–990.

Hajirasouliha,I. *et al.* (2010) Detection and characterization of novel sequence insertions using paired-end next-generation sequencing. *Bioinformatics*, **26**, 1277–1283.

Hormozdiari,F. *et al.* (2010) Next-generation variationhunter: combinatorial algorithms for transposon insertion discovery. *Bioinformatics*, **26**, i350–i357.

Iqbal,Z. *et al.* (2012) De novo assembly and genotyping of variants using colored de bruijn graphs. *Nat. Genet.*, **44**, 226–232.

Kidd,J.M. *et al.* (2010) A human genome structural variation sequencing resource reveals insights into mutational mechanisms. *Cell*, **143**, 837–847.

Kim,S. *et al.* (2013) Reprever: resolving low-copy duplicated sequences using template driven assembly. *Nucleic Acids Res.*, **41**, e128.

Li,H. *et al.* (2009) The sequence alignment/map format and samtools. *Bioinformatics*, **25**, 2078–2079.

Li,S. *et al.* (2013) Soapindel: efficient identification of indels from short paired reads. *Genome Res.*, **23**, 195–200.

Medvedev,P. *et al.* (2009) Computational methods for discovering structural variation with next-generation sequencing. *Nat. Methods*, **6** (**11 Suppl.**), S13–S20.

Narzisi,G. *et al.* (2014) Accurate de novo and transmitted indel detection in exome-capture data using microassembly. *Nat. Methods*, **11**, 1033–1036.

Pabinger,S. *et al.* (2013) A survey of tools for variant analysis of next-generation genome sequencing data. *Brief. Bioinform.*, **15**, 256–278.

Parrish,N. *et al.* (2011) Assembly of non-unique insertion content using next-generation sequencing. *BMC Bioinformatics*, **12** (**Suppl. 6**), S3.

Salikhov,K. *et al.* (2013) Using cascading bloom filters to improve the memory usage for de brujin graphs. *Algorithms Bioinformatics*, **9**, 364–376.

Sindi,S. *et al.* (2009) A geometric approach for classification and comparison of structural variants. *Bioinformatics*, **25**, i222–i230.

Stewart,C. *et al.* (2011) A comprehensive map of mobile element insertion polymorphisms in humans. *PLoS Genet.*, **7**, e1002236.

Ye,K. *et al.* (2009) Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, **25**, 2865–2871.

Downloaded from http://bioinformatics.oxfordjournals.org/ at INRIA Rennes on December 8, 2014

## A.2 Publication 2

**Towards a better understanding of the low recall of insertion variants with short-read based variant callers**

Wesley Delage, Julien Thevenon, **Claire Lemaitre**.
*BMC Genomics* 2020, 21(1) :762.

Delage *et al. BMC Genomics* (2020) 21:762
https://doi.org/10.1186/s12864-020-07125-5

BMC Genomics

RESEARCH ARTICLE Open Access

# Towards a better understanding of the low recall of insertion variants with short-read based variant callers

Wesley J. Delage[1*], Julien Thevenon[2] and Claire Lemaitre[1]

## Abstract

**Background:** Since 2009, numerous tools have been developed to detect structural variants using short read technologies. Insertions >50 bp are one of the hardest type to discover and are drastically underrepresented in gold standard variant callsets. The advent of long read technologies has completely changed the situation. In 2019, two independent cross technologies studies have published the most complete variant callsets with sequence resolved insertions in human individuals. Among the reported insertions, only 17 to 28% could be discovered with short-read based tools.

**Results:** In this work, we performed an in-depth analysis of these unprecedented insertion callsets in order to investigate the causes of such failures. We have first established a precise classification of insertion variants according to four layers of characterization: the nature and size of the inserted sequence, the genomic context of the insertion site and the breakpoint junction complexity. Because these levels are intertwined, we then used simulations to characterize the impact of each complexity factor on the recall of several structural variant callers. We showed that most reported insertions exhibited characteristics that may interfere with their discovery: 63% were tandem repeat expansions, 38% contained homology larger than 10 bp within their breakpoint junctions and 70% were located in simple repeats. Consequently, the recall of short-read based variant callers was significantly lower for such insertions (6% for tandem repeats vs 56% for mobile element insertions). Simulations showed that the most impacting factor was the insertion type rather than the genomic context, with various difficulties being handled differently among the tested structural variant callers, and they highlighted the lack of sequence resolution for most insertion calls.

**Conclusions:** Our results explain the low recall by pointing out several difficulty factors among the observed insertion features and provide avenues for improving SV caller algorithms and their combinations.

**Keywords:** Short reads, Variant calling, Structural variants; Insertions

## Background

The widespread use of short read massively parallel sequencing has allowed the fine characterization of the human genome variability on single nucleotide variants and small insertions/deletions (<50 bp) [1, 2]. Structural variants (SVs) are larger variants. They are defined as a fragment of DNA of more than 50 bp that differs between an individual and the reference genome [3]. There is a great variety of SVs, with various proposed stratifications. A common categorisation differentiates a deletion (DEL) for a loss of a fragment, an insertion (INS) for a gain of a fragment, an inversion for a reversion of a fragment (INV) and a translocation (TRANS) for moving a fragment to another position in the genome. SVs are drivers of the genome evolution along generations, and some of them can have a significant functional impacts on the organism and be responsible for rare Mendelian disorders [4].

*Correspondence: wesley.delage@irisa.fr
[1]Univ Rennes, Inria, CNRS, IRISA, F-35000 Rennes, France
Full list of author information is available at the end of the article

© The Author(s). 2020 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

The classical approach to discover SVs from Whole Genome Sequencing (WGS) with short reads relies on a first step consisting in mapping the reads to a reference genome. Then SV callers look for atypical mapping signals, such as discordant read pairs, clipped reads or abnormal read depth, to identify putative SV breakpoints along the reference genome [5, 6]. More than 70 SV callers have been developed up to date and several benchmarks have revealed great variability between results obtained by different methods, demonstrating that SV detection using short read sequencing remains challenging [7, 8]. The challenge is to resolve two issues: a technical and a methodological one. The technical issue concerns the sequencing technology: insert size, read size and sequencing coverage have been shown to impact SV discovery. The second issue concerns SV caller algorithms and their ability to decipher and translate the biological signal from the alignments. Thus, SVs located in repeated regions or containing repeats larger than the read size are difficult to detect [9].

In particular, insertions are one of the most difficult SV types to call [7, 8]. Because the inserted sequence is absent from the reference genome, or at least at the given locus of insertion, calling such variants and resolving the exact inserted sequence requires finely tuned approaches such as de novo or local assembly [10, 11]. This increased difficulty is well exemplified by the dramatic under-representation of such SV type in usual reference databases or standard variant callsets. For instance, dbVar at present references only 28% of insertions or duplications among the SVs larger than 50 bp. On the opposite, deletions represent more than 70% of the database, although both types are expected to be roughly equally abundant in human populations [12]. Moreover, only 1.5% of the reported insertions are sequence-resolved, that is with an inserted sequence fully characterized.

One explanation is that the size of the reads is small compared to the target event size and the detection is mainly based on alignments which may produce artefacts [13]. Another source of difficulty for insertion detection is the presence of repeated patterns at the precise rearrangement breakpoints. Several molecular mechanisms involved in rearrangement genesis are known to produce such repeated sequences, referred as junctional homology [14–16]. Junctional homology is defined as a DNA sequence that has two identical or nearly identical copies at the junctions of the two genomic segments involved in the rearrangement, when the sequence is short (<70 bp) this is often called a micro-homology [16]. The repair of DNA double strand breaks by diverse mechanisms, such as Non-Allelic Homologous Recombination (NAHR), Non-Homologous End Joining (NHEJ) or Microhomology-Mediated Break-Induced Replication (MMBIR), generate such homologies whose size depend precisely on the type of the involved mechanism. These homologies can have an impact on insertion calling performance, since the concerned region at the inserted site is no longer specific to the reference allele and it is no longer possible to identify the exact location of the insertion site. However, little is known at present about the prevalence of these homologies and their sizes for human insertion variants due to their poor referencing in databases.

More recently, novel long reads sequencing technologies have overcome these limitations and allowed the generation of more accurate datasets, finally referencing sequence-resolved insertion variants in the human genome [8, 17]. Thanks to several international efforts, some gold standard callsets have been produced in 2019, referencing tens of thousands of insertions in several human individuals [18, 19]. Among the reported insertions by Chaisson et al, a great majority (83%) could not be discovered by any of the tested short-read based SV callers. This result of recall below 17% is drastically different from the announced performances of insertion callers when evaluated on simulated datasets [20]. Indeed, Chaisson et al showed that 59% of insertion variants were found in a tandem repeat context, suggesting that most of the real insertion variants in human individuals are probably occurring in complex regions and involving complex sequences. So far, such complexity factors were rarely included nor analysed in method benchmarks and to do so, actual insertion variants require to be better characterized.

Numerous countries are developing genomic medicine programs, based on short-read sequencing. Although third generation sequencing offers an unprecedented technique for exploring the complexity of individual structural variants, most of the genomic sequencing facilities will still use short-read based sequencing in coming years for its reduced cost. Hence, there is a critical need to measure and control the caveats of standard procedures for detecting SVs with short-read sequencing data.

In this work, we performed an in-depth analysis of these unprecedented insertion callsets, in order to investigate the causes of short read based caller failures. We have first established a precise classification of insertion variants according to four different layers of characterization: the nature and size of the inserted sequence, the genomic context of the insertion site and the breakpoint junction complexity. Because these levels are intertwined, we then used simulations to characterize the impact of each complexity factor on the recall of several SV callers.

## Results

### In-depth analysis of an exhaustive insertion variant callset

In this work, we first aimed at precisely characterizing an exhaustive set of insertion variants present in a given

human individual. We based our study on a recently published SV callset published by Chaisson and colleagues in 2019 [18]. Using extensive sequencing datasets, combining different sequencing technologies and methodological approaches (short, linked and long reads, mapping-based and assembly-based SV calling), three human trios were thoroughly analysed to establish exhaustive and gold standard SV callsets (Supplementary Table S1). We first focused our study on the individual NA19240, son of the so-called Yoruban (YRI) Nigerian trio, whose SV callset contained 15,693 insertions greater than 50 bp.

#### *Nature and size of the inserted sequences*

Insertion variants can be classified in different sub-types according to the nature of the inserted sequence. Three insertion categories were distinguished in the original publication, namely tandem repeats, mobile element insertions and complex ones for all the other types. We proposed to refine this classification in five insertion sub-types, illustrated in Fig. 1. A classical subdivision consisted in distinguishing *novel sequence* insertions from insertions of exiting sequences, namely duplicative insertions. Several sub-types of duplicative insertions were then defined according to the location or amount of the inserted sequence copies in the reference genome. Among duplicative insertions, we proposed to stratify (i) *tandem duplications*, with at least one copy of the inserted sequence being adjacent to the insertion site, (ii) *dispersed duplications*, with copies that can be located anywhere else in the genome. Among tandem duplications, *tandem repeats* are characterized by multiple tandem repetitions of a seed motif within the inserted sequence. *Mobile element* insertions are a very specific sub-type whose sequences are known and referenced in families. They are notably characterized by very high copy numbers in the genome (typically greater than 500). Other dispersed duplication types were then required to have a copy number lower than 50, in order not to be confounded with potential mobile element insertions. We did not define segmental duplications and CNVs as additional sub-types of dispersed duplications, as they are defined in the literature by their size (above 1 Kb), the size being another independant level of characterization.

In order to classify the insertion callset, all inserted sequences were aligned against the human reference genome, a mobile element database and were scanned for tandem repeats (see Methods). We used a minimal sequence coverage threshold to annotate each insertion to an insertion sub-type according to the decision tree described in Fig. 1. Insertions that did not meet any requirement to be annotated as one of the previous sub-types were qualified as *unassigned* insertions.

We set the threshold to 80% for our analysis to ensure a compromise between specificity and quantity of annotated insertions in all sub-types. With such threshold, 88% of insertions could be assigned to a given type. Among the 13,850 annotated insertions, 8,735 (63%) were annotated as tandem repeats, 2,473 (17%) as mobile elements, 1,000

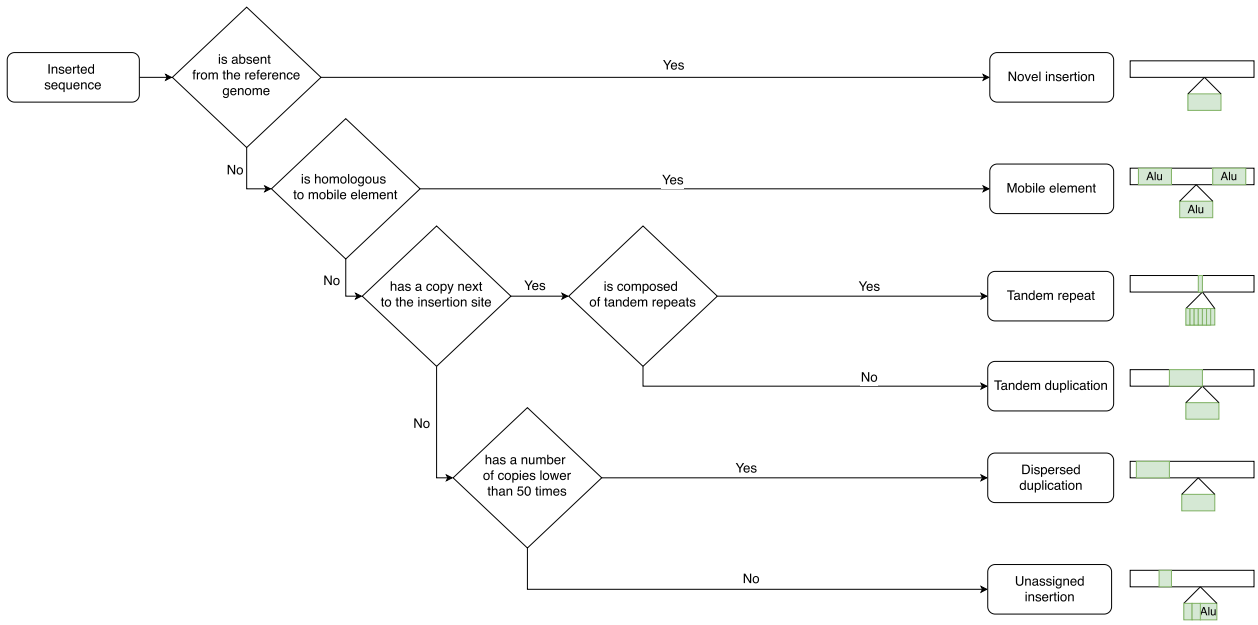


**Fig. 1** Decision tree used to classify insertion variants. Five insertion sub-types are defined according to the nature of the inserted sequence : novel sequence, tandem repeat and mobile element insertions and tandem and dispersed duplications. Unassigned insertions refer to insertions which do not meet the requirements to be assigned to at least one sub-type

(7%) as tandem duplications, 869 (6%) as novel sequences and 773 (5%) as dispersed duplications (Fig. 2b and Supplementary Table S2 for results obtained with other coverage thresholds). 46% of tandem repeats had a repeat seed smaller than 10 bp and 93% smaller than 50 bp. Compared to the classification of Chaisson et al, the proportions of tandem repeats (57% vs 56%) and mobile elements (23% vs 16%) were close. The difference in mobile element proportions mainly represented insertions that were unassigned in our annotation. The 1,843 (12%) unassigned insertions at 80% threshold showed partial annotations of mobile element (57%), tandem repeats (22%), tandem duplications (15%) or dispersed duplications (5%).

Concerning the size of the insertions, 67% of the insertions were smaller than 250 bp and only 8% had a size greater than 1 Kb (Fig. 2a). Interestingly, the size distributions differed between insertion types (Fig. 3a). Mobile elements showed the most contrasting size distribution with a strong over-representation of the 250-500 bp size class (61%). This can be explained by the most frequent and active mobile element class in the human genome being the SINE elements of size around 300 bp. Notably, the novel sequence insertion type carried a greater proportion of large insertions than other types, with 164 (19%) of the 869 novel sequences larger than 1,000 bp.

### *Characterization of insertion locations in the genome*

We then characterized the insertions based on the genomic context of their insertion site. We investigated in particular genomic features that might make read mapping and SV calling difficult, such as the

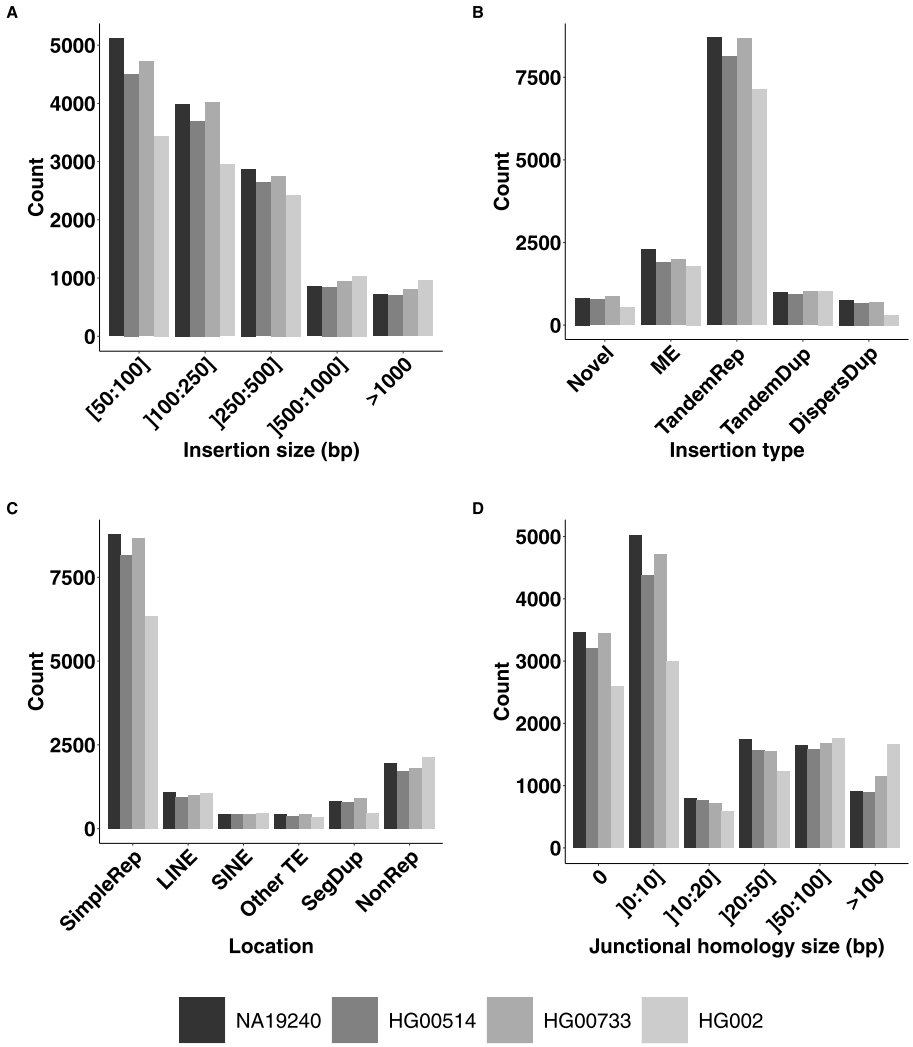


**Fig. 2** Distributions of insertion variant features across several callsets. Distributions of **a** insertion size, **b** insertion type, **c** repeated context of insertion and **d** homology size at the breakpoint for NA19240, HG00514, HG00733 and HG002 insertion variant callsets. Abbreviations: SimpleRep for simple repeat, ME for mobile element, TandemRep for tandem repeat, TandemDup for tandem duplication, DispersDup for dispersed duplication

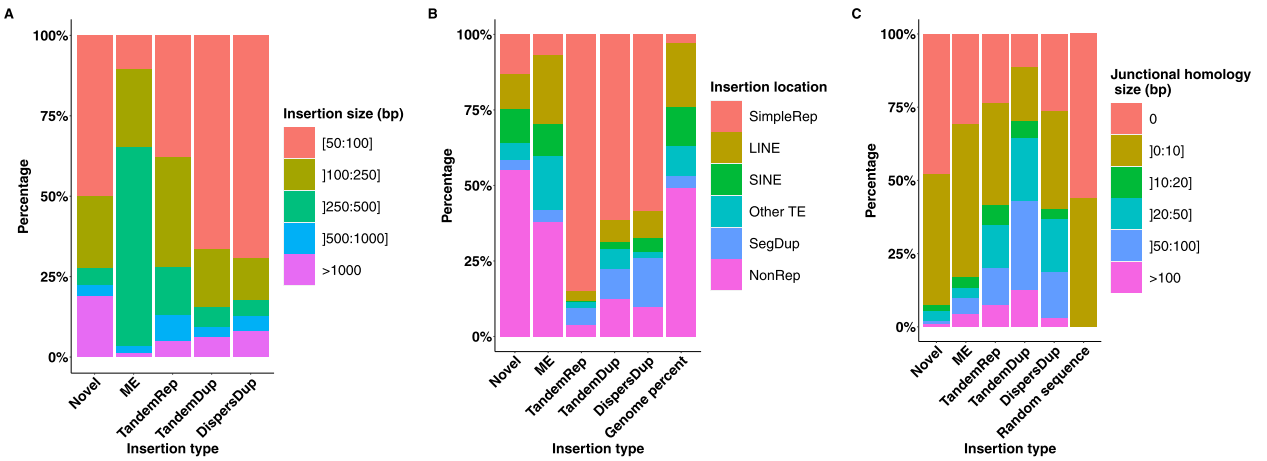


**Fig. 3** Proportions of insertion variant features according to the type of insertion. Proportions of classes of **a** insertion size, **b** insertion location, **c** homology size at the breakpoint according to the type of insertion in the NA19240 callset. Abbreviations: SimpleRep for simple repeat, ME for mobile element, TandemRep for tandem repeat, TandemDup for tandem duplication, DispersDup for dispersed duplication

repetitive content. A strong over-representation was found in regions annotated as simple repeats, with 9,675 (70%) of the annotated insertions located in these regions that only represent 1.2% of the genome (Fig. 2c). The preferred genomic context of insertions varied between insertion types (Fig. 3b). 8,047 (92%) tandem repeats, 723 (73%) tandem duplications and 519 (63%) dispersed duplications were found in simple repeat regions. Conversely, 580 (67%) novel sequence insertions and 1,383 (56%) mobile element insertions were located in other regions. We did not find a higher rate of insertions within exonic, intronic or intergenic regions compared to a uniform distribution along the genome.

### *Junctional homology*

We systematically compared the insertion site junction sequences with the inserted sequence extremities to identify stretches of identical or nearly identical sequences, here-after called junctional homologies as in [16] (see Methods). Overall 5,119 (38%) insertions showed junctional homologies larger than 10 bp (Fig. 2d). This proportion is greater than the one obtained with random sequence insertions, the largest observed junctional homology being of 7 bp among 2,000 randomly simulated insertions (see Methods). All insertion types carried junctional homologies greater than expected with random sequences. Tandem duplications and tandem repeats were the types with the greatest junctional homologies, with 428 (43%) tandem duplications and 1,751 (20%) tandem repeats that were identified with a junctional homology larger than 50 bp (Fig. 3c). This could be expected by their tandem nature. However, the homology was still smaller than their insertion size for many of them. The explanation for tandem repeat lies in their structure which is an amplification of a seed in the reference genome. Thus the largest homology size corresponded to the seed size presents at the right breakpoint (in case of left normalization). As for tandem duplications, the discordance between their annotation as tandem duplication and the smaller size of the detected junctional homology is related to the difference in the methods used to define the homology, where small distances (<10 bp) to the insertion site and to the inserted sequence extremity were required in the junctional homology case, whereas in the tandem duplication annotation case, the homologous segment had only to cover at least 80% of the inserted sequence.

### Comparison with other individual callsets

These observations were performed on the NA19240 individual callset. Hence, we asked whether they could be recurrent across individuals from various genetic backgrounds. We first considered the two other individuals of the Chaisson et al study, namely HG00514 (14,363 insertions), son of a Han Chinese (CHS) trio, and HG00733 (15,476 insertions), son of a Puerto Rican (PUR) trio. These callsets were obtained with the same sequencing technologies and SV calling methodologies as for the NA19240 individual. Then, we analyzed a callset obtained by a different study, namely the SV callset for individual HG002 ( 11,630 insertions) provided by the Genome in a Bottle (GiaB) Consortium [19]. In this study, Zook and colleagues also used multiple sequencing technologies and SV calling methods to achieve a high confidence insertion and deletion callset (see Supplementary Table S1 for a summary of the technologies and methods used for all the callsets). Before comparing insertion features between callsets, we first checked whether they contained

different variants. Using a rough estimation of shared variants, we identified only 1,169 insertion sites common to the four callsets within a 1 kb size window. On average 3,344 insertions were shared between two given callsets, and overall, more than 55% of the studied insertions were specific to a given callset. The distributions of insertion types, sizes, locations and junctional homology sizes were similar between the three individuals of the Chaisson et al study and the GiaB callset (Fig. 2).

### Short-read-based recall

In order to investigate whether the previously described insertion features impacted the recall of short-read-based (SR-based) SV callers, we reproduced our previous analysis according to the technology involved in the variant call as annotated in the callsets. For the individual NA19240, 2,363 (17%) insertion variants were comforted by SR-based SV callers. As shown in Fig. 4, this SR-based recall was highly heterogeneous with respect to the previously described insertion features. Each described feature in this work (ie. nature and size of the inserted sequence, insertion site genomic context and junctional homologies) impacted the SR-based recall. As shown in Fig. 4a, insertions larger than 500 bp were poorly discovered by SR-based methods (<3%). An increased SR-based recall for the 250-500 bp insertion size class corresponded to mobile element insertions. The greatest difference in SR-based recall was observed among the insertion types: 1,410 (56%) mobile elements and 342 (40%) novel sequence insertions could be detected with SR-based SV callers compared to only 87 (9%) tandem duplications, 484

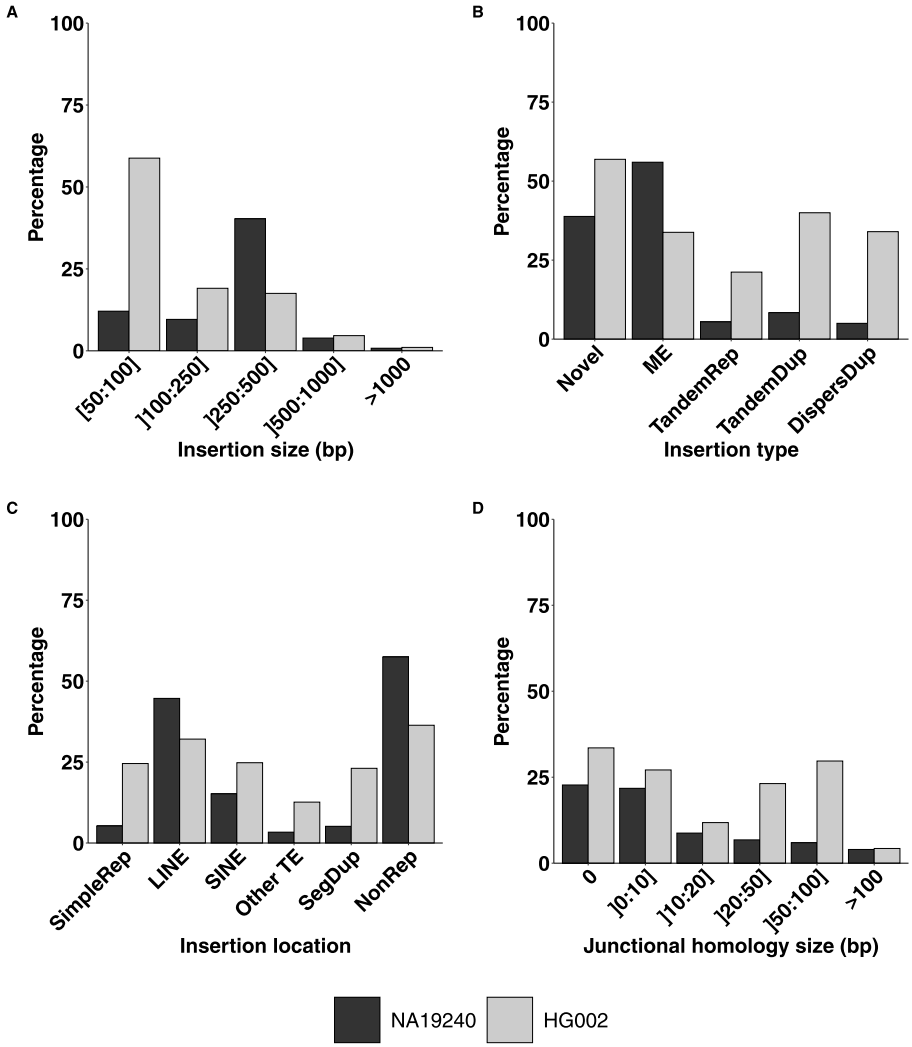


**Fig. 4** Proportions of SR-based insertion discoveries according to insertion features. Proportions of insertions that were called using short read technology data according to **a** insertion size, **b** insertion type, **c** insertion location and **d** homology size at the breakpoint, in the NA19240 and HG002 callsets. These callsets were provided by two different studies using different discovery tools and methodologies

(6%) tandem repeats and 40 (5%) dispersed duplications (Fig. 4b).

The variations of the SR-based recall with respect to insertion features were very similar between the three studied individuals from the Chaisson et al. study (Supplementary Figure S2). However, the same comparison across two different studies with different methodologies was much more contrasted. Firstly, overall around 1.6 times more insertions in proportion could be detected by SR-based methods in the GiaB study compared to the Chaisson et al study (SR-based recalls of 28% and 17% for HG002 and NA19240 callsets respectively). Secondly, the SR-based recall was more homogeneous with respect to insertion features in the GiaB callset (Fig. 4). The feature showing the most impact was the insertion size with a decrease of the SR-based recall with the insertion size, reaching below 5% for insertions larger than 500 pb for both studies (Fig. 4a). Similarly to the NA19240 callset, tandem repeats appeared more difficult to discover with SR-based methods, but to a lesser extent in the GiaB callset (Fig. 4b). Insertions located in simple repeats were less discovered using SR-based methods but this SR-based recall of 25% remained higher than for NA19240 where it only reached 5% on these locations (Fig. 4c). Junctional homology of the insertions of individual HG002 did influence its SR-based recall, but in a different manner than in the Chaisson et al study (Fig. 4d).

### Using simulations to investigate the factors impacting the insertion calling recall

In real insertion callsets, most of the previously identified factors impacting SV discovery are intertwined. In order to quantify the impact of each factor independently, we produced various simulated datasets of 2x150 bp reads at 40x coverage, containing each 200 homozygous insertion variants on the human chromosome 3. As a baseline, we simulated 250 bp novel sequences taken from *Saccharomyces cerevisiae* exonic sequences inserted inside human exons. This is meant to represent the easiest type of insertions to detect. Then, we considered four scenarios of simulations, where only one of the four previously studied factors is changed at a time with respect to the baseline simulation.

Four insertion variant callers were evaluated on these datasets. They were chosen according to their good performances in recent benchmarks [7] and to maximise the methodological diversity. GRIDSS [11], Manta [20] and SvABA [6] are based on a first mapping step to the reference genome, contrary to MindTheGap [10] which uses solely an assembly data structure (the De Bruijn graph). Two types of recall were computed depending on the precision and information given for each call: *insertion-site only* recall only evaluated if an insertion was called at an expected genomic position regardless of the predicted size or inserted sequence. As a more stringent evaluation, the *sequence-resolved* recall considered as true positives only those insertion calls having a correct genomic position and whose inserted sequence was very similar to the simulated one (>90% sequence identity and +/- 10% insertion size).

#### *Factors impacting insertion site recall*

Recalls of insertion sites for all four methods are presented for the different simulated datasets in Table 1. On the baseline simulation, all tools succeeded to detect 100% of simulated insertions, except for GRIDSS with 81% of recall. The size of the inserted sequence impacted the recall of the insertion sites for most tools, except MindTheGap. GRIDSS was challenged by small insertions (50 bp) whereas Manta and SvABA had more issues with large insertions. The most extreme behavior was observed for SvABA which was not able to find the insertion sites of any of the simulated novel sequences larger than 500 bp.

When simulating various insertion types, GRIDSS was the only tool whose recall was not negatively impacted. Manta could not find any type of dispersed duplications and showed a lower recall to detect tandem repeats with 25 bp size seeds. MindTheGap was unable to detect any type of tandem duplications and found only 58% of mobile element insertions. SvABA was not able to detect any tandem repeat insertion but was able to detect all dispersed and tandem duplications and mobile elements.

Concerning junctional homology, the tools showed contrasting behaviors. GRIDSS was the only tool unaffected by the presence and size of repeated sequence at the insertion junctions. On the contrary, MindTheGap was the most impacted by junctional homology, being unable to detect insertions with homology at any tested size. This feature is actually controlled by a parameter of MindTheGap, increasing the max-repeat parameter value to 15 bp (default : 5bp), MindTheGap discovered 99% of the insertion sites with 10 bp junctional homomolgies. Manta's recall decreased with the size of junctional homologies, whereas SvABA handled small (less than 20 bp) or very large (150 bp) junctional homologies but was affected by medium sizes.

Concerning the impact of the genomic context of insertions, no loss of recall was observed in non repeated locations. Alignment-based SV callers showed no change in recall in small simple repeat (<300 bp), SINE and LINE locations. Manta and SvABA recalls lost 5 to 6% of recall in simple repeat regions larger than the insert size (>300 bp). MindTheGap lost 42 and 47% of recall in large simple repeat and SINE location simulations. Simulating insertions close to each other on the genome, at less than 150 bp, reduced the recall of SvABA (-98%), MindTheGap (-33%) and Manta (-15%).

**Table 1** Insertion site recall of several short-read insertion callers according to different simulation scenarios. Cells of the table are colored according to the variation of the recall value of the given tool with respect to the recall obtained with the baseline simulation (first line, colored in blue): cells in red show a loss of recall >10%, cells in grey show no difference compared to baseline recall at +/- 10%

| | | Insertion site only recall (%) | | | |
|---|---|---|---|---|---|
| | | GRIDSS | Manta | SvABA | MindTheGap |
| **Baseline simulation**: 250 bp novel seq. in exons | | 83 | 100 | 100 | 100 |
| **Scenario 1 Insertion size** | 50 bp | 56 | 100 | 100 | 100 |
| | 500 bp | 100 | 86 | 0 | 99 |
| | 1,000 bp | 100 | 88 | 0 | 98 |
| | Dispersed duplication | 100 | 1 | 100 | 96 |
| **Scenario 2 Insertion type** | Tandem duplication | 100 | 100 | 100 | 0 |
| | Mobile element | 100 | 2 | 100 | 58 |
| | Tandem repeat (6 bp pattern) | 100 | 90 | 1 | 0 |
| | Tandem repeat (25 bp pattern) | 99 | 66 | 0 | 2 |
| | 10 bp | 100 | 100 | 96 | 0 |
| **Scenario 3 Junctional homology** | 20 bp | 100 | 100 | 85 | 0 |
| | 50 bp | 77 | 68 | 12 | 0 |
| | 100 bp | 100 | 22 | 49 | 0 |
| | 150 bp | 100 | 0 | 100 | 0 |
| **Scenario 4 Genomic location** | Non repeat | 83 | 100 | 99 | 96 |
| | Simple repeat (<300 bp) | 82 | 100 | 100 | 73 |
| | Simple repeat (>300 bp) | 87 | 94 | 95 | 58 |
| | SINE | 90 | 100 | 99 | 53 |
| | LINE | 80 | 100 | 97 | 90 |
| | Clustered insertions (<150 bp) | 85 | 85 | 2 | 77 |
| | Novel sequences at real locations | 84 | 80 | 71 | 38 |
| **Scenario 5 Real insertions** | Real insertions in exonic regions | 84 | 74 | 57 | 24 |
| | Real insertions at real locations | 39 | 35 | 44 | 6 |

Finally, when simulating the 889 insertions of NA19240 callset located on chromosome 3, with their reported inserted sequence at their real locations as described in the variant calling file (scenario 5), the recall of all tools dropped to less than 44%, reaching for many tools their lowest values among the different simulated datasets. This was particularly marked for GRIDSS whose recall was greater than 77% in all simulated scenarios, but achieved only 39% on this simulation. When relaxing one complexity factor, the type or the location, ie. simulating either novel sequences at the real locations or the real types in exonic regions, the drop of recall is much smaller for all tools, indicating that there is a synergetic effect of combining in a single insertion event these two factors, insertion type and insertion location.

### *Impact of quality filtering*

Previous results were computed using only the calls assessed with sufficient quality by each tool and annotated as PASS in the FILTER field of the VCF file. Removing this quality filtering allowed to increase the recall mainly for GRIDSS and SvABA (see Supplementary Table S3). Remarkably, GRIDSS reached a 100% recall on almost every scenario, except the scenario simulating the real insertions where still a 35% loss of recall was observed (Supplementary Table S3). These differences indicated that a substantial amount of true positive insertions were detected but reported as low quality calls.

### *Sequence-resolution of predicted insertions*

We then investigated whether the SV callers were also able to recover the full inserted sequences in the different simulation scenarios (Table 2). On the baseline simulation with 250 bp novel sequence insertions, every tools reported for almost all detected insertion sites a resolved and correct inserted sequence. However, these high sequence-resolved recalls dropped dramatically when deviating from the baseline scenario. Although the discovery of insertion sites was not much impacted by

**Table 2** Sequence-resolved recall of several short-read insertion callers according to different simulation scenarios. Cells of the table are colored according to the variation of the recall value of the given tool with respect to the recall obtained with the baseline simulation (first line, colored in blue): cells in red show a loss of recall >10%, cells in grey show no difference compared to baseline recall at +/- 10%

| | | Sequence-resolved recall (%) | | | |
|---|---|---|---|---|---|
| | | GRIDSS | Manta | SvABA | MindTheGap |
| **Baseline simulation**: 250 bp novel seq. in exons | | 81 | 100 | 96 | 100 |
| **Scenario 1 Insertion size** | 50 bp | 56 | 100 | 100 | 100 |
| | 500 bp | 0 | 0 | 0 | 99 |
| | 1,000 bp | 0 | 0 | 0 | 98 |
| **Scenario 2 Insertion type** | Dispersed duplication | 0 | 0 | 16 | 96 |
| | Tandem duplication | 0 | 0 | 0 | 0 |
| | Mobile element | 0 | 0 | 61 | 58 |
| | Tandem repeat (6 bp pattern) | 0 | 0 | 1 | 0 |
| | Tandem repeat (25 bp pattern) | 0 | 0 | 0 | 0 |
| **Scenario 3 Junctional homology** | 10 bp | 99 | 100 | 92 | 0 |
| | 20 bp | 100 | 100 | 78 | 0 |
| | 50 bp | 6 | 46 | 10 | 0 |
| | 100 bp | 0 | 11 | 0 | 0 |
| | 150 bp | 0 | 0 | 0 | 0 |
| **Scenario 4 Genomic location** | Non repeat | 80 | 99 | 98 | 96 |
| | Simple repeat (<300 bp) | 77 | 98 | 97 | 73 |
| | Simple repeat (>300 bp) | 77 | 93 | 90 | 58 |
| | SINE | 77 | 99 | 94 | 53 |
| | LINE | 76 | 97 | 95 | 89 |
| | Clustered insertions (<150 bp) | 75 | 73 | 2 | 77 |
| **Scenario 5 Real insertions** | Novel sequences at real locations | 64 | 73 | 67 | 37 |
| | Real insertions in exonic regions | 11 | 14 | 14 | 9 |
| | Real insertions at real locations | 6 | 23 | 30 | 6 |

the insertion size, all tools but MindTheGap were not able to recover any of the inserted sequences when it was larger than 500 bp (Table 2). On the contrary, MindTheGap assembled correctly nearly all simulated novel sequences, even those of 1 Kb. Concerning the other insertion types, tools were not able to provide sequence resolved calls, except for MindTheGap and SvABA for some dispersed duplications and mobile element insertions (Table 2). In the case of tandem repeats, GRIDSS which detected all insertion sites, reported inserted sequences of at most 150 bp (instead of 250), corresponding to the simulated read size. The increase of junctional homology size reduced the sequence resolution of GRIDSS and SvABA. Insertions located in repeated regions were less resolved than in the baseline simulation for every tools. Finally, the sequence resolution of real insertions simulated at their real locations decreased compared to the insertion site recall, GRIDSS suffering the greatest loss (-33%).

#### *False positive amount variations*

The tools with the largest recalls were also the tools producing the largest amounts of false positive discoveries (in the order of several hundreds for GRIDSS and SvABA, see Supplementary Table S4). More surprisingly, the amount of false positives was not constant for most tools between the different simulation scenarios. It increased when simulated insertions presented a duplicative pattern (mobile element, dispersed duplication and junctional homologies above 50 bp). For those, some SV callers predicted variants not only at the insertion site but also at the locations of homologous copies of the inserted sequences. Removing the quality filter led to a large increase of the amount of false positive discoveries for GRIDSS and SvABA (5 to 17 times more respectively).

#### *Unions and intersections of SV callers*

A classical strategy to report SVs on real data is to reconcile several SV callsets keeping only variants that are sim-

ilarly called by different SV callers. This strategy ensures a balance between true and false disovery rate. On the last simulation scenario, only 12% of the insertion sites were validated by the three tools, GRIDSS, Manta and SvABA, and 39% by at least two tools. However, the union of all three methods comprised 65% of the real insertion sites, which represented an increase of 20% of the best recall obtained by a single method (Fig. 5).

#### *Evaluation of insertion recall with long read simulated data*

For each of these short-read simulated datasets, we also simulated a corresponding PacBio long read dataset, with 40 X coverage and 16% error rate. We then applied a state-of-the-art long-read SV caller, Sniffles [17], on each of them to assess whether the previously identified difficulty factors for short read data have also an impact on the recall with long read data (see Supplementary Table S5). For most insertion scenarios, Sniffles reported accurately 100% of the insertions sites, except for the tandem duplication type and for the insertions with large junctional homologies (recall below 20%). In these cases, insertions were in fact reported but at more than 10 bp from the simulated insertion site. This is probably due to imprecise sequence resolution preventing the correct left normalization of breakpoint positions. Another difficulty factor was the close proximity of insertion locations, for which Sniffles reported one complex event instead of several close insertions. This mainly explained the low recall of 58% for the dataset with the real chromosome 3 insertions at their real locations. Concerning sequence resolution, although Sniffles calls contained systematically a full inserted sequence, the latter was imprecise and contained sequencing errors leading to sequence-resolved recalls around only 20% when requiring at least 90% of sequence identity. When relaxing the identity threshold to 80% or using the dedicated benchmark tool SVanalyzer from GiaB which relies on a less stringent validation, the sequence-resolved recall was similar to the insertion site recall for most insertion scenarios (Supplementary Tables S5 and S6). These results reveal that long read technologies enable the discovery of every types of insertion but the calls remain imprecise.

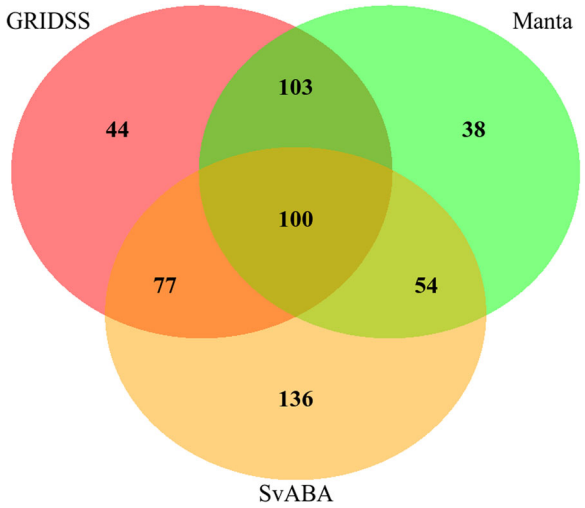


**Fig. 5** Intersections of true positive insertion callsets between different SV callers. Intersections of true positive insertion callsets between GRIDSS, SvABA and Manta on the scenario 5 simulation (real insertions at real locations). In this scenario, the 889 insertions located on the chromosome 3 from the NA19240 callset were simulated as described in the vcf file. Insertion calls were validated and compared based solely on the insertion site prediction

## Discussion

The discovery of genomic variants is an important step towards the understanding of genetic diseases and species evolution [21, 22]. The detection of insertions too small (<1kb) to be detected using comparative genomic hybridization array (CGH array) but larger than indel size (>50 bp) to be detected by the gold standard small variant discovery pipeline (GATK), remained a challenge with short read technology [4]. Thus these variations were poorly characterised in databases as compared to other SVs such as deletions. Numerous variant callers have been developed to overcome this issue but without resolving it [7]. Long read technologies or the crossing of various sequencing technologies overcome these limitations but are not affordable for many applications such as routine diagnosis of genetic diseases [18]. Thus, to improve current and future SR based SV callers, a better understanding of the actual insertion variants present in human populations is required.

We have presented here one of the most detailed and comprehensive analyses of actual insertion variants in the human genome looking for factors impacting their detection with short read re-sequencing data. This could be possible thanks to the publication of two exceptional SV callsets by Chaisson et al. [18] and Zook et al. (GiaB) [19]. These catalogs of insertions are considered as the most exhaustive for a given human individual and are qualified as gold standards thanks to their extensive validation by extensive and cross technology sequencing datasets. Unlike in the Chaisson et al study, the GiaB callset contained two categories of variants : 7640 insertions that were reported with a higher confidence (*PASS* in the *FILTER* field) and 6210 other insertions. As mentioned by the authors, the first category is likely to be biased towards easier to discover variants. Because we did not want to introduce this potential bias, and after checking that these two categories showed similar insertion feature distributions (see Supplementary Figure S1), we decided to conduct our analyses on the whole callset.

Not only, these catalogs of insertion variants are considered as the most exhaustive for a given human individual, but they are also the first sets with sequence-resolved events for any size and type of insertions. The fine resolution of the inserted sequences, present in these datasets, enabled us to propose a refined classification of insertion variants. In the two datasets, insertion types were not formally defined and the classifications differed between the datasets. Our classification allowed to normalize these heterogeneous annotations and was a direct application of variant definitions from the dbVar database which is based on the sequence ontology (SO) [12]. We based our insertion type annotation on a minimal sequence coverage threshold, that was set to a relatively high value, 80%, in order to ensure a good specificity of our annotation. Increasing this value led to many more unassigned insertions, as the annotations were based on sequence alignments that were affected by potential remaining sequencing errors in the inserted sequences, polymorphism with the reference genome and the usage of alignment heuristics. If the amount of unassigned insertions decreased with the coverage threshold value, proportions of the different insertion types remained quite stable (Supplementary Table S2). Among the 12% of unassigned insertions, some could correspond to a mixture of several insertion types, which particular case was not considered in this study.

As previously reported in the Chaisson et al and GiaB studies, we observed a highly heterogeneous distribution of insertion types and locations along the genome. The vast majority of insertions consisted in tandem repeats (63%) and most insertion sites were located in simple repeat regions (70%). These regions of low complexity, although representing a small proportion of the genome (1.2%), are therefore a major source of inter-individual variability.

The sequence-resolution provided in these SV callsets also enabled us to analyze precisely the breakpoint junctions of each insertion variant. Junctional homology has been shown to be a frequent feature of SVs, that can be used to infer the rearrangement molecular mechanism [14, 15]. Although, it has been previously described for human SV callsets (around 2,000 SV breakpoints, including less than 400 insertions) [15], this is, to our knowledge, the first exhaustive quantification of junctional homology for such a large and almost complete set of insertions in a human individual. However, our measure of homology size is highly dependent on the callset precision of the insertion site location and of the inserted sequence. As SVs are often difficult to precisely localize, are subject to left-normalization processes, and their inserted sequences were mostly obtained from error-prone long reads, our measures may likely result in an under-estimation of the actual homology sizes. Despite these potential biases, our results show that real insertion variants harbour substantially larger junctional homologies than insertions that would be drawn randomly. Our measures allowed us to compare such feature between insertion types and all insertion types have been found to have a substantial proportion of variants with large junctional homologies (greater than 20 bp). Results showed also that large insertions tended to carry larger junctional homologies. As expected by their tandem nature, tandem repeats and tandem duplications had larger homology sizes than other insertion types.

All the features of insertions characterized in our study (ie. nature and size of the inserted sequence, insertion site genomic context and junctional homologies) showed to impact the ability of SR-based SV callers to discover these variants, as defined by method annotations in the SV callsets. However, an important difference was observed between the two studies, with the GiaB study being able to detect with short reads almost twice as many insertions in proportion than in the Chaisson et al study. The difference in SR-based recalls between the two studies can certainly be explained by the difference in the read depths of sequencing datasets (77X vs 300X for Chaisson et al and GiaB studies respectively), by the different SR-based tool sets used and by the different callset filtering and merging methodologies. The two studies used roughly the same number of SV-callers (13 and 15), but with a poor intersection: only one SV-caller (Manta) was common to both studies. Additionally, the method annotation of each variant is highly dependant on the study methodology to filter and merge the numerous callsets obtained for the same individual with different sequencing technologies and SV callers. For instance, it is not clear if the presence of an SR-based tag for a given variant does necessarily mean in both studies that the latter can be sequence-resolved solely using short reads. However, both studies showed similar weaknesses to detect tandem repeats, large insertions and insertions located inside simple repeats. These observations are in-line with the already known difficulties of mapping short reads in such contexts.

These disparities between studies and the fact that most identified factors responsible of low SR-based recall are intertwined with one another in real insertion variants led us to pursue these investigations with simulated data. Our simulations did not aim at providing an exhaustive benchmark of SV callers but at identifying the precise genomic factors of insertion variants that prevent their correct discovery with short reads. As a consequence, we selected a small but diverse set of SV callers and we deliberately ran them with their default parameters. We based our selection of SV callers on a recent and comprehensive benchmark study by Kosugui et al. [7]. SV callers selected in our study were chosen for their good performance in this benchmark, for their diversity of algorithms and for

their ease of installation and usage. MindTheGap was not among the best insertion callers identified by Kosugui et al but was the only one not based on read mapping and using intensively de novo assembly with the whole read dataset.

Simulations remain a powerful approach to identify the strengths and weaknesses of SV callers but they were not meant to reflect perfectly real situations. In our simulations, several features may be far from the real complexity of human genome re-sequencing, such as some sequencing technology biases, the use of one chromosome instead of the whole genome, and the absence of other polymorphisms than insertion variants (SNPs, small indels and other SVs). As a consequence, the reported recalls are likely to be over-estimations of the ones obtained with real data. Although absolute values should be interpreted with caution, they can readily be compared between SV callers and between simulation scenarios. As a matter of fact, we often observed strong differences in recalls allowing to provide interesting insights in terms of impacting factors and SV caller behaviors. Our simulation protocol enabled to study each difficulty factor independently and highlighted the larger impact of insertion type compared to insertion location. However, all studied factors taken independently could not explain the whole loss of recall when simulating the real insertions at their real locations and there is probably an important synergetic effect of combining in a single insertion event several of the studied factors. For instance, the discovery of novel sequences in repeated regions was not a problem for almost every tested tools. However, the change of novel sequences to real inserted sequences, most of them corresponding to tandem repeats, reduced by half the recall of SV callers.

Our simulations revealed that junctional homologies as small as 10-20 bp impacted the recall of all tested tools. Such repeated sequences are likely to alter the mapping signature targeted by SV callers. Although such features of SV breakpoints and their relation to the molecular mechanisms generating SVs have long been described, they seem to be rarely taken into account in the design of SV caller algorithms. Our study of the real insertions showed that such junctional homology sizes are relatively common, with almost 40% of insertions with junctional homologies larger than 10 bp. Therefore, SV callers algorithms would benefit from taking into account such properties of the breakpoints, that are likely to generate very specific signals in terms of read mapping.

One striking result of our simulations is the absence of sequence resolution for most of the simulated insertion features and most of the tested SV callers. In addition to the obvious loss of information about the variant event, this also limits the identification of the insertion type, the genotyping and the validation of the predicted call. As a matter of fact, we observed that most insertions regardless of their type and insertion genomic context were detectable but often not reported with a sufficient quality due to this lack of resolution. Furthermore, sequence resolution is essential for the comparison and genotyping of SVs in many individuals. As these tasks are the basis for association studies and medical diagnosis, efforts should be directed towards a better resolution of the sequence of these variants [8, 23]. Results obtained with the local assembly tool MindTheGap showed that the use of the whole read dataset allowed many insertions and even large ones to be assembled. The restriction to a small subset of reads to perform local assembly may therefore be the shortcoming of the other tested SV callers. Resolving the inserted sequence is possible to some extent, but tandem repeats larger than the read size will remain difficult to resolve with short reads technology.

Interestingly, sequence resolution appeared also to be an issue with long read sequencing data. In this case, the tested long read SV caller did report full inserted sequences but with a poor sequence precision, due the higher sequencing error rate. This issue also prevented the correct left normalization of insertion sites leading to erroneous insertion locations. This low accuracy of predicted calls is likely to hamper the genotyping and comparison of SV calls between individuals. Our results therefore showed that there is also a need to improve long read SV callers as well.

Overall, the different SV callers did not performed well in every situation and in every aspects of insertion calling. Each caller showed its own strengths and weaknesses, often different from the other tools. Precisely identifying these in terms of insertion variant features and genomic contexts will enable each tool to be used to its best advantage. To do so, benchmark studies should take into account the wide variability of variant features that this present work has highlighted. Two recent SV benchmarks have raised awareness of the variability in the performances of SV callers depending on data sets and approaches [7, 9]. They looked at several factors that could be responsible for this variability. Technical factors (reads size, insert size and sequencing coverage) and biological factors (nearby SNVs or indels, genomic context, and variant size) showed to impact the recall of SV callers. However, the latter factors were analyzed for all SV types combined and none of these studies took into account the different types of insertion variants. Best practices for benchmarking small variant calling have been suggested based on gold standard callsets in high confidence regions, leaving structural variation in the fog [24]. However, it is precisely this type of variation that requires best practices for benchmarking and a standardization of annotation as they are harder to identify and report. We hope that the present fine characterization of gold standard human SV callsets will help in the development of better practices

for benchmarking SV callers, for both short and long read sequencing data.

Advises to improve the detection using short read technology have already been described such as the careful combination of complementary SV callers [7]. Meta SV callers such as Meta-sv, Parliament2 or sv-callers reconcile SV calls produced by different SV callers [25–27]. However, only the calls that are discovered concordantly between different tools are returned. This strategy allows the precision to be increased, but at the expense of the recall. Our simulations showed that the intersection of only three SV callers reduced the recall of 30%, whereas taking their union could increase the recall by at least 20%. Considering unions of callsets would require a careful control of false positive rates. A better control could probably be achieved with sequence-resolved variants and by taking into account the observed characteristics of the different insertion types. Another alternative, less described, could be the use of dedicated tools for each type of insertion, instead of using only general-purpose SV callers. Among them, Expansion Hunter has been designed to detect tandem repeats, Pamir and Popins for novel insertions and TARDIS for large duplications [28–31].

## Conclusion

In this work, we produced a detailed characterization of the insertion variants in a given human individual. We identified many factors of human insertion variants that explain their low recall with SR-based SV callers, including complex insertion types, difficult genomic contexts, large insertion sizes and junctional homologies at the breakpoints. The significant variability in the characteristics of the insertion variants, as well as the fact that all difficulties were handled differently by the different tested SV callers, call for a better characterization and comparison of SV callers according to the targeted variant features. The comparison results presented here already provide some concrete suggestions to improve insertion variant calling with short reads. First, insertion site detection could be improved by taking into account the atypical mapping signals generated by large junctional homologies. Then, sequence-resolution recall could be improved by using the whole read set instead of recruited read subsets for the assembly of the inserted sequence. Our simulation protocol also allowed us to identify complementarities between different SV callers and showed that insertion recall could be significantly improved by taking the union of calls. Finally, based on these complementarities and with improved sequence-resolution, smarter consensus selections, than simply callset unions, taking into account insertion type, size and context, could be designed to reach a high recall while controlling the False Discovery Rate. Such improvements are crucial for the generalization of population genomics and association studies to variants other than punctual ones, allowing for instance the development of personalised medicine and the resolution of diagnostic bottlenecks for many rare diseases.

## Methods

### Data origin

SV callsets from the Chaisson et al. study [18] were obtained from dbVar with the accession nstd152. The HG002 SV callset, Tier 1 version v0.6, from the GiaB study [19] was used (see the full ftp links in the Declarations section). Only insertions from the core genome, that were larger than 50 bp and sequence resolved (ie. with an inserted sequence entirely defined) and called also in at least one of the parents were kept. No filtering related to quality or coverage was applied. In the HG002 callset, insertion calls containing the "LongHomRef" tag in the FILTER field were removed because they were not confirmed by long read genotyping methods and they had thus a higher probability to be false positive discoveries (359 insertions). The human reference genome version for this study was Hg38 (GRCh38). To compare the callsets on the same reference genome, the HG002 callset produced on hs37d5 build was converted into Hg38 build using Picard, the hs37d5 to hg19 and the hg19 to hg38 chain files from GATK public chain files. Noteworthy, this process can have some impacts on a few SV calls, since some genomic regions can differ between the reference versions. In particular, the conversion (liftover) induced a loss of 60 SV calls.

### Comparison of the callsets

As a rough estimation of the amount of shared insertion variants between callsets, insertion locations were compared regardless of the insertion type or sequence. Insertion variants located less than 1,000 bp apart from one another were considered as the same variant.

### Insertion type annotation

TandemRepeatFinder (TRF) was used to annotate tandem repeats within each inserted sequence [32]. Recommended parameters were used, except for the maximum expected TR length (-l) which was set to 6 millions. In order to annotate mobile elements in inserted sequences, we used dfam, one of the annotation tools of RepeatMasker [33]. Each inserted sequence was scanned by dfam with the standard HMM profile database of human mobile elements provided by the tool. For the annotation of dispersed duplications and the occurrence count of their copies in the reference genome, each inserted sequence was locally aligned against the Hg38 genome using Blat with default parameters [34]. Only the alignments with at least 90% of sequence identity were kept. For the annotation of tandem duplications, the two sequences on either

side of the insertion site and of the same size as the insertion were aligned against the inserted sequence using Blat.

We used a minimal sequence coverage threshold, $Min_{cov}$, to annotate the insertions. To be assigned to a given sub-type, the inserted sequence had to contain at least one contiguous segment annotated with the corresponding type and covering at least $Min_{cov}$ % of the inserted sequence. Novel sequence insertions were a special case where the contiguity of the annotation was not required: more than $Min_{cov}$ % of the inserted sequence should not be covered by any alignment with the reference genome nor with the mobile element reference sequences, nor contain tandem repeats. When several types fulfilled the minimal coverage requirement, only one type was assigned according to the decision tree described in Fig. 1.

#### Junctional homology detection

Junctional homology, as referred to and defined in [16], is a DNA sequence that has two identical or nearly identical copies at the junctions of the two genomic segments involved in the rearrangement. In the case of an insertion, a junctional homology is a sequence segment at the left (resp. right) side of the insertion site which is nearly identical to the end (resp. beginning) of the inserted sequence. Small junctional homologies (<10 bp on each side) were searched in a strict manner by scanning simultaneously the 10 bp sequence at the left (resp. right) side of the insertion site and the 10 bp end (resp. beginning) of the inserted sequence, counting the number of identical nucleotides starting from the insertion site until a mismatch is encountered. For larger homologies, both the 100% identity and strict adjacency to the insertion site constraints were relaxed. We used the local alignments between the breakpoint junctions and the inserted sequence that were previously obtained with BLAT. Only the alignments with at least 90% identity and occurring at a maximum of 10 bp before (resp. after) the insertion site and at a maximum of 10 bp from the end (resp. beginning) of the inserted sequence were retained. In case of multiple candidates hits at one side of the junction, the one located at the closest position from the extremities was kept. If homologies (small or large) were found at both sides of the junction, the homology size was obtained by summing both homology sizes after removing potential overlap on the inserted sequence. To compute the expected distribution of junctional homology sizes that could be observed by chance, we generated 2,000 random insertions on the human chromosome 3 sequence. Inserted sequences were generated by concatenating 250 nucleotides sampled uniformly on the A,C,G,T alphabet. The insertion sites were sampled uniformly along the chromosome 3 sequence. Junctional homology sizes of these random insertions were identified using the same previously described methodology as for real insertions.

#### Genomic context characterization

To study the genomic context of insertions, we used the repeat content annotations of RepeatMasker from the UCSC genome browser for the Hg38 genome and the gene annotations from the Gencode v32 [35–38]. Simple repeat location were extracted from the dedicated simple repeat file from the UCSC genome browser.

#### SR-based recall of the gold standard callsets

Each callset was partitioned in two parts based on the discovery technology. The first part, referred as *Short read technology*, contained insertion calls that carried the Illumina (short reads) tag or a SR-based caller tag. For Chaisson et al callsets (NA19240, HG00514 and HG0733), the selection was performed on the vcf *INFO* field and the *UNION* variable. The *UNION* variable can take three potential values, *Pacbio*,*Bionano* or *Illumina*, that corresponded to the sequencing technology allowing the variant to be discovered. For the GiaB callset (HG002), insertions that could be discovered with short reads were identified by the *Ill* tag contained in the *ExactMatchID* located in the *INFO* field of the vcf file. Insertion calls that were labelled *Ill* only with refining methodologies and not any discovery methodologies were not taken into account for the *Short read technology* part. The second part, referred as *Other technologies*, contained all the remaining insertions. It should be noted that all insertion calls in the first part carried also at least one long read technology tag and were not discovered using only short read technology.

#### Simulations

Twenty two sequencing datasets were simulated to characterize the impact of the different insertion features on SR-based insertion variant calling. Each dataset was obtained by altering the human chromosome 3 with 200 insertions. Sequencing reads were generated using ART with the following parameters : 2x150 bp reads, at 40 X coverage, with insert size of 300 bp on average and 20 bp standard deviation [39].

##### *Baseline simulation*

The simulation referred as the baseline was meant to represent the easiest type of insertions to detect, where inserted sequences contained very few repeats and are novel in the genome, the genomic context of insertion was also simple and repeat-free, and breakpoint junctions did not have any homology. To do so, we simulated 250 bp novel sequence insertions located in exons without any homology at the breakpoint junc-

tions. Novel sequences were extracted from randomly chosen exonic regions of the *Saccharomyces cerevisae* genome.

#### *Scenario 1: varying the insertion size*

Insertion locations used in the baseline simulation were kept and the 200 inserted sequences were alternatively replaced by sequences extracted from *Saccharomyces cerevisae* exons of 3 different sizes: 50, 500, and 1000 bp.

#### *Scenario 2: varying the insertion type*

Insertion locations were identical to the baseline simulation, but the 250 bp inserted sequences were alternatively replaced by dispersed duplications, tandem repeats, tandem duplications and mobile elements. Two types of tandem repeats were simulated, with a pattern size of 6 bp or 25 bp, the pattern originating from the left breakpoint junction. As mobile elements, 200 Alu mobile element sequences with a size ranging between 200 and 300 bp were randomly extracted from the human genome based on the RepeatMasker annotation. Tandem duplications were generated by duplicating the 250 bp right breakpoint sequence. The inserted sequences of simulated dispersed duplications were extracted from exons of the chromosome 3.

#### *Scenario 3 : varying the junctional homology size*

The 250 bp insertion sequences produced in the baseline simulation were altered with junctional homology. To simulate junctional homologies, we replaced the $X$ first bases of each insertion with the same size sequence originating from the right breakpoint sequence. We simulated five junctional homology sizes ($X$ value): 10, 20, 50, 100 and 150 bp.

#### *Scenario 4 : varying the genomic context of insertion*

The 250 bp insertions from the baseline simulation were alternatively inserted in specific genomic contexts : either inside different types of mobile elements, namely SINEs and LINEs, in small (<300 bp) and large (>300 bp) simple repeats or in other regions not annotated by RepeatMasker (non repeated). A dataset with closely located variants was simulated by adding insertions closed to the insertions simulated in the baseline scenario. The distance between insertions varied uniformly from 5 to 150 bp.

#### *Scenario 5 : Real insertions*

The 889 insertions located on the chromosome 3 from the NA19240 callset were used to simulate three additional datasets. Novel sequences were first simulated at the real chromosome 3 locations, then the real insertions were simulated inside exonic regions of the chromosome 3. Finally, the 889 insertions were simulated as described in the vcf file.

### Insertion calling and benchmarking

Simulated reads were aligned with bwa against the hg38 reference genome, and read duplicates were marked with samblaster v.0.1.24 and converted into bam file with samtools v1.6 [40, 41]. Bam index and reference dictionary were obtained by picard tools v2.18.2. GRIDSS v2.8.0, Manta v1.6.0, MindTheGap v2.2.1 and SvABA v1.1.0 were all run using recommended, or otherwise default, parameters [6, 10, 11, 20]. Only "PASS" insertions, that were larger than 50 bp, were kept for the recall calculation. Two types of recalls were computed depending on the precision and information given for each call: insertion-site only recall and sequence-resolved recall. The insertion-site only recall was assessed solely based on the insertion site location prediction with a 10 bp margin around the expected location. As a more stringent evaluation, the sequence-resolved recall took also into account the inserted sequence. When it was reported, the inserted sequence had to share at least 90% of sequence identity to the simulated one and had to have a similar size of +/- 10%, to be considered as a true positive. In case of absence of alternative sequence in the vcf file but the provided annotation of the event allowed us to extract the insertion sequence from the reference genome (for instance for dispersed duplication with the duplicated copy coordinates), it was evaluated similarly as for alternative sequences. Recall was computed as the ratio between the amount of true positive discoveries and the amount of simulated insertions. We compared the absolute amounts of false positive discoveries between tools and simulations, rather the precision or FDR metrics, as the latter are dependant of the amount of true positive discoveries.

### Long read simulation and benchmark

For each short read simulated dataset, a corresponding PacBio long read simulated dataset was produced, using Simlord at 40 X coverage with probabilities of deletion, insertion and substitution equal to 11%, 4% and 1% respectively [42]. Reads were aligned with Minimap2, alignments were sorted with samtools and variants were called with Sniffles [17, 43]. The evaluation of insertion site recalls followed the same process than for short read-based variant callers. For the sequence-resolved recall, two sequence identity thresholds, 90 and 80%, were used to validate the inserted sequences. We also used the evaluation tool from GiaB, SVBenchmark module from SVanalyzer tools suite, with parameters similarly set as our benchmark method: -minsize set to 50 bp and -maxdist to 10 bp.

## Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12864-020-07125-5.

**Additional file 1:** Supplementary Figures and Tables.

### Abbreviations

SV: Structural variation; TE: Transposable element; ME: Mobile element; SR-based: Based on short reads

### Acknowledgments

We are thankful to the Genouest bioinformatics platform, computations have been made possible thanks to their computing resources. We are grateful to Justin Zook for his helpful advices to filter the GiaB callset.

### Authors' contributions

WD, JT and CL conceived the study. WD developed the annotation and simulation scripts and carried out the analysis of the results. All author(s) contributed to the writing and read and approved the final manuscript.

### Funding

Not applicable.

### Availability of data and materials

The human reference genome version for this study was Hg38 (GRCh38). SV callsets analysed in this study are publicly available (dbVar accessions: nstd152 and nstd175 for Chaisson et al and Genome in a Bottle callsets respectively), and were downloaded from the following links:
**NA19240:** http://ftp.ncbi.nlm.nih.gov/pub/dbVar/data/Homo_sapiens/by_study/genotype/nstd152/NA19240.BIP-unified.vcf.gz.
**HG00514:** http://ftp.ncbi.nlm.nih.gov/pub/dbVar/data/Homo_sapiens/by_study/genotype/nstd152/HG00514.BIP-unified.vcf.gz.
**HG00733:** http://ftp.ncbi.nlm.nih.gov/pub/dbVar/data/Homo_sapiens/by_study/genotype/nstd152/HG00733.BIP-unified.vcf.gz.
**HG002:** http://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/NIST_SVs_Integration_v0.6/HG002_SVs_Tier1_v0.6.vcf.gz.
**Reference Genome,GRCh38/hg38:** http://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz
Custom scripts used in this study are freely available at https://github.com/WesDe/DeepAn and at https://github.com/WesDe/InserSim.

### Ethics approval and consent to participate

Not applicable.

### Consent for publication

Not applicable.

### Competing interests

The authors declare that they have no competing interests.

### Author details

[1]Univ Rennes, Inria, CNRS, IRISA, F-35000 Rennes, France. [2]Inserm U1209, CNRS UMR 5309, Univ. Grenoble Alpes, Institute for Advanced Biosciences, Grenoble, France & Genetics, Genomics and Reproduction Service, Centre Hospitalo-Universitaire Grenoble-Alpes, Grenoble, France.

Received: 7 July 2020 Accepted: 6 October 2020
Published online: 04 November 2020

### References

1. Zook JM, Chapman B, Wang J, Mittelman D, Hofmann O, Hide W, Salit M. Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. Nat Biotechnol. 2014;32(3):246.
2. Mullaney JM, Mills RE, Pittard WS, Devine SE. Small insertions and deletions (indels) in human genomes. Hum Mol Genet. 2010;19(R2):131–6.
3. Baker M. Structural variation: the genome's hidden architecture. Nat Methods. 2012;9(2):133–7.
4. Feuk L, Carson AR, Scherer SW. Structural variation in the human genome. Nat Rev Genet. 2006;7(2):85–97.
5. Pirooznia M, Kramer M, Parla J, Goes FS, Potash JB, McCombie WR, Zandi PP. Validation and assessment of variant calling pipelines for next-generation sequencing. Hum Genomics. 2014;8(1):14.
6. Wala JA, Bandopadhayay P, Greenwald N, Rourke RO, Sharpe T, Stewart C, Schumacher S, Li Y, Weischenfeldt J, Yao X, Nusbaum C, Campbell P, Getz G, Meyerson M, Zhang C-Z, Imielinski M, Beroukhim R. SvABA: genome-wide detection of structural variants and indels by local assembly. Genome Res. 2018;28(4):581-91. https://doi.org/10.1101/gr.221028.117.
7. Kosugi S, Momozawa Y, Liu X, Terao C, Kubo M, Kamatani Y. Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. Genome Biol. 2019;20:117. https://doi.org/10.1186/s13059-019-1720-5.
8. Mahmoud M, Gobet N, Cruz-Dávalos DI, Mounier N, Dessimoz C, Sedlazeck FJ. Structural variant calling: the long and the short of it. Genome Biol. 2019;20:246. https://doi.org/10.1186/s13059-019-1828-7.
9. Cameron DL, Stefano LD, Papenfuss AT. Comprehensive evaluation and characterisation of short read general-purpose structural variant calling software. Nat Commun. 2019;10:324. https://doi.org/10.1038/s41467-019-11146-4.
10. Rizk G, Gouin A, Chikhi R, Lemaitre C. Mindthegap : integrated detection and assembly of short and long insertions. Bioinformatics. 2014;30(24):3451–7. https://doi.org/10.1093/bioinformatics/btu545.
11. Cameron DL, Schröder J, Penington JS, Do H, Molania R, Dobrovic A, Speed TP, Papenfuss AT. Gridss: sensitive and specific genomic rearrangement detection using positional de bruijn graph assembly. Genome Res. 2017;27(12):2050–60. https://doi.org/10.1101/gr.222109.117.
12. Lappalainen I, Lopez J, Skipper L, Hefferon T, Spalding JD, Garner J, Chen C, Maguire M, Corbett M, Zhou G, et al. Dbvar and dgva: public archives for genomic structural variation. Nucleic Acids Res. 2013;41(D1):936–41.
13. Abnizova I, te Boekhorst R, Orlov Y. Computational errors and biases of short read next generation sequencing. J Proteomics Bioinform. 2017;10(1):1–17.
14. Conrad DF, Bird C, Blackburne B, Lindsay S, Mamanova L, Lee C, Turner DJ, Hurles ME. Mutation spectrum revealed by breakpoint sequencing of human germline cnvs. Nat Genet. 2010;42(5):385.
15. Kidd JM, Graves T, Newman TL, Fulton R, Hayden HS, Malig M, Kallicki J, Kaul R, Wilson RK, Eichler EE. A human genome structural variation sequencing resource reveals insights into mutational mechanisms. Cell. 2010;143(5):837–47.
16. Ottaviani D, LeCain M, Sheer D. The role of microhomology in genomic structural variation. Trends Genet. 2014;30(3):85–94.
17. Sedlazeck FJ, Rescheneder P, Smolka M, Fang H, Nattestad M, von Haeseler A, Schatz MC. Accurate detection of complex structural variations using single-molecule sequencing. Nat Methods. 2018;15(6):461–8.
18. Chaisson MJP, Sanders AD, ..., Marschall T, Korbel J, Eichler EE, Lee C. Multi-platform discovery of haplotype-resolved structural variation in human genomes. Nat Commun. 2019;10:1784. https://doi.org/10.1038/s41467-018-08148-z.
19. Zook JM, Hansen NF, ..., Chaisson MJ, Spies N, Sedlazeck FJ, Salit M, the Genome in a Bottle Consortium. A robust benchmark for detection of germline large deletions and insertions. Nat Biotechnol. 2020. https://doi.org/10.1038/s41587-020-0538-8.
20. Chen X, Schulz-Trieglaff O, Shaw R, Barnes B, Schlesinger F, Källberg M, Cox AJ, Kruglyak S, Saunders CT. Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. Bioinforma (Oxford, England). 2016;32:1220–2. https://doi.org/10.1093/bioinformatics/btv710.
21. Boycott KM, Vanstone MR, Bulman DE, MacKenzie AE. Rare-disease genetics in the era of next-generation sequencing: discovery to translation. Nat Rev Genet. 2013;14(10):681–91.
22. Wellenreuther M, Mérot C, Berdan E, Bernatchez L. Going beyond snps: the role of structural genomic variants in adaptive evolution and species diversification. Mol Ecol. 2019;28(6):1203–9.
23. Chander V, Gibbs RA, Sedlazeck FJ. Evaluation of computational genotyping of structural variation for clinical diagnoses. GigaScience. 2019;8(9):giz110. https://doi.org/10.1093/gigascience/giz110.
24. Krusche P, Trigg L, Boutros PC, Mason CE, Francisco M, Moore BL, Gonzalez-Porta M, Eberle MA, Tezak Z, Lababidi S, et al. Best practices for

benchmarking germline small-variant calls in human genomes. Nat Biotechnol. 2019;37(5):555–60.
25. Mohiyuddin M, Mu JC, Li J, Bani Asadi N, Gerstein MB, Abyzov A, Wong WH, Lam HY. Metasv: an accurate and integrative structural-variant caller for next generation sequencing. Bioinformatics. 2015;31(16):2741–4.
26. Zarate S, Carroll A, Krashenina O, Sedlazeck FJ, Jun G, Salerno W, Boerwinkle E, Gibbs R. Parliament Fast structural variant calling using optimized combinations of callers. bioRxiv. 2018, 424267. https://doi.org/10.1101/424267. https://www.biorxiv.org/content/early/2018/09/23/424267.full.pdf.
27. Kuzniar A, Maassen J, Verhoeven S, Santuari L, Shneider C, Kloosterman WP, de Ridder J. sv-callers: a highly portable parallel workflow for structural variant detection in whole-genome sequence data. PeerJ. 2020;8:8214.
28. Dolzhenko E, Deshpande V, Schlesinger F, Krusche P, Petrovski R, Chen S, Emig-Agius D, Gross A, Narzisi G, Bowman B, et al. Expansionhunter: a sequence-graph-based tool to analyze variation in short tandem repeat regions. Bioinformatics. 2019;35(22):4754–6.
29. Kavak P, Lin Y-Y, Numanagić I, Asghari H, Güngör T, Alkan C, Hach F. Discovery and genotyping of novel sequence insertions in many sequenced individuals. Bioinformatics. 2017;33(14):161–9.
30. Kehr B, Melsted P, Halldórsson BV. Popins: population-scale detection of novel sequence insertions. Bioinformatics. 2016;32(7):961–7.
31. Soylev A, Le TM, Amini H, Alkan C, Hormozdiari F. Discovery of tandem and interspersed segmental duplications using high-throughput sequencing. Bioinformatics. 2019;35(20):3923–30.
32. Benson G. Tandem repeats finder: a program to analyze dna sequences. Nucleic Acids Res. 1999;27(2):573–80.
33. Hubley R, Finn RD, Clements J, Eddy SR, Jones TA, Bao W, Smit AF, Wheeler TJ. The dfam database of repetitive dna families. Nucleic Acids Res. 2016;44(D1):81–89.
34. Kent WJ. Blat–the blast-like alignment tool. Genome Res. 2002;12(4):656–64.
35. Harrow J, Frankish A, Gonzalez JM, Tapanari E, Diekhans M, Kokocinski F, Aken BL, Barrell D, Zadissa A, Searle S, et al. Gencode: the reference human genome annotation for the encode project. Genome Res. 2012;22(9):1760–74.
36. Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. The human genome browser at ucsc. Genome Res. 2002;12(6):996–1006.
37. Smit AFA, Hubley R, Green P. Repeatmasker open-3.0. 1996-2010. http://www.repeatmasker.org.
38. Jurka J. Repbase update: a database and an electronic journal of repetitive elements. Trends Genet. 2000;16(9):418–20.
39. Huang W, Li L, Myers JR, Marth GT. Art: a next-generation sequencing read simulator. Bioinformatics. 2012;28(4):593–4.
40. Faust GG, Hall IM. Samblaster: fast duplicate marking and structural variant read extraction. Bioinformatics. 2014;30(17):2503–5.
41. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. The sequence alignment/map format and samtools. Bioinformatics. 2009;25(16):2078–9.
42. Stöcker BK, Köster J, Rahmann S. Simlord: simulation of long read data. Bioinformatics. 2016;32(17):2704–6.
43. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34(18):3094–100.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Ready to submit your research? Choose BMC and benefit from:**

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

**At BMC, research is always in progress.**

**Learn more** biomedcentral.com/submissions

## A.3 Publication 3

### SVJedi : Genotyping structural variations with long reads

Lolita Lecompte, Pierre Peterlongo, Dominique Lavenier, **Claire Lemaitre**.
*Bioinformatics* 2020 36(17) :4568–4575

*Bioinformatics*, 36(17), 2020, 4568–4575
doi: 10.1093/bioinformatics/btaa527
Advance Access Publication Date: 21 May 2020
Original Paper

Sequence analysis

# SVJedi: genotyping structural variations with long reads

**Lolita Lecompte *, Pierre Peterlongo , Dominique Lavenier and Claire Lemaitre**

Univ Rennes, Inria, CNRS, IRISA, F-35000 Rennes, France

*To whom correspondence should be addressed.

Associate Editor: Peter Robinson

Received on November 22, 2019; revised on March 27, 2020; editorial decision on May 10, 2020; accepted on May 18, 2020

## Abstract

**Motivation:** Studies on structural variants (SVs) are expanding rapidly. As a result, and thanks to third generation sequencing technologies, the number of discovered SVs is increasing, especially in the human genome. At the same time, for several applications such as clinical diagnoses, it is important to genotype newly sequenced individuals on well-defined and characterized SVs. Whereas several SV genotypers have been developed for short read data, there is a lack of such dedicated tool to assess whether known SVs are present or not in a new long read sequenced sample, such as the one produced by Pacific Biosciences or Oxford Nanopore Technologies.

**Results:** We present a novel method to genotype known SVs from long read sequencing data. The method is based on the generation of a set of representative allele sequences that represent the two alleles of each structural variant. Long reads are aligned to these allele sequences. Alignments are then analyzed and filtered out to keep only informative ones, to quantify and estimate the presence of each SV allele and the allele frequencies. We provide an implementation of the method, SVJedi, to genotype SVs with long reads. The tool has been applied to both simulated and real human datasets and achieves high genotyping accuracy. We show that SVJedi obtains better performances than other existing long read genotyping tools and we also demonstrate that SV genotyping is considerably improved with SVJedi compared to other approaches, namely SV discovery and short read SV genotyping approaches.

**Availability and implementation:** https://github.com/llecompte/SVJedi.git

**Contact:** lolita.lecompte@inria.fr

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

## 1 Introduction

Structural variants (SVs) are characterized as genomic segments of at least 50 base pair (bp) long that are rearranged in the genome. There are several types of SVs such as deletions, insertions, duplications, inversions or translocations. With the advent of next-generation sequencing (NGS) technologies and the re-sequencing of many individuals in populations, SVs have been admitted as a key component of human polymorphism (Audano *et al.*, 2019). This kind of polymorphism has been shown involved in many biological processes such as diseases or evolution (Lupski, 2015). Databases referencing such variants grow as new variants are discovered. At this time, dbVar, the reference database of human genomic SVs (Phan *et al.*, 2017) now contains more than 36 million variant calls, illustrating that many SVs have already been discovered and characterized in human populations.

When studying SV in newly sequenced individuals, one can distinguish two distinct problems: discovery and genotyping. In the SV discovery problem, the aim is to identify all the variants that differentiate the given re-sequenced individual with respect usually to a reference genome. In the SV genotyping problem, the aim is to evaluate if a given known SV (or set of SVs) is present or absent in the re-sequenced individual, and assess, if it is present, with which ploidy (heterozygous or homozygous). At first glance, the genotyping problem may seem included in the discovery problem, since present SVs should be discovered by discovery methods. However, in discovery algorithms, SV evidence is only investigated for present variants (i.e. incorrect mappings) and not for absent ones. If an SV has not been called, we cannot know if the caller missed it (false negative) or if the variant is truly absent in this individual and this could be validated by a significant amount of correctly mapped reads in this region. Moreover, in the genotyping problem, knowing what we are looking for should make the problem simpler and the genotyping result hopefully more precise. With the fine characterization of a growing number of SVs in populations of many organisms, genotyping newly sequenced individuals becomes very interesting and informative, in particular in human medical diagnosis contexts or more generally in any association or population genomics studies.

In this work, we focus on the second problem: genotyping already known SVs in a newly sequenced sample. Such genotyping methods already exist for short read data (Alkan *et al.*, 2011; Chander *et al.*, 2019): for instance, SVtyper (Chiang *et al.*, 2015),

© The Author(s) 2020. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com


Downloaded from https://academic.oup.com/bioinformatics/article/36/17/4568/5841661 by INRIA Rennes user on 18 December 2020

SV$^2$ (Antaki *et al.*, 2018), Nebula (https://www.biorxiv.org/content/10.1101/566620v1). Though short reads are often used to discover and genotype SVs, this is well known that their short size makes them ill-adapted for predicting large SVs or SVs located in repeated regions. SVs are often located alongside repeated sequences such as mobile elements (Kidd *et al.*, 2010), resulting in mappability issues that make the genotyping problem harder when using short read data.

Third generation sequencing technology, such as Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT), can produce much longer reads compared to NGS technologies. Despite their high error rate, long reads are crucial in the study of SVs and have enabled new SV discoveries (Huddleston *et al.*, 2017; Jain *et al.*, 2018; Norris *et al.*, 2016; Stancu *et al.*, 2017). Indeed, the size range of these sequences can reach a few kilobases (kb) to megabases, thus long reads can extend over rearranged sequence portions as well as over the repeated sequences often present at SV's breakpoint regions.

Following long read technology's development, many SV discovery tools have emerged, such as Sniffles (Sedlazeck *et al.*, 2018), NanoSV (Stancu *et al.*, 2017), pbsv (unpublished) or SVIM (Heller and Vingron, 2019). Among these tools, some implement a genotyping module that gives the frequency of alleles after calling SVs of the sequenced samples. Nonetheless, discovery tools require post-processing to evaluate if a set of SVs is present or not in the sample and to compare the SV calls between different samples. To our knowledge, there exist only two tools that can perform genotyping with long read data for a given set of SVs, the discovery tool Sniffles with a specific option and the SV visualization tool svviz2, but both are not purely dedicated to the genotyping task.

The main contribution of this work is a novel method to genotype known SVs using long read data. We also provide an implementation of this method in the tool named SVJedi. SVJedi accuracy and robustness were evaluated on simulated data of real deletions in a human chromosome. It was also applied to a real human dataset and compared to a gold standard call set provided by the Genome in a Bottle (GiaB) Consortium, containing both deletions and insertions. High genotyping accuracy was achieved on both simulated and real data. We also demonstrated the improvement of such a dedicated method over other long read genotyping tools and other approaches, namely SV discovery with long reads and SV genotyping with short reads.

## 2 Methods

The method assigns a genotype for a set of already known SVs in a given individual sample sequenced with long read data. It assesses for each SV if it is present in the given individual, and if so, how many variant alleles it holds, i.e. whether the individual is heterozygous or homozygous for the particular variant.

For clarity purposes, we describe here the method for deletion genotyping only. The genotyping of insertions is perfectly symmetrical, the genotyping of inversions and translocations differs only by the number of breakpoints to examine and follows the same strategy. The method takes as input a variant file with SV coordinates in VCF format, a reference genome and the sample of long read sequences. It outputs a variant file complemented with the individual genotype information for each input variant in VCF format.

The method consists of four different steps that are illustrated in Figure 1. The fundamentals of the method lie in its first step, which generates *representative allele sequences* that represent the two alleles of each SV. Long reads are then aligned on the whole set of representative allele sequences. An important step consists in selecting and counting only informative alignments to finally estimate the genotype for each input variant.

### 2.1 Representative allele sequence generation

Starting from a known variant file in VCF format and the corresponding reference genome, the first step consists in generating two sequences for each SV, corresponding to the two possible allele sequences. These representative allele sequences are hereafter simply called *allele sequences*. In the case of deletions, these are parts of the reference genome that may be absent in a given individual. They are characterized in the VCF file by a starting position on the reference genome and a length. We define the reference allele sequence (allele 0) as the sequence of the deletion with adjacent sequences at each side, and the alternative allele sequence (allele 1) consists in the joining of the two previous adjacent sequences. Given that reads of several kilobytes will be mapped on these allele sequences, the size of the adjacent sequences, denoted by $L_{\text{adj}}$, was set to 5000 bp at each side, giving a 10 kb sequence for allele 1 and 10 kb plus the deletion size for allele 0. For deletions larger than $2 \times L_{\text{adj}}$, that is here larger than 10 kb, two representative sequences are generated for allele 0, one for each breakpoint. The same adjacent sequence size is used, i.e. 5000 bp, on each side of the breakpoints, giving then three 10 kb sequences: one for allele 1, and two for allele 0 (left and right breakpoints).

### 2.2 Mapping

Sequenced long reads are aligned on all previously generated allele sequences, using Minimap2 (Li, 2018) (version 2.17-r941). Option -c is specified to generate a CIGAR for each alignment. Alignments are then output in a pairwise read mapping format (PAF) file.

### 2.3 Informative alignment selection

Minimap2 raw alignment results have to be carefully filtered out to remove (i) uninformative alignments, which are those not discriminating between the two possible alleles and (ii) spurious false-positive alignments that are mainly due to repeated sequences.

Informative alignments for the genotyping problem are those that overlap the SV breakpoints that is the sequence adjacencies that are specific to one or the other allele. In the case of a deletion, the reference allele contains two such breakpoints, the start and end positions of the deletion sequence; the alternative sequence, the shortest one, contains one such breakpoint at the junction of the two adjacent sequences (see the red thick marks of Fig. 1).

To be considered as overlapping a breakpoint, an alignment must cover at least $d_{\text{over}}$ bp from each side of the breakpoint ($d_{\text{over}}$ is set by default to 100 bp). In other words, if $x$ and $y$ are the sizes of the aligned parts on the allele sequence at the, respectively, left and

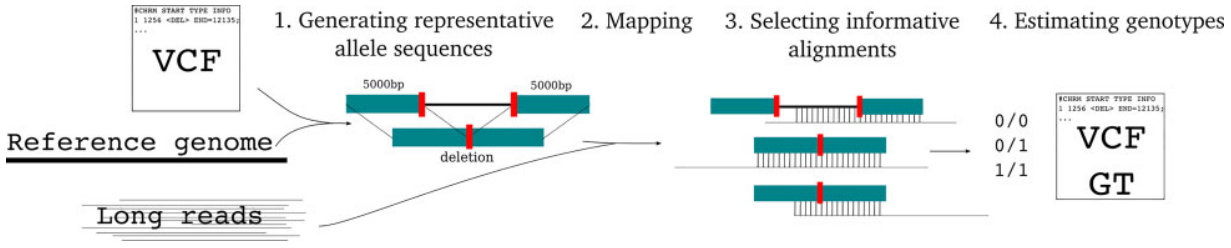


**Fig. 1.** SVJedi steps for deletion genotyping. Steps for insertion genotyping are symmetrical and are not shown on the figure for clarity purposes. 1. Two corresponding representative allele sequences are generated for each selected SV, one corresponds to the original sequence and the other to the sequence with the deletion. 2. Long read sequenced data are aligned on these allele sequences using Minimap2. 3. Informative alignments are selected. 4. Genotypes are estimated

Downloaded from https://academic.oup.com/bioinformatics/article/36/17/4568/5841661 by INRIA Rennes user on 18 December 2020

right sides of the breakpoint, they must satisfy the following condition in Equation (1) for the alignment to be kept:

$$(x > d_{over}) \,\&\, (y > d_{over}) \quad (1)$$

Concerning the filtering of spurious false-positive alignments, Minimap2 alignments are first filtered based on the mapping quality (MAPQ) score. To focus uniquely on mapped reads, the MAPQ score of the alignments must be >10. This is not sufficient to filter out alignments due to repetitive sequences since mapping is performed on a small subset of the reference genome and these alignments may appear as uniquely mapped on this subset.

As Minimap2 is a sensitive local aligner, many of the spurious alignments only cover subsequences of both the allele and the read sequences. To maximize the probability that the aligned read originates from the genomic region holding the SV, we, therefore, require the read to be aligned with the allele sequence in a semi-global manner. Each alignment extremity must correspond to an extremity of at least one of the two aligned sequences. This criterion gathers four types of situations, namely the read is included in the allele sequence, or *vice-versa*, or the read left end aligns on the allele sequence right end or *vice-versa*.

Indeed this criteria is not strictly applied and a distance of $d_{end}$ of the alignment to an extremity of at least one of the two aligned sequences is tolerated ($d_{end}$ is set by default to 100 bp). More formally, if $a_1$ and $a_2$ (resp. $r_1$ and $r_2$) are the sizes of the unaligned parts at the, respectively, left and right sides of the alignment on the allele sequence (resp. read sequence) (see Fig. 2), then the alignment must fulfill the following condition in Equation (2) to be kept:

$$(a_1 < d_{end} \,||\, r_1 < d_{end}) \,\&\, (a_2 < d_{end} \,||\, r_2 < d_{end}) \quad (2)$$

The left member of Equation (2) imposes that the unaligned part at the left of the alignment is small in at least one of the two aligned sequences; the right member imposes the same condition at the right side of the alignment.

### 2.4 Genotype estimation

For each variant, the genotype is estimated based on the ratio of amounts of reads informatively aligned to each allele sequence. In the case of deletions and insertions, the allele sequences of a given variant are of different sizes and contain a different number of breakpoints (for a deletion for instance, the reference allele contains two breakpoints, whereas the alternative allele contains only one), so even if both alleles are covered with the same read depth, there would be fewer reads that align on the shortest allele sequence and that overlap at least one breakpoint. To prevent a bias toward the longest allele, reported read counts for the longest allele are normalized according to the allele sequence length ratio, assuming that read count is proportional to the sequence length. More precisely, in the case of a deletion, the reference allele is the longest allele. Its allele sequence size is the deletion size plus $2 \times L_{adj}$ (cumulative size of the adjacent sequences) if the deletion is smaller than $2 \times L_{adj}$.

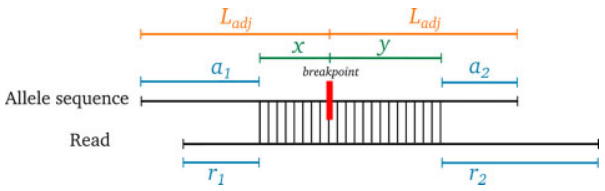


**Fig. 2.** Definition of the different distances used to select informative alignments between an allele sequence and a read. The aligned parts of the sequences are illustrated by vertical bars. The allele sequence is composed of two adjacent sequences of size $L_{adj}$ on either side of the breakpoint, which is represented by a red vertical thick bar. $x$ and $y$ are the distances of the breakpoint to, respectively, the start and end coordinates of the alignment on the allele sequence. $a_1$ and $a_2$ (resp. $r_1$ and $r_2$) are the distances of the alignment left and right extremities to the, respectively, left and right extremities of the allele sequence (resp. read sequence). It follows here, that $a_1 + x = L_{adj}$ and $a_2 + y = L_{adj}$

Otherwise it is composed of two sequences of size $2 \times L_{adj}$ each centered on each breakpoint. We therefore apply the following formula to compute the normalized read count for the reference allele, $c_0^*$, as a function of the observed read count for the reference allele, $c_0$, and the deletion size, $del_{size}$:

$$c_0^* = \begin{cases} c_0 \times \dfrac{2 \times L_{adj}}{(2 \times L_{adj} + del_{size})} & \text{if} \quad del_{size} < 2 \times L_{adj} \\ c_0 \times \dfrac{1}{2} & \text{otherwise} \end{cases} \quad (3)$$

Finally, a genotype is estimated if the variant presence or absence is supported by at least $min_{cov}$ different reads after normalization (sum of the read counts for each allele). By default, this parameter is set to 3.

Genotypes are estimated according to a maximum likelihood strategy. The likelihoods of the three possible genotypes given the observed normalized read counts ($c_0^*$ and $c_1$) are computed based on a simple binomial model, assuming a diploid individual, as described in Nielsen *et al.* (2011) [see also (Li, 2011)]:

$$\ell(0/0) = (1 - err)^{c_0^*} \times err^{c_1} \times C^{c_0^*}_{c_0^* + c_1} \quad (4)$$

$$\ell(1/1) = err^{c_0^*} \times (1 - err)^{c_1} \times C^{c_0^*}_{c_0^* + c_1} \quad (5)$$

$$\ell(0/1) = \left(\frac{1}{2}\right)^{c_0^* + c_1} \times C^{c_0^*}_{c_0^* + c_1} \quad (6)$$

where *err* is the probability that a read maps to a given allele erroneously, assuming it is constant and independent between all observations. *err* was fixed to $5.10^{-5}$, after empirical experiments on a simulated dataset (see Supplementary Fig. S1).

Finally, the genotype with the largest likelihood is assigned and all three likelihoods are also output (-log10 transformed) as additional information in the VCF file.

### 2.5 Implementation and availability

We provide an implementation of this method named SVJedi, freely available at https://github.com/llecompte/SVJedi, under the GNU Affero GPL license. SVJedi can also be installed from Bioconda. SVJedi is written in Python 3, it requires as input a set of SVs (VCF format), a reference genome (fasta format) and a sequencing read file (fastq or fasta format). Notably, the main steps are implemented in a modular way, allowing the user to start or re-run the program from previous intermediate results. As an example, the first step is not to be repeated if there are several long read datasets to be genotyped on the same SV set. Results shown here were obtained with release version 1.1.0.

## 3 Materials

### 3.1 Long read simulated dataset

SVJedi was assessed on simulated datasets on the human chromosome 1 (assembly GRCh37) based on real characterized deletions for the human genome. From the dbVar database (Phan *et al.*, 2016), we selected 1000 existing deletions on chromosome 1 (defined as <DEL> SV type), which are separated by at least 10 000 bp. The sizes of the deletions vary from 50 bp to 10 kb (with median and average sizes of 950 and 2044 bp, respectively). In this experiment, deletions were distributed into the three different genotypes: 333 deletions are simulated with 0/0 genotype, 334 deletions with 0/1 genotype and the 333 remaining deletions with 1/1 genotype. Two different sequences were simulated containing each overlapping sets of deletions, representing the two haplotypes of the simulated individual. 1/1 genotype deletions were simulated on both haplotype sequences, whereas deletions of 0/1 genotype were simulated each on one randomly chosen of the two haplotype sequences. Then PacBio data were simulated on both haplotypes, using

Downloaded from https://academic.oup.com/bioinformatics/article/36/17/4568/5841661 by INRIA Rennes user on 18 December 2020

SimLoRD (Stöcker *et al.*, 2016) (version v1.0.2) with varying sequencing error rates (6%, 10%, 16% and 20%), and at varying total sequencing depths (6×, 10×, 16×, 20×, 30×, 40×, 50× and 60×). Most results presented in the main text are for 16% error rate and 30× sequencing depth. Ten such datasets were simulated to assess the reproducibility of results.

### 3.2 Real data

SVJedi was applied on a real human dataset, from the individual HG002, son of the so-called *Ashkenazi trio* dataset. A PacBio Continuous Long Read (CLR) sequencing dataset for HG002 was downloaded from the FTP server of GiaB and down-sampled to 30× read depth (FTP links are given in Supplementary Material). We considered the assembly GRCh37.p13 as the human genome reference and as a gold standard call set, we used the SV benchmark set (v0.6) of HG002 individual provided by the GiaB Consortium (https://www.biorxiv.org/content/10.1101/664623v3). This set contains 5464 high confidence deletions and 7281 insertions (PASS filter tag), whose sizes range from 50 bp to 125 kb (median sizes of 149 and 215 bp for deletions and insertions, respectively). We used the TRgt100 = TRUE tags present in the GiaB VCF file to identify SVs located in Tandem Repeats >100 bp (denoted as TRs, $n = 6469$). Forty-eight SVs were found located inside large (>10 kb) segmental duplications, using the UCSC Segmental Dups feature track (Bailey *et al.*, 2002).

These SVs were also genotyped in PacBio sequencing datasets of the two parents (HG003 and HG004, 30× and 27×, respectively) to assess the level of Mendelian inheritance consistency of the son predicted genotypes.

SVJedi was applied on a real human ONT PromethION 44× dataset for the individual HG002 as well. Finally, we considered a real short read dataset for the HG002 individual, 2 × 250 bp Illumina dataset from GiaB that was down-sampled to 30× read depth. This short read dataset is used for comparison with a short read based SV genotyping approach. FTP links for all real sequencing datasets are given in Supplementary Material Section 1).

### 3.3 Evaluation

To evaluate the accuracy of the method, a contingency table between the estimated genotypes and the true (simulated) ones is computed, providing a clear view of the number and type of correctly and incorrectly estimated genotypes. The genotyping accuracy of the method is then assessed as the number of correctly estimated genotypes overall all estimated genotypes, as shown in Equation (7). The percentage of SVs for which a genotype could be estimated is also measured, and hereafter called the genotyping rate [Equation (8)].

$$\text{Genotyping accuracy} = \frac{\text{of correctly estimated genotypes}}{\text{of estimated genotypes}} \quad (7)$$

$$\text{Genotyping rate} = \frac{\text{of estimated genotypes}}{\text{of known SVs}} \quad (8)$$

### 3.4 Comparison with other genotyping approaches

Comparisons with other genotyping approaches were performed on the real PacBio 30× HG002 dataset.

SVJedi was first compared to two tools, Sniffles (Sedlazeck *et al.*, 2018) and svviz2 (Spies *et al.*, 2015). Both tools, although not dedicated to genotyping, have options that allow them to also do SV genotyping from a set of SVs and with a long read sequencing dataset. Following the recommendations of Sniffles (https://github.com/fritzsedlazeck/Sniffles/wiki/SV-calling-for-a-population), reads were first aligned with NGMLR (version 0.2.7) on the human reference genome. Then, we used Sniffles (version 1.0.11) with the –Ivcf option to genotype the GiaB call set. For svviz2, reads were aligned on the human reference genome using Minimap2 (version 2.17-r941). Genotyping was then performed from the sorted Minimap2 alignments using svviz2 (version 2.0a3) with default parameters.

**Table 1.** Contingency table of SVJedi results on PacBio simulated data (30x) of human chromosome 1 with 1000 deletions from dbVar

| | | SVJedi predictions | | | |
|---|---|---|---|---|---|
| | | 0/0 | 0/1 | 1/1 | ./. |
| Truth | 0/0 | **331** | 1 | 0 | 1 |
| | 0/1 | 3 | **330** | 0 | 1 |
| | 1/1 | 0 | 18 | **313** | 2 |
| | | Genotyping accuracy: 97.8 % | | | |

*Notes:* SVJedi genotype predictions are indicated by column and the expected genotypes are shown by row. The genotype './.' column corresponds to deletions for which SVJedi could not assess the genotype.

We also compared our approach with two SV discovery tools, Sniffles again but in its default mode and Pacific Biosciences SV caller, pbsv (https://github.com/PacificBiosciences/pbsv). Sniffles were run with the –genotype parameter with the previously obtained NGMLR read alignments. For pbsv, reads were aligned to the reference genome using its own mapper pbmm2 (version 1.1.0) with the –sort, –median-filter and –sample parameters. SVs were then discovered and called with pbsv (version 2.2.2) using default parameters. Both Sniffles and pbsv analyses do not always predict SVs at the exact simulated coordinates, so a predicted SV is considered identical as the expected SV if both SVs overlap by at least 70%.

Finally, SVJedi was also compared to an SV genotyping approach based on short read data. To do this, the short reads are first aligned with SpeedSeq (Chiang *et al.*, 2015) (version 0.1.2), then the known variants are genotyped with SVtyper (version 0.7.0) with the default settings.

All tools were run on a Linux 40-CPU node running at 2.60 GHz, all command lines are given in Supplementary Material Section 2.

## 4 Results

### 4.1 Assessing SVJedi accuracy and robustness on simulated deletions

To comprehensively assess the accuracy and robustness of SVJedi, it was first applied to simulated data. Results for SVJedi are shown here only for deletion type SVs, as insertions variants are simply the counterpart of deletions, results for inversions and translocations are shown in Supplementary Table S1. PacBio long reads were simulated on artificial diploid genomes obtained by introducing deletions in the human chromosome 1. Importantly, the sets of introduced and genotyped deletions are made of real characterized deletions in human populations, to reflect the real size distribution and the real complexity of deletion breakpoints and neighboring genomic contexts. To do so, 1000 deletions located on human chromosome 1 were randomly selected from the dbVar database, ranging from 50 to 10 000 bp in size.

Table 1 shows the obtained genotypes compared with expected ones for one simulated dataset at 30× read depth. On this dataset, SVJedi achieves 97.8% genotyping accuracy, with 974 deletions correctly predicted over 996 with an assigned genotype. Among the 1000 assessed deletions, only 4 could not be assigned a genotype due to insufficient coverage of informative reads, the genotyping rate being thus 99.6%. Among the few genotyping errors, most concern 1/1 genotypes that were incorrectly predicted as 0/1.

SVJedi genotyping accuracy results were evaluated in terms of varying sequencing depths, ranging from 6× to 60× (see Fig. 3). As expected, the accuracy of SVJedi increases with the read depth, but interestingly, even at low coverage (6×) the accuracy is on average above 94% and a plateau is quickly reached between 20× and 30×, with already 97% of genotyping accuracy at 20×. The genotyping rate reaches its plateau at a sequencing depth of 16×.

Similarly, SVJedi results were evaluated in terms of varying sequencing error rates. In this case, both genotyping accuracy and

Downloaded from https://academic.oup.com/bioinformatics/article/36/17/4568/5841661 by INRIA Rennes user on 18 December 2020

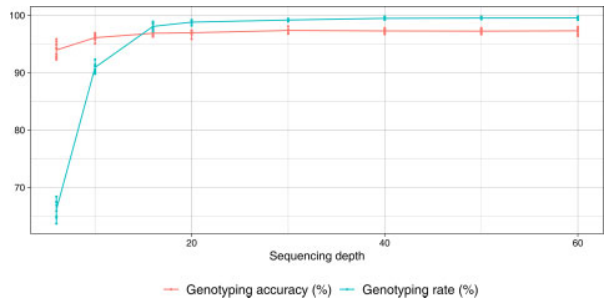


**Fig. 3.** SVJedi genotyping accuracy results as a function of the sequencing depth for nine simulated PacBio datasets of human chromosome 1, containing 1000 deletions from the dbVar database. The red dots correspond to the average genotyping accuracy and the red segments represent the standard deviations, at each sequencing depth

genotyping rate were not impacted by a lower or higher sequencing error rate as long as it stays realistic (see Supplementary Fig. S2).

The breakpoint coordinates of SVs detected by SV discovery methods are not always defined at the base pair resolution. To assess to what extent this potential imprecision can impact the genotyping accuracy of SVJedi, we performed experiments with altered breakpoint positions in the input variant VCF file. All breakpoint positions have been randomly shifted according to a Normal distribution centered on the exact breakpoint position with several standard deviations ($\sigma$) values ranging from 10 to 100 bp. We show that the genotyping accuracy with $\sigma$ equals 50 bp does not fall below 94% (see Supplementary Fig. S3), indicating that SVJedi is not much impacted by the exact definition of the positions of the reference breakpoints.

## 4.2 Application of SVJedi to a real human dataset

To get closer to the reality of biological data, we applied our tool to a real human dataset, the HG002 individual, son of the so-called *Ashkenazi trio* dataset, which has been highly sequenced and analyzed in various benchmarks and especially by the GiaB Consortium (Zook *et al.*, 2016). The latter, precisely, provides a set of high confidence SV calls together with their genotype in the individual HG002. SV discovery and genotyping were based on several sequencing technologies, SV callers and careful call set merging. Their work estimated genotypes for 5464 deletions and 7281 insertions, which can then be considered as the ground truth. It should be noted that we can focus only on heterozygous (0/1) and homozygous for the alternative allele (1/1) genotypes. Indeed, the SV call set was obtained from SV discovery methods, which can only detect variations between the individual and the reference genome. SVJedi was applied on a 30× PacBio long read dataset from individual HG002, to assess the genotypes of both deletions and insertions of this high confidence set.

### 4.2.1 SVJedi results on the HG002 individual

We observe a good overlap of 92.2% between the estimated genotypes of SVJedi and the GiaB call set. More precisely, among the assigned genotypes, there are 91.7% of deletions and 92.5% of insertions that are genotyped by SVJedi identically as the GiaB call set (the detailed contingency tables are provided in Supplementary Table S2).

Among the SVs differently genotyped between SVJedi and GiaB, a large part is represented by small variants (57% are smaller than 100 bp). The genomic context of the SVs seems also to impact the genotyping accuracy: 75% of the differently genotyped variants are located in Tandem Repeats >100 bp (TRs), compared to 51% for the whole SV set. Both features, size and location in TR, have similar impacts on the genotyping accuracy, with a difference of 9% and 11% for small-versus-large and TR-versus-non-TR located SVs, respectively. Combining the two factors leads to a larger difference, with the small SVs that are located in TRs having the lowest genotyping accuracy of 81.3% compared to near perfect accuracy of 97.9% for the larger ones outside TRs (see the cross table in Supplementary Table S3).

Compared to previous results on simulated data, SVJedi shows a lower genotyping rate on this real dataset, for both deletions and insertions (85.8% and 93.6%, respectively). As in the case of accuracy, we notice that the great majority of the non-genotyped variants are either small or located in TRs: 87% are of size <100 bp and 84% are located in TRs. The factor impacting most the genotyping rate is the SV size (genotyping rates of 74.6% and 98.1% for small and large SVs, respectively, see Supplementary Table S3). The presence of a TR at the breakpoint of small SVs worsens the genotyping task, with only 68.2% of such SVs that could be genotyped. Notably, the GiaB deletion set contains more in proportion of such small SVs (39% versus 29% for deletions and insertions, respectively), explaining the observed difference in genotyping rate between the two SV types. Interestingly, these kinds of variants seem to be more impacted by the heterogeneity of PacBio sequencing depth since when using the full 63× dataset, the overall genotyping rate increases to 96.6%.

### 4.2.2 Mendelian inheritance analysis

Since sequencing data are available for the parents of the studied individual (HG003 for the father and HG004 for the mother), we can check, as an alternative validation approach, if the predicted genotypes for the son are consistent with his parent genotypes, assuming perfect Mendelian inheritance and a very low *de novo* mutation rate. To do so, from the same set of deletions and insertions, which is the GiaB call set, SVJedi was applied to three PacBio sequence datasets, one per individual, with a sequencing depth of about 30× for each dataset. Overall, the Mendelian inheritance consistency of SVJedi on this trio dataset is high, with 96.9% of the son genotypes that are consistent with his parent genotypes. As expected, most inconsistent genotypes concern SVs that were genotyped differently between SVJedi and GiaB (48.7%, $n = 154$), confirming for those that they are probably wrongly assessed by SVJedi. However, these confirmed errors represent only 1.2% of the dataset.

### 4.2.3 SVJedi results on ONT data

SVJedi was applied on the same SV call set and for the same HG002 individual, but with sequencing data obtained by a different long read technology, namely Oxford Nanopore. With a 44× PromethION dataset, SVJedi shows very similar genotyping performances as with the PacBio dataset (90.7% accuracy and 86.2% rate, see Supplementary Table S4), highlighting its versatility with respect to long read sequencing technologies.

## 4.3 Comparison with other approaches

### 4.3.1 Comparison with other genotyping tools

SVJedi was compared on the PacBio HG002 dataset to two other tools that can genotype a set of SVs with long read sequencing data, Sniffles (Sedlazeck *et al.*, 2018) and svviz2 (Spies *et al.*, 2015). Both tools are not purely dedicated to the genotyping problem. Sniffles is an SV discovery tool that has an option (–Ivcf) enabling a genotyping mode instead of a discovery mode, we will thereafter refer to this tool usage as Sniffles–Ivcf. svviz2 is a visualization tool enabling to visualize how reads align to the reference and alternative alleles of a given SV, as a byproduct it can estimate a genotype based on the aligned read counts.

As shown in Table 2, both Sniffles–Ivcf and svviz2 have genotyping rates close to 100% but at the expense of lower genotyping accuracies (detailed results for all genotypes are given in Supplementary Table S5). Sniffles–Ivcf is 10% less accurate than SVJedi (82.0% versus 92.2%). svviz2 obtained the lowest genotyping accuracy (65.9%, with a 10% difference between deletions and insertions).

A stratified analysis of the genotyping performances of all three tools with respect to the SV size is presented in Figure 4. We can observe that SVJedi has a better accuracy than Sniffles–Ivcf for all SV size classes. As mentioned previously, the lowest accuracy of SVJedi

Downloaded from https://academic.oup.com/bioinformatics/article/36/17/4568/5841661 by INRIA Rennes user on 18 December 2020

**Table 2.** Comparison of several tools and approaches for genotyping the 12 745 deletions and insertions of the GiaB call set in the HG002 individual

| | Deletions | | Insertions | | |
|---|---|---|---|---|---|
| Tool | Genotyping accuracy | Genotyping rate | Genotyping accuracy | Genotyping rate | Time |
| SVJedi | **91.7** | 85.8 | **92.5** | 93.6 | **2 h 25min** |
| Sniffles–Ivcf | 82.5 | **99.9** | 81.7 | **99.8** | 17 h 16 min |
| svviz2 | 72.5 | **100** | 61.0 | **100** | 5 days[a] |
| SVtyper (Illumina dataset) | 46.5 | 99.2 | – | – | 5 h 32 min |
| Sniffles (discovery mode) | 48.7 | 52.4 | 39.8 | 44.8 | 18 h 4 min |
| pbsv | 90.1 | 72.7 | 68.8 | 59.8 | 5 h 29 min |

*Notes:* Three approaches are compared: using long read genotyping tools (first three tools), using a short read genotyping tool (SVTyper), and using long read discovery tools (last two tools). Except for the short read genotyping tool (SVtyper) that uses a 30× Illumina sequencing dataset, all other tools were run with a 30× PacBio long read dataset. Runtimes were measured on a 40-CPU computing node.

[a]svviz2 is not parallelized.

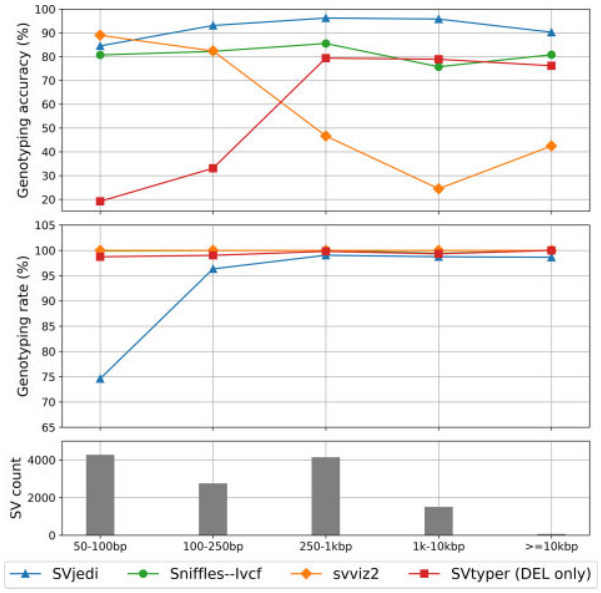


**Fig. 4.** Results of genotyping tools for the 12 745 deletions and insertions from the GiaB call set in the HG002 individual according to different SV size classes: 50–100 bp, 100–250 bp, 250 bp to 1 kb, 1–10 kb and ≥10 kb. The two figures on top represent the genotyping accuracies and the genotyping rates of SVJedi, Sniffles–Ivcf and svviz2 on a 30× PacBio dataset, and of SVtyper for deletions only on a 30× Illumina dataset. The bottom figure represents the SV count of each SV size class

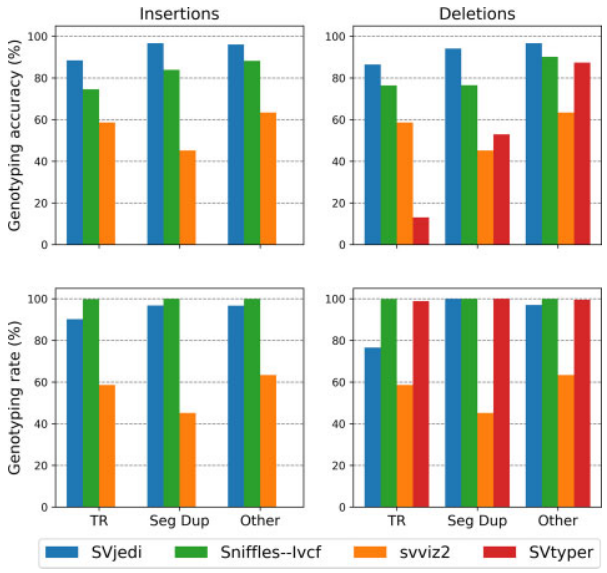


**Fig. 5.** Stratified analysis of genotyping accuracy and rate for several genotyping tools with respect to the genomic context of the SVs. Three categories of genomic context are considered: tandem repeats >100 bp (TR), segmental duplications larger than 10 kb (SeqDup) and all other regions (Other)

is observed for small SVs (<100 bp), but, apart from this size class, its accuracy is quite robust with respect to the size of SVs. On the opposite, svviz2 obtained its best genotyping accuracy for the smallest SVs (<100 bp) and it rapidly drops for SVs larger than 250 bp, falling below 30% for SV sizes between 1 and 10 kb. When comparing between SV types, svviz2 genotyping accuracy is significantly lower for insertions than deletions, with, in particular, <10% of the insertions larger than 1 kb that are correctly genotyped (see Supplementary Fig. S4). This inability for genotyping large SVs can probably be explained by the way svviz2 identifies informative reads for a given SV: only the reads mapped initially to the reference genome are selected before re-aligning them against both the reference and alternative alleles. Consequently, most reads coming from large insertion alternative alleles could probably not be used for estimating these genotypes. To a lesser extend, Sniffles–Ivcf genotyping accuracy is also lower for large insertions than large deletions (69.6% for insertions versus 85.7% for deletions, ≥1 kb), whereas SVJedi genotyping accuracy is unaffected by SV type for all size classes.

We then compared the genotyping performances with respect to the genomic context of the SVs (Fig. 5). As shown previously, SVs falling in a Tandem Repeat >100 bp are harder to genotype with SVJedi, with a 9% decrease of accuracy for these SVs, compared to those outside TRs. A larger decrease of genotyping accuracy (14%) is observed with Sniffles–Ivcf in these regions. Although less frequent (here, only 48 concerned SVs), large segmental duplications, typically larger than 10 kb, are also likely to affect long read mapping accuracy and thus genotyping accuracy. SVJedi accuracy seemed not to be affected by these duplications, contrary to Sniffles–Ivcf (Fig. 5).

#### 4.3.2 Comparison with a short read based genotyping approach

For this same individual (HG002), some short read datasets are also available. We, therefore, can compare SV genotyping performances between two approaches and data types, namely long versus short reads. SVJedi predictions were compared to an SV genotyping tool for short reads, SVtyper, known as a reference tool in the state of the art (Chiang *et al.*, 2015; Chander *et al.*, 2019). Since SVtyper does not support insertion variants, we focus here only on deletions,

Downloaded from https://academic.oup.com/bioinformatics/article/36/17/4568/5841661 by INRIA Rennes user on 18 December 2020

and the 5464 deletions from the GiaB call set were genotyped with SVtyper using a 2 × 250 bp 30x Illumina read dataset of HG002.

Table 2 shows that more than half of the deletions are genotyped differently by SVtyper than in the high confidence GiaB call set, resulting in a genotyping accuracy of only 46.5%, while this percentage rises to 91.7% for SVJedi with long reads. Remarkably, many of the discrepancies of SVtyper with GiaB are totally contradictory with 0/0 genotypes instead of 1/1 ones (see Supplementary Table S5). We can clearly see, in Figure 5, that short read based genotyping is much more impacted by the presence of TRs at the breakpoint. As expected, mapping reads in these regions is much more challenging for short than long reads. This demonstrates the higher benefit of using long reads and a dedicated genotyping tool such as SVJedi rather than short reads.

#### 4.3.3 Comparison with SV discovery approaches

One can wonder if these SVs could be easily detected and genotyped by long read SV discovery tools. We applied here two such tools, among the bests to date, Sniffles and the Pacific Biosciences SV caller, pbsv (Sedlazeck *et al.*, 2018; De Coster *et al.*, 2019). As a result, both tools obtained the lowest genotyping rates over all genotyping approaches: among the 12 745 SVs, only 6127 were discovered by Sniffles, and 8326 by pbsv (genotyping rates of 48.1% and 65.3%, respectively, see Table 2 and details in Supplementary Table S5). As expected, most of the missed SVs have an heterozygous genotype in the GiaB call set. More surprisingly, for the discovered SVs, their genotyping accuracy is overall smaller than with other approaches, with 43.9% and 78.9% for Sniffles and pbsv, respectively. In particular, Sniffles misassigns 85% of the discovered SVs with a 1/1 genotype in GiaB as heterozygous. pbsv shows the same type of errors but mainly for insertions, resulting in an important difference of genotyping accuracy between deletions and insertions (90.1% versus 68.8%). These results highlight the fact that SV discovery tools are much less precise for the genotyping task than a dedicated genotyping tool.

#### 4.3.4 Runtime comparison

Importantly, SVJedi does not come with a high computational cost. On a 40-CPU computing node, genotyping the 12 745 SVs with the 30× PacBio HG002 dataset took only 2 h 25 min. The alignment step is actually the most time-consuming step and took 2 h 15 min. Compared to other tools, SVJedi was the fastest among the tested ones (Table 2). Among the long read genotypers, SVJedi was 7 times faster than Sniffles–Ivcf and 50 times faster than svviz2. The large runtime of svviz2 (more than 5 days) can be explained by the fact that it is not natively parallelized, when manually parallelized on 20 CPU (only 20 due to memory limits), it took roughly 11 h.

## 5 Discussion and conclusion

In conclusion, we provide a novel SV genotyping approach for long read data that showed good results on simulated and real datasets. The approach is implemented in the SVJedi software for most SV types (insertions, deletions, inversions and translocations). The robustness of our tool, SVJedi, was highlighted in this work, for several sequencing depths and error rates, and also related to the precision of the breakpoint positions. On a real human dataset with more than 12 000 insertions and deletions, SVJedi obtained a better genotyping accuracy than other tested genotyping and discovery tools. SVJedi, like the other tools, had more difficulties to accurately genotype small variants (<100 bp) and those located in large tandem repeat regions. However, SVJedi showed a more conservative behavior than other tools, with a lower genotyping rate for these most difficult SVs: instead of estimating an incorrect genotype, it favored not assigning any genotype at all.

This work also demonstrated that this is crucial to develop dedicated SV genotyping methods, as well as SV discovery methods. Firstly, because this is the only way to get evidence for the absence of SVs in a given individual. Secondly, and more surprisingly, because SV discovery tools are not as efficient and precise to genotype variants once they have been discovered, at least with long read data as was shown here. Indeed, without a priori SV discovery is a much harder task than genotyping. Because the alternative allele is not known in discovery, discovery methods rely on fewer or noisier signals to identify the SVs than genotyping methods. Consequently, both approaches would likely benefit from different optimal parameter settings, with for instance discovery methods requiring a more stringent set of parameters to limit the false discovery rate. However, in this paper, we have no intention to oppose both approaches but rather argue that they are complementary and are intended to different purposes: when the aim is strictly to genotype or compare individuals on a set of already known variants, we have shown that using as much as possible the known features of variants is much more efficient.

Also, on real human data, we were able to quantify the impact of the sequencing technology on SV genotyping. Although this was expected that long read data would perform better than short read ones, the observed difference is considerable with a 2-fold increase of the genotyping accuracy with long reads. This is in particular due to the very poor performances obtained with short reads that are ill-adapted to deal with the complex and repeat-rich regions often present at SV junctions. On the opposite, this work shows that the long-distance information contained in long reads can be efficiently used to discriminate between breakpoints, despite relatively high sequencing error rates and variability in sequencing coverage. This result underlines the relevance of such a method dedicated to genotyping from long read data.

Although long read sequencing technology remains to date more expensive than short read ones, to be used for instance in routine in the clinical setting (Merker *et al.*, 2018), we can hope that this situation will improve in the next few years. The high genotyping accuracy and low computational requirements of SVJedi make it ready for such happening and to be integrated into routine pipelines to screen for instance disease-related SVs and therefore improve medical diagnosis or disease understanding.

## Acknowledgements

We are thankful to the Genouest bioinformatics platform, computations have been made possible thanks to the resources of the Genouest infrastructure. We also thank the reviewers for their insightful and constructive feedback that helped to improve this manuscript.
*Financial Support*: none declared.

*Conflict of Interest*: none declared.

## References

Alkan,C. *et al.* (2011) Genome structural variation discovery and genotyping. *Nat. Rev. Genet.*, **12**, 363–376.
Antaki,D. *et al.* (2018) SV2: accurate structural variation genotyping and de novo mutation detection from whole genomes. *Bioinformatics*, **34**, 1774–1777.
Audano,P.A. *et al.* (2019) Characterizing the major structural variant alleles of the human genome. *Cell*, **176**, 663–675.
Bailey,J.A. *et al.* (2002) Recent segmental duplications in the human genome. *Science*, **297**, 1003–1007.
Chander,V. *et al.* (2019) Evaluation of computational genotyping of structural variation for clinical diagnoses. *GigaScience*, **8**, giz110
Chiang,C. *et al.* (2015) SpeedSeq: ultra-fast personal genome analysis and interpretation. *Nat. Methods*, **12**, 966–968.
De Coster,W. *et al.* (2019) Structural variants identified by Oxford Nanopore PromethION sequencing of the human genome. *Genome Res.*, **29**, 1178–1187.
Heller,D. and Vingron,M. (2019) SVIM: structural variant identification using mapped long reads. *Bioinformatics*, **35**, 2907–2915.
Huddleston,J. *et al.* (2017) Discovery and genotyping of structural variation from long-read haploid genome sequence data. *Genome Res.*, **27**, 677–685.
Jain,M. *et al.* (2018) Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.*, **36**, 338–345.
Kidd,J.M. *et al.* (2010) A human genome structural variation sequencing resource reveals insights into mutational mechanisms. *Cell*, **143**, 837–847.

Downloaded from https://academic.oup.com/bioinformatics/article/36/17/4568/5841661 by INRIA Rennes user on 18 December 2020

Li,H. (2011) A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, **27**, 2987–2993.
Li,H. (2018) Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, **34**, 3094–3100.
Lupski,J.R. (2015) Structural variation mutagenesis of the human genome: impact on disease and evolution. *Environ. Mol. Mutagen.*, **56**, 419–436.
Merker,J.D. *et al.* (2018) Long-read genome sequencing identifies causal structural variation in a Mendelian disease. *Genet. Med.*, **20**, 159–163.
Nielsen,R. *et al.* (2011) Genotype and SNP calling from next-generation sequencing data. *Nat. Rev. Genet.*, **12**, 443–451.
Norris,A.L. *et al.* (2016) Nanopore sequencing detects structural variants in cancer. *Cancer Biol. Ther.*, **17**, 246–253.
Phan,L. *et al.* (2016) dbVar structural variant cluster set for data analysis and variant comparison. *F1000Research*, **5**, 673.
Sedlazeck,F.J. *et al.* (2018) Accurate detection of complex structural variations using single-molecule sequencing. *Nat. Methods*, **15**, 461–468.
Spies,N. *et al.* (2015) svviz: a read viewer for validating structural variants. *Bioinformatics*, **31**, 3994–3996.
Stancu,M.C. *et al.* (2017) Mapping and phasing of structural variation in patient genomes using nanopore sequencing. *Nat. Commun.*, **8**, 1326.
Stöcker,B.K. *et al.* (2016) SimLoRD: simulation of long read data. *Bioinformatics*, **32**, 2704–2706.
Zook,J.M. *et al.* (2016) Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci. Data*, **3**, 160025.

Downloaded from https://academic.oup.com/bioinformatics/article/36/17/4568/5841661 by INRIA Rennes user on 18 December 2020

# Curriculum Vitæ

**Claire Lemaitre**
Docteur en bioinformatique

## Etat civil

Claire Lemaitre, née le 7 novembre 1982 à Grenoble (38), vie maritale, 2 enfants, nationalité française.

Coordonnées professionnelles
Equipe Genscale
Inria Rennes Bretagne Atlantique
Campus de Beaulieu
35042 Rennes cedex
tel : 02 99 84 71 16
claire.lemaitre@inria.fr

Page personnelle : http://people.rennes.inria.fr/Claire.Lemaitre/

## Situation actuelle

| | | |
|---|---|---|
| Depuis oct. 2010 | **Chargée de recherche**<br>Equipe Genscale | Inria Rennes Bretagne Atlantique, et UMR 6074 IRISA |

## Études et diplômes

| | | |
|---|---|---|
| 2009 | **Qualification aux fonctions de maître de conférence** en sections **27** et **65**. | |
| 2005–2008 | **Doctorat de bioinformatique**<br>Titre : *Réarrangements chromosomiques dans les génomes de mammifères : caractérisation des points de cassure.*<br>Diplôme décerné le 6/11/2008. Composition du jury :<br>- Marie-France Sagot, Directrice de Recherche INRIA, directrice de thèse;<br>- Christian Gautier, Professeur à l'université Lyon I, co-directeur de thèse;<br>- Eric Rivals, Directeur de Recherche CNRS, rapporteur;<br>- Nicolas Galtier, Directeur de Recherche CNRS, rapporteur;<br>- Roderic Guigó, Professeur au CRG, rapporteur;<br>- Pierre Capy, Professeur à l'université Paris VI, examinateur;<br>- Hubert Pinon, Professeur à l'université Lyon I, président. | Université Claude Bernard (Lyon I) |
| 2004–2005 | **Master Recherche de bioinformatique**<br>mention Très Bien<br>Stage de master : *Analyse informatique des régions de cassure dans les génomes de mammifères*, sous la direction de Marie-France Sagot (DR INRIA, Lyon). | Université Claude Bernard (Lyon I) |
| 2000–2005 | **Diplôme d'ingénieur de l'INSA**[1] **de Lyon**<br>Département de spécialité : *Bioinformatique et Modélisation*. | INSA de Lyon |
| 2000 | **Baccalauréat, Série S**<br>Obtenu avec mention Très Bien. | Lycée Champollion, Grenoble |

[1] Institut National des Sciences Appliquées

## Expérience professionnelle

| | |
|---|---|
| 2010 - | **Chargée de Recherche** CRCN, Inria,<br>Inria Rennes Bretagne Atlantique, Rennes |
| 2008-2010 | **Post-doctorat**,<br>Centre de Bioinformatique de Bordeaux |
| 2005-2008 | **Allocataire de recherche MENRT**,<br>Laboratoire de Biométrie et Biologie Evolutive (UMR CNRS 5558), Lyon |
| 2005-2008 | **Monitrice du CIES**,<br>Département de mathématiques du Premier Cycle de l'INSA de Lyon |
| Jan-Juin 2005 | **Stage de Master recherche**,<br>Laboratoire de Biométrie et Biologie Evolutive (UMR CNRS 5558), Lyon |

## Activités d'enseignement

*Le détail des enseignements est présenté page 3.*

- **50h** par an d'enseignement, au niveau Master, en algorithmique du texte et bioinformatique.
- Responsabilité d'1 UE en Master.

## Encadrement

*Le détail des encadrements est présenté page 4.*

- En cours : 1 post-doctorant, 1 ingénieure, 1 stagiaire.
- Passés : 4 doctorants, 1 ingénieur, 3 post-doctorants, 13 stagiaires de master entre 2011 et 2020.

## Publications & communications

*Le détail des publications et communications est présenté page 9.*

- **28 articles** publiés dans des revues internationales, dont 5 en premier auteur et 9 en dernier auteur.
- 2 chapitres de livre.
- 28 communications et séminaires invités

## Développement de logiciels

*Le détail des logiciels est présenté page 15.*

**13 logiciels** développés et distribués en open source, dont 4 majeurs (C++, $> 5000$ lignes de code), maintenus sur la durée: MindTheGap, Simka, discoSnp++, GATB.

## Suite du CV

*Activités d'enseignement* ..... *page 3*
*Encadrements de stagiaires, doctorants, post-doctorants* ..... *page 4*
*Responsabilités et tâches collectives* ..... *page 7*
*Liste des publications* ..... *page 9*
*Liste des logiciels* ..... *page 15*

## Activités d'enseignement

### Actuellement

2 demies UEs d'algorithmique des séquences (pattern matching, structures d'indexation, alignement de séquences, assemblage de séquences), pour des publics différents :

- niveau M1, master d'informatique de l'université Rennes 1 (ISTIC) : UE "Bioinformatique des séquences" (BIF)
  - **25 heures** par an depuis 2012.
  - $\sim$ 15 étudiants.
  - Conception de l'UE (contenu et supports), avec Pierre Peterlongo, en 2012.
  - **Responsabilité** de l'UE depuis 2018.
- niveau M2, master de bioinformatique de l'université Rennes 1 (UFR SVE) : UE "Algorithmique des séquences" (ALG)
  - **25 heures** par an depuis 2017.
  - $\sim$ 35 étudiants.
  - Conception de l'UE (contenu et supports), avec Pierre Peterlongo, en 2017.

### Enseignements passés

- Algorithmique des séquences, niveau M1, parcours informatique, ENS de Rennes (**12 heures** par an), 2017.
- TPs de biostatistiques sous R, niveau L3, licence de biologie de l'université de Rennes 1 (**12 heures** par an), 2013 à 2017.
- Modélisation des systèmes dynamiques, niveau M2, master de bioinformatique de l'université de Rennes 1 (**20 heures** par an), 2011 à 2017. Co-responsabilité de l'UE de 2013 à 2017.
- **192 heures** dans le cadre du **monitorat**, de mathématiques, algorithmique et bioinformatique, enseignées au *Premier Cycle* et dans le département *Bioinformatique et Modélisation* de l'INSA de Lyon sur la période 2005-2008.
- Organisation et animation de **formations continues** de bioinformatique, pour un public de biologistes, au Centre de Bioinformatique de Bordeaux en 2009 et 2010 (enseignement : **20 heures**).

## Encadrements

Total : 4 doctorants, 2 ingénieures, 4 post-doctorants, 14 stagiaires de master.

- **Encadrements en cours :**
  - Pierre Morisse, **post-doctorant**, 2 ans (2020-2022) : détection de variants de structure avec des données linked-reads et application à des génomes d'organismes non modèles.
    Encadrement à 100 %.
    Production : 2 publications soumises (S2, S3), 1 logiciel (L8 : LRez).
  - Anne Guichard, **ingénieure**, 2 ans (2019-2021) : assemblage de génomes complexes avec des données de séquençage de 3ème génération.
    Co-encadrement avec Fabrice Legeai, taux réel 50 %.
    Production : 2 posters, 1 publication en cours d'écriture, 1 logiciel (L7 : MTG-link).
  - Sandra Romain, **stagiaire** de niveau M2, 6 mois (janvier-juillet 2021) : utilisation de graphes de génomes pour le génotypage de variants complexes.
- **Encadrements passés :**
  - Wesley Delage, **doctorant**, 3 ans (2017-2020, thèse soutenue et obtenue le 11/12/2020) : caractérisation et détection d'insertions constitutionnelles de grande taille dans le cadre d'un usage médical.
    Directrice officielle (obtention d'une dérogation d'HDR de l'ED MathSTIC).
    Co-encadrement avec Julien Thévenon, taux officiel 50%, taux réel 90 %.
    Production : 1 publication (P1: *Bmc Genomics*), amélioration d'un logiciel (L1: MindTheGap).
    Devenir : post-doctorant à l'INSERM à Nantes.
  - Lolita Lecompte, **doctorante**, 3 ans (2017-2020, thèse soutenue et obtenue le 04/12/2020) : génotypage de variants de structure avec des données de lectures longues.
    Co-encadrement avec Dominique Lavenier, taux officiel 50%, taux réel 90 %.
    Production : 1 publication (P2: *Bioinformatics*), 1 logiciel (L5: SVJedi).
    Devenir : Ingénieure de Recherche (CDD) à l'Institut Curie à Paris.
  - Jérémy Gauthier, **post-doctorant**, 18 mois (2017-2018) : Analyse et comparaison de méthodes de génomique de populations de papillons.
    Encadrement à 100 %.
    Production : 2 publications (P4: *PeerJ*, P5: *Molecular Ecology*), 1 logiciel (L3: DiscoSnpRAD).
    Devenir : post-doctorant au Museum d'Histoire Naturelle de Genève en Suisse.
  - Cervin Guyomar, **doctorant**, 3 ans (2015-2018, thèse soutenue et obtenue le 07/12/2018) : développement de méthodes pour la métagénomique de communautés symbiotiques.
    Co-encadrement avec Jean-Christophe Simon et Christophe Mougel (INRA Rennes), taux officiel 40%, taux réel 70 %.
    Production : 2 publications (P8: *Microbiome*, P3: *NAR Genomics and Bioinformatics*), 1 logiciel (L6: MinYS).
    Devenir : ingénieur de recherche titulaire à l'INRAE, Toulouse, depuis 2020.
  - Gaëtan Benoit, **doctorant**, 3 ans (2014-2017, thèse soutenue et obtenue le 29 novembre 2017) : développement de méthodes pour la métagénomique comparative.
    Co-encadrement avec Dominique Lavenier, taux officiel 50%, taux réel 90 %.
    Production: 2 publications (P13: *PeerJ Comp Sci*, P7: *Bioinformatics*), 1 logiciel (L2: Simka et simkaMin).
    Devenir : post-doctorant à Earlham Institute (Angleterre), après 3 ans d'auto-entreprenariat dans le jeu vidéo.

- Anaïs Gouin, **ingénieure**, 3 ans (2012-2015) : gestion et analyse des données de séquençage d'insectes ravageurs de culture.
  Co-encadrement avec Fabrice Legeai, taux réel 70 %.
  Production : 2 publications (P18: *Heredity*, P12: *Scientific Reports*).
  Devenir : ingénieure dans le privé, Aix en Provence.
- Liviu Ciortuz, **post-doctorant**, 1 an (2012-2013) : développement de nouveaux algorithmes pour la détection de variants complexes (variants de structure) dans des données NGS non assemblées.
  Co-encadrement avec Pierre Peterlongo, taux réel 70 %.
  Production: 1 publication (C1: *Conférence AlCoB*), 1 logiciel (L11: TakeABreak).
  Devenir: Maître de Conférence à l'université de Iasi en Roumanie.
- Erwann Scaon, **doctorant**, 1 an (2012-2013) : Algorithmes pour l'assemblage de novo de génomes fortement redondants.
  Arrêt de la thèse à la fin de la 1ère année pour des raisons personnelles.
  Co-encadrement avec Dominique Lavenier, taux réel 90 %.
  Devenir: CDD ingénieur bioinformaticien en France.
- Thomas Derrien, **post-doctorant**, 1 an (2011-2012) : détection de variants structuraux dans les génomes re-séquencés du puceron du pois.
  Encadrement à 100 %.
  Devenir: CR CNRS titulaire à l'Institut de Génétique de Rennes (IGDR), depuis 2012.

- **Encadrements de stagiaires :**
  - Arthur Le Bars, stage de 6 mois niveau M2 (2019) : assemblage de génomes avec des données *linked-reads*.
  - Wesley Delage, stage de 6 mois niveau M2 (2017) : reconstruction de génomes par assemblage guidé dans des données métagénomiques.
  - Charlotte Mouden, stage de 6 mois niveau M2 (2017) : Adaptation et application de l'outil discoSnp à des données de RAD-seq (logiciel L3: DiscoSnpRAD).
  - Mael Kerbiriou, stage de 6 mois niveau M2 (2017) : Amélioration du logiciel discoSnp. (taux 30%)
  - Mathieu Perrotin, stage de 6 mois niveau M2 (2016) : Développement d'une méthode de requêtage dans des fichiers de séquençages métagénomiques. (taux 30 %)
  - Pierre Marijon, stage de 4 mois niveau M1 (2015) : amélioration d'un logiciel existant (L1: MindTheGap).
  - Gaëtan Benoit, stage de 6 mois niveau M2 (2014) : développement d'une méthode de compression des fichiers de séquençage génomique.
    Production: 1 publication (P16: *BMC Bioinformatics*, 1 logiciel (L9: Leon), .
  - Julien Erabit, stage de 6 mois niveau M2 (2013) : développement d'un environnement de benchmark d'assemblage de génomes polyploïdes.
  - Gaëtan Benoit, stage de 3 mois niveau M1 (2013) : développement d'une méthode de correction des reads.
    Production: un logiciel (L10: Bloocoo).
  - Bastien Hervé, stage de 5 mois niveau L2 (2013) : évaluation d'une méthode de barcoding à partir de données NGS plein-génome non assemblées.
  - Elise Larsonneur, stage de 6 mois niveau M2 (2012) : intégration de données "omics", application sur le puceron du pois.
  - Quentin Oliveau, stage de 3 mois niveau M1 (2012) : détection d'insertions virales et de points de jonction dans des données de séquences à haut débit.

– Olivier Rué, stage de 6 mois niveau M2 (2011) : détection de variants (SNP et SV) dans plusieurs génotypes du puceron du pois re-séquencés par NGS.

## Responsabilités, tâches collectives

- **Contrats de recherche :**
  - Responsable de tâches et de partenaire de projets ANRs :
    - Divalps (2021-2025), appel générique "Terre vivante", crédits alloués de 590 K€, porté par Laurence Desprès (CNRS LECA, Grenoble). Financement et encadrement d'une thèse à venir (2021-2024).
    - Supergene (2020-2024), appel générique "Terre vivante", crédits alloués de 650 K€, porté par Mathieu Joron (CNRS CEFE, Montpellier). Financement et encadrement d'un post-doctorant (Pierre Morisse, 2 ans).
    - SpecRep (2015-2019), appel générique "Terre vivante", crédits alloués de 490 K€, porté par Marianne Elias (MNHN, Paris). Financement et encadrement d'un post-doctorant (Jérémy Gautier, 2 ans).
    - Ada-Spodo (2012-2015), appel générique "Biodiversité, évolution, écologie, agronomie"; crédits alloués de 430 K€, porté par Emmanuelle d'Alençon (INRA, Montpellier). Financement et encadrement d'une ingénieure (Anaïs Gouin, 18 mois).
  - Participation à d'autres projets ANRs (sans responsabilité de partenariat) :
    - Hydrogen (2014-2019), porté par Dominique Lavenier, puis Pierre Peterlongo (Inria, Rennes). Financement et encadrement d'une thèse (Gaëtan Benoit, 3 ans).
    - Colib'read (2013-2016), porté par Pierre Peterlongo (Inria, Rennes).
    - SpeciAphid (2011-2014), porté par Jean-Christophe Simon (INRA, Rennes). Financement et co-encadrement d'une ingénieure (Anaïs Gouin, 18 mois).
  - Responsable du projet "Mirage" de la Région Bretagne, Stratégie Attractivité Durable, durée : 1 an (2012-2013). Financement et encadrement d'un post-doctorant (Liviu Ciortuz, 1 an).
  - Responsable du projet PEPS Bio-Math-Info, "barcoding de nouvelle génération", durée : 2 ans (2012-2014)
- **Relectures :**
  - Relecture d'articles pour des journaux (Bioinformatics, Nature Reviews Genetics, Nucleic Acids Research, BMC Evolutionary Biology, PLoS One, IEEE/ACM Transactions on Computational Biology and Bioinformatics), et des conférences internationales de bioinformatique (RECOMB, WABI, ISMB).
  - Membre du Comité de Programme des conférences nationales JOBIM (2012, 2013, 2017, 2019, 2021), SeqBio (2011, 2013, 2014), seqBIM (2019, 2020).
  - Relecture d'1 projet ANR, programme blanc SVSE6 en 2012.
- **Organisation de conférences :**
  - Co-présidente du Comité de Programme de la conférence JOBIM 2022. Cette manifestation est le rendez-vous annuel de la communauté bioinformatique francophone ($\sim$500 participants, 6 conférenciers invités, $\sim$ 250 soumissions (dont 50 proceedings, 200 posters)).
  - Organisation (co-responsable des comités d'organisation et de programme) des journées annuelles du Groupe de Travail seqBIM en 2019 et 2020 (2 jours, 60-100 participants, 2 orateurs invités et 15 exposés sélectionnés, chaque année).
  - Membre du comité d'organisation de l'école Jeunes Chercheurs de Bioinformatique Moléculaire ($(JC)^2BIM$), 2018 et 2020-2021 (5 jours, 30 élèves par session).
  - Organisation de journées de formation à la librairie GATB, 2018 et 2019 (1 jour, 10-15 participants par session).

- Co-présidente du Comité d'Organisation de JOBIM 2012. Cette manifestation est le rendez-vous annuel de la communauté bioinformatique francophone ($\sim$400 participants, budget $>$ 100 000 euros).
- Membre du comité d'organisation du workshop Seq BI 2011 à Rennes.
- Organisation de formations continues de bioinformatique (2009-2010, au sein de la plateforme de bioinformatique de Bordeaux).

- **Participation à des jurys ou comités :**
  - Membre de 5 jurys de thèses en tant que examinatrice (Joseph Lucas, 2016; Arnaud Meng, 2017; Yoann Seeleuthner, 2018, Lyam Baudry, 2019, Guillaume Gautreau, 2020).
  - Membre de comités de recrutement d'un poste de Maître de Conférence (Université de Lyon, 2014) et de 2 postes d'ingénieurs de recherche (INRA et Université de Bordeaux, 2015), d'1 poste d'ATER (Université de Rennes, 2017).
  - Membre de la commission CORDIS de recrutement de doctorants ou post-doctorants du centre Inria Rennes Bretagne Atlantique en 2013, 2014, 2017.
  - Membre de 7 comités de thèses entre 2015 et 2020.
  - Membre de jurys de soutenances de stage (niveau master) : école d'ingénieur ESTBB en 2010 à Bordeaux, master de bioinformatique de l'université Rennes 1 en 2012 et 2017.

- **Animation de communautés :**
  - Responsable du Groupe de Travail seqBIM, commun aux GDRs IM et BIM du CNRS, depuis 2018 (charge partagée avec Gilles Didier et Laurent Bulteau). Animation de la communauté française travaillant sur la combinatoire, l'algorithmique du texte et leurs applications en bioinformatique (environ 150 participants dans 45 équipes recensées en France en 2019). Je suis responsable de la gestion du site internet (http://seqbim.cnrs.fr/) et des outils de communication, du recensement et de la cartographie des équipes et thématiques, et de l'organisation de journées d'animation (au moins un rendez-vous annuel).
  - Membre du comité scientifique du GDR BIM du CNRS (Bioinformatique moléculaire) depuis 2018.
  - Chargée de mission pour l'axe "Environnement" au sein de l'UMR IRISA depuis 2013 : recensement et cartographie des équipes impliquées et animation de l'axe (charge partagée avec Géraldine Pichot puis Nicolas Courty).

- **Autres tâches collectives :**
  - Compilation du rapport annuel d'activité Inria de l'équipe Symbiose, puis Genscale (depuis 2011).

## Publications & communications

Dans mon domaine de recherche, la bioinformatique (et en biologie également), l'ordre des auteurs a de l'importance : les premières et dernières places montrent une contribution plus importante. En particulier, le ou les auteurs en dernières positions sont les personnes qui ont supervisé le travail.

En bioinformatique, les publications se font majoritairement par des soumissions dans des journaux plutôt que des participations à des conférences, notamment pour avoir plus d'impacts auprès des biologistes utilisateurs des méthodes. J'ai surligné ci-dessous en gris souligné les revues majeures de bioinformatique, et en vert, les revues appartenant à des domaines applicatifs (génomique, écologie).

### Revues internationales, avec comité de lecture

28 publications, dont 9 en dernier auteur (ou co-dernier auteur).

**P1** W. Delage, J. Thevenon, **C. Lemaitre**.
*Towards a better understanding of the low recall of insertion variants with short-read based variant callers.*
BMC Genomics, **2020**, 21(1):762.

**P2** L. Lecompte, P. Peterlongo, D. Lavenier, **C. Lemaitre**.
SVJedi: Genotyping structural variations with long reads.
Bioinformatics, **2020**, 36(17):4568–4575.

**P3** C. Guyomar, W. Delage, F. Legeai, C. Mougel, J.C. Simon, **C. Lemaitre**.
MinYS: Mine your symbiont by targeted genome assembly in symbiotic communities.
NAR Genomics and Bioinformatics, **2020**, 2(3):lqaa047.

**P4** J. Gauthier, C. Mouden, T. Suchan, N. Alvarez, N. Arrigo, C. Riou, **C. Lemaitre**, P. Peterlongo.
DiscoSnp-RAD: de novo detection of small variants for population genomics.
PeerJ, **2020**, 8:e9291.

**P5** J. Gauthier, D. de-Silva, Z. Gompert, A. Whibley, C. Houssin, Y. Le Poul, M. McClure, **C. Lemaitre**, F. Legeai, J. Mallet, M. Elias.
Contrasting genomic and phenotypic outcomes of hybridization between pairs of mimetic butterfly taxa across a suture zone.
Molecular Ecology, **2020**, 29(7):1328-1343.

**P6** F. Legeai, B. Santos, S. Robin, ..., **C. Lemaitre**, ... , A.N. Volkoff.
*Genomic architecture of endogenous ichnoviruses reveals distinct evolutionary pathways leading to virus domestication in parasitic wasps.*
BMC Biology, **2020**, 18(1):89.

**P7** G. Benoit, M. Mariadassou, S. Robin, S. Schbath, P. Peterlongo, **C. Lemaitre** .
*SimkaMin: fast and resource frugal de novo comparative metagenomics.*
Bioinformatics, **2019**, 36(4):1275–1276.

**P8** C. Guyomar, F. Legeai, E. Jousselin, C. Mougel C, **C. Lemaitre**, J.C. Simon. (co-dernier auteur)
*Multi-scale characterization of symbiont diversity in the pea aphid complex through metagenomic approaches.*
Microbiome, **2018**, 6:181.

**P9** P. Nouhaud, M. Gautier, A. Gouin, J. Jaquiéry, J. Peccoud, F. Legeai, L. Mieuzet, C. Smadja, **C. Lemaitre**, R. Vitalis, J.C. Simon.
*Identifying genomic hotspots of differentiation and candidate genes involved in the adaptive*

*divergence of pea aphid host races*.
Molecular Ecology, **2018**, 27(16):3287-3300.

**P10** J. Jaquiéry, J. Peccoud, T. Ouisse, F. Legeai, N. Prunier-Leterme, A. Gouin,..., **C. Lemaitre**, ..., J.C. Simon, C. Rispe.
*Disentangling the causes for faster-X evolution in aphids*.
Genome Biology and Evolution, **2018**, 10(2):507-520.

**P11** A. Sczyrba, ... , **C. Lemaitre**, ..., A.C. McHardy (Plus de 20 auteurs)
*Critical Assessment of Metagenome Interpretation – a benchmark of computational metagenomics software*.
Nature Methods, **2017**, 14:1063–1071.

**P12** A. Gouin, ... , **C. Lemaitre**, F. Legeai, E. d'Alençon, P. Fournier (Plus de 20 auteurs, co-dernier auteur).
*Two genomes of highly polyphagous lepidopteran pests (Spodoptera frugiperda, Noctuidae) with different host-plant ranges*.
Scientific Reports, **2017**, 7:11816.

**P13** G. Benoit, P. Peterlongo, M. Mariadassou,E. Drezen, S. Schbath, D. Lavenier, **C. Lemaitre**.
*Multiple comparative metagenomics using multiset k-mer counting*.
PeerJ Computer Science, **2016**, 2:e94.

**P14** Y. Le Bras, O. Collin, C. Monjeaud, V. Lacroix, E. Rivals, **C. Lemaitre**, V. Miele, G. Sacomoto, C. Marchet, B. Cazaux, A. Zine El Aabidine, L. Salmela, S. Alves-Carvalho, A. Andrieux, R. Uricaru, P. Peterlongo.
*Colib'read on galaxy: a tools suite dedicated to biological information extraction from raw NGS reads*.
Gigascience **2016**, 5:9.

**P15** P. Dumas, F. Legeai, **C. Lemaitre**, E. Scaon, M. Orsucci, K. Labadie, S. Gimenez, A.L. Clamens, H. Henri, F. Vavre F, J.M. Aury, P. Fournier, G.J. Kergoat , E. d'Alençon.
*Spodoptera frugiperda (Lepidoptera: Noctuidae) host-plant variants: two host strains or two distinct species?*
Genetica **2015**, 143(3):305-16.

**P16** G. Benoit, **C. Lemaitre**, D. Lavenier, E. Drezen, T. Dayris, R. Uricaru, G. Rizk.
*Reference-free compression of high throughput sequencing data with a probabilistic de Bruijn graph*.
BMC Bioinformatics **2015**, 16:288.

**P17** R. Uricaru, G. Rizk, V. Lacroix, E. Quillery, O. Plantard, R. Chikhi, **C. Lemaitre**, P. Peterlongo.
*Reference-free detection of isolated SNPs*.
Nucleic Acids Research **2015** 43(2):e11.

**P18** A. Gouin, F. Legeai, P. Nouhaud, A. Whibley, J.-C. Simon, **C. Lemaitre**.
*Whole genome re-sequencing of non-model organisms: lessons from unmapped reads*.
Heredity **2015**, 114:494-501.

**P19** G. Rizk, A. Gouin, R. Chikhi, **C. Lemaitre**.
*MindTheGap : integrated detection and assembly of short and long insertions*.
Bioinformatics **2014** (Dec) 30(24):3451-3457.

**P20** E. Drezen, G. Rizk, R. Chikhi, C. Deltel, **C. Lemaitre**, P. Peterlongo, D. Lavenier.
*GATB: Genome Assembly & Analysis Tool Box*.
Bioinformatics **2014**, 30(20):2959-2961.

**P21** N. Maillet, **C. Lemaitre**, R. Chikhi, D. Lavenier, Pierre Peterlongo.
*Compareads: comparing huge metagenomic experiments*.
BMC Bioinformatics, **2012**, 13 (Suppl 19):S10.

**P22** **C. Lemaitre**, A. Barré, C. Citti, F. Tardy, F. Thiaucourt, P. Sirand-Pugnet, P. Thébault.
*A novel substitution matrix fitted to the compositional bias in Mollicutes improves the prediction of homologous relationships*.
BMC Bioinformatics, **2011**, 12:457.

**P23** A. Veron, **C. Lemaitre**, C. Gautier, V. Lacroix, M.-F. Sagot.
*Close 3D proximity of evolutionary breakpoints argues for the notion of spatial synteny*.
BMC Genomics, **2011**, 12:303.

**P24** C. Baudet, **C. Lemaitre**, D. Zanoni, C. Gautier, E. Tannier, M.-F. Sagot.
*Cassis: Precise detection of genomic rearrangement breakpoints.*
Bioinformatics, **2010**, 26(15):1897–1898.

**P25** **C. Lemaitre**, L. Zaghloul, M.-F. Sagot, C. Gautier, A. Arnéodo, E. Tannier, B. Audit.
*Analysis of fine-scale mammalian evolutionary breakpoints provides new insight into their relations to genome organisation and open chromatin*.
BMC Genomics, **2009**, 10:335.

**P26** **C. Lemaitre**, M. Braga Dias Vieira, C. Gautier, M.-F. Sagot, E. Tannier, G.A.B. Marais.
*Footprints of inversions at present and past pseudoautosomal boundaries in human sex chromosomes*.
Genome Biology and Evolution, **2009**, 1(1):56–66.

**P27** **C. Lemaitre**, E. Tannier, C. Gautier, M.-F. Sagot.
*Precise detection of rearrangement breakpoints in mammalian genomes*.
BMC Bioinformatics, **2008**, 9(1):286.

**P28** **C. Lemaitre**, M.-F. Sagot.
*A Small Trip in the Untranquil World of Genomes : A survey on the detection and analysis of genome rearrangement breakpoints*.
Theoretical Computer Science, **2008**, 395(2-3):171–192.

### Chapitres de livre

⋆ C. Guyomar, **C. Lemaitre**.
*Métagénomique et métatranscriptomique.*
Dans *Du texte aux graphes : méthodes et structures discrètes pour la bioinformatique*, ISTE Science Publishing, à paraître **2021**.

⋆ G. Benoit, **C. Lemaitre**, G. rizk, E. Drezen, D. Lavenier.
*de-novo NGS Data Compression.*
Dans *Algorithms for Next-Generation Sequencing Data: Techniques, Approaches, and Applications*, M. Eloumi (editor), Springer, Juillet **2017**.

### Communications à des conférences internationales avec comité de lecture avec actes

**C1** **C. Lemaitre**, L. Ciortuz, P. Peterlongo.
*Mapping-Free and Assembly-Free Discovery of Inversion Breakpoints from Raw NGS Reads*.
AlCoB **2014**, July 2014, Tarragona, Spain.
Publié dans Algorithms for Computational Biology, LNCS **2014**, vol. 8542, pp. 119–130.

## Communications à des conférences internationales avec comité de lecture (sans actes)

**C2** L. Lecompte, P. Peterlongo, D. Lavenier, **C. Lemaitre**.
*SVJedi : Structural variation genotyping using long reads.*
HiTSeq 2019 (High Throughput Sequencing, Algorithms and applications), Basel, Suisse, Juillet **2019**.

**C3** C. Guyomar, F. Legeai, C. Mougel, **C. Lemaitre**, Jean-Christophe Simon.
*Multi-scale characterization of symbiont diversity in the pea aphid complex through metagenomic approaches.*
International Conference on Holobionts, Paris, France, 19-21 avril **2017**.

**C4** G. Benoit G, P. Peterlongo, M. Mariadassou,E. Drezen, S. Schbath, D. Lavenier, **C. Lemaitre**.
*Simka: fast kmer-based method for estimating the similarity between numerous metagenomic datasets*
RCAM 2015 (Recent Computational Advances in Metagenomics), Paris, France, October **2015**.

**C5** P. Sirand-Pugnet, M. Breton, E. Dorset-Frisoni, E. Baranowski, A. Barré, C. Couture, V. Dupuy, P. Gaurivaud, D. Jacob, **C. Lemaitre**, L. Manso-Silvan, M. Nikolski, LX. Nouvel, F. Poumarat, F. Tardy, P. Thébault, S. Theil, C. Citti, F. Thiaucourt, A. Blanchard.
*Evaluation of the relative importance between gene loss and gene gain during mollicute evolution*
*19th Congress of the International Organization for Mycoplasmology (IOM)*, Toulouse , 15-20 juillet **2012**.

**C6** A. Barré, P. Thébault, **C. Lemaitre**, M. Nikolski, A. Blanchard, P. Sirand-Pugnet.
*MolliGen 3.0, a database dedicated to the comparative genomics of mollicutes*
*19th Congress of the International Organization for Mycoplasmology (IOM)*, Toulouse , 15-20 juillet **2012**.

**C7** A. Barré, **C. Lemaitre**, P. Thébault, A. de Daruvar, A. Blanchard, P. Sirand-Pugnet.
*Molligen 3.0, evolution of a database dedicated to the comparative genomics of mollicutes.*
*18th Congress of the International Organization for Mycoplasmology (IOM)*, Chianciano Terme (ITA), 11-16 juillet **2010**.

**C8** A. Veron, **C. Lemaitre**, C. Gautier, V. Lacroix, M.-F. Sagot.
*Close 3D proximity of evolutionary breakpoints argues for the notion of spatial synteny.*
*Integrative Post Genomics (IPG) 2010*, Lyon, 25-26 novembre **2010**.

**C9** **C. Lemaitre**, E. Tannier, C. Gautier, M.-F. Sagot.
*Precise detection and analysis of rearrangement breakpoints in mammalian genomes.*
*13th Evolutionary Biology Meeting at Marseilles (EBM)*, Marseille, 22-25 septembre **2009**.

**C10** **C. Lemaitre**, M. Braga Dias Vieira, C. Gautier, M.-F. Sagot, E. Tannier, G.A.B. Marais.
*Footprints of inversions at present and past pseudoautosomal boundaries in human sex chromosomes.*
*Integrative Post Genomics (IPG) 2008*, Lyon, 19-20 novembre **2008**.

**C11** L. Zaghloul, **C. Lemaitre**, M.-F. Sagot, C. Gautier, A. Arnéodo, E. Tannier, B. Audit.
*Analysis of fine-scale mammalian evolutionary breakpoints provides new insight into their relations to genome organisation and open chromatin.*
*Integrative Post Genomics (IPG) 2008*, Lyon, 19-20 novembre **2008**.

## Communications à des conférences nationales avec comité de lecture

**C12** W. Delage, J. Thevenon, **C. Lemaitre**.
*Towards a better understanding of the low discovery rate of short-read based insertion variant callers.*
JOBIM 2020, Montpellier, juillet **2020** (8 pages).

**C13** C. Guyomar, W. Delage, F. Legeai, C. Mougel, Jean-Christophe Simon, **C. Lemaitre**.
*Reference guided genome assembly in metagenomic samples.*
JOBIM 2019, Nantes, juillet **2019** (8 pages).

**C14** L. Lecompte, P. Peterlongo, D. Lavenier, **C. Lemaitre**.
*Genotyping Structural Variations using Long Read Data.*
JOBIM 2019, Nantes, juillet **2019** (8 pages).

**C15** G. Benoit, P. Peterlongo, M. Mariadassou,E. Drezen, S. Schbath, D. Lavenier, **C. Lemaitre**.
*Multiple comparative metagenomics using multiset k-mer counting*.
JOBIM 2017, Lille, 3-6 juillet **2017**.

**C16** A. Gouin, F. Legeai, P. Nouhaud, G. Rizk, J.-C. Simon **C. Lemaitre**.
*Whole genome re-sequencing : lessons from unmapped reads.*
JOBIM 2013, Toulouse, 1-4 juillet **2013**.

Note : JOBIM (Journées Ouvertes en Bioinformatique) est LA conférence nationale de bioinformatique, organisée notamment par la société savante de bioinformatique (SFBI).

### Séminaires invités

- ⋆ *Looking for genomic variants in the De Bruijn Graph.*
  Institute for Advanced Biosciences, Université de Grenoble Alpes, Grenoble, France, Dec. **2017**.
- ⋆ *Comparaison (massive) de (nombreux) metagénomes. Passons par les kmers pour passer à l'échelle.*
  Journée scientifique sur "le Microbiome" organisée par Biogenouest, Rennes, décembre **2016**.
- ⋆ *Comparing numerous metagenomics datasets.* Laboratoire de Biométrie et Biologie Evolutive, Lyon, novembre **2016**.
- ⋆ *Reference-free detection of genomic variants: from SNPs to inversions.*
  Workshop de restitution du projet ANR Colib'read, Institut Curie, Paris, novembre **2016**.
  Workshop ABS4NGS, Institut Curie, Paris, juin **2015**.
- ⋆ *Chromosomal evolution and genome organisation : a complex relationship.*
  Conférence invitée au Center for Genome Regulation, Santiago, Chile, avril **2012**.
  Conférence invitée au LINA, Nantes, décembre **2012**.
- ⋆ *Réarrangements chromosomiques dans les génomes de mammifères : caractérisation des points de cassure.*
  Conférence invitée Inria Lille, équipe Bonsai, Lille, février **2012**.
  Conférence invitée à l'IRISA équipe Symbiose, Rennes, novembre **2009**.
  Conférence invitée au Laboratoire Bordelais de Recherche en Informatique (LABRI), Bordeaux, janvier **2009**.
- ⋆ *Détection et analyse des points de cassure de réarrangements dans les génomes de mammifères.*
  Conférence invitée au Centre de BioInformatique de Bordeaux, Bordeaux, avril **2008**.
- ⋆ *A method to detect precisely rearrangement breakpoints in mammalian genomes*.
  Workshop *Dynamics of genomes*, Valparaiso, Chili, mars **2008**.

## Manuscrit de thèse

⋆ **C. Lemaitre**
*Réarrangements chromosomiques dans les génomes de mammifères : caractérisation des points de cassure.*
Université Claude Bernard Lyon 1, **2008**.
https://tel.archives-ouvertes.fr/tel-00364265/

## Pédagogie

⋆ E. Billoir, **C. Lemaitre**, S. Charles.
*La modélisation des réseaux de gènes : une situation-problème pour l'apprentissage de méthodes mathématiques avancées.*
Poster au *25e Congrès de l'Association Internationale de Pédagogie Universitaire (AIPU)*, Montpellier, Mai **2008**.
Article dans les actes de la conférence.

## Médiation scientifique

⋆ S. Alizon, F. Cazals, S. Guindon, **C. Lemaitre**, T. Mary-Huard, A. Niarakis, M. Salson, C. Scornavacca, H. Touzet.
*SARS-CoV-2 Through the Lens of ComputationalBiology: How bioinformatics is playing a key role in the study of the virus and its origins.*
Mars 2021, déposé sur HAL (https://hal-cnrs.archives-ouvertes.fr/hal-03170023).
Rapport de médiation scientifique produit sous l'égide du GDR BIM, à destination de non scientifiques ou scientifiques d'autres disciplines que la bioinformatique.

⋆ **C. Lemaitre**
*Le génome humain, la bioinformatique et le métier de bioinformaticien(ne).*
Dans le cadre du dispositif "La découverte de la recherche".
2 Interventions pour des élèves de 1ère du Lycée Bertrand d'Argentré à Vitré, mai 2013.

## Articles soumis en 2021

**S1** D.J. Richter, ... , **C. Lemaitre**, ... , D. Iudicone, O. Jaillon (Plus de 20 auteurs).
*Genomic evidence for global ocean plankton biogeography shaped by large-scale current systems.*
En cours de review à eLife. BioRxiv 2020 https://www.biorxiv.org/content/10.1101/867739v2.

**S2** P. Morisse, F. Legeai, **C. Lemaitre**.
*LEVIATHAN: efficient discovery of large structural variants by leveraging long-range information from Linked-Reads data.*
BioRxiv 2021 https://www.biorxiv.org/content/10.1101/2021.03.25.437002v1.

**S3** P. Morisse, **C. Lemaitre**, F. Legeai.
*LRez: C++ API and toolkit for analyzing and managing Linked-Reads data.*
aRxiv 2021 https://arxiv.org/abs/2103.14419.

## Logiciels

En bioinformatique, les logiciels sont un élément très important pour la visibilité de la recherche auprès des biologistes.

Tous ces logiciels sont distribués sous licence GNU GPL ou Affero GPL en Open Source, la majorité d'entre eux sont déposés à l'Agence de Protection des Programmes (APP).

- **Logiciels majeurs** (C++, $> 5000$ lignes de code), maintenus sur la durée et qui évoluent encore :

  **L1** MindTheGap (2014 - ...) : logiciel d'assemblage local pour la détection de variants génomiques de type insertions ou l'assemblage de génomes bactériens.
  https://github.com/GATB/MindTheGap
  Ma contribution : j'ai supervisé les premiers développements du logiciel, j'ai implémenté moi-même de nouvelles fonctionalités, j'assure sa diffusion et sa maintenance.

  **L2** Simka et simkaMin (2016 - ...) : logiciels de comparaisons massives de données de séquençages métagénomiques.
  https://github.com/GATB/simka
  Ma contribution : j'ai supervisé le développement du logiciel depuis sa naissance. J'assure sa diffusion et sa maintenance.

  **L3** DiscoSnp et DiscoSnpRAD (2012 - ...) : logiciel de détection de polymorphisme sans génome de référence.
  https://github.com/GATB/DiscoSnp
  Ma contribution : j'ai participé à la supervision du développement, et j'ai implémenté de nouvelles fonctionalités en 2018-2019 (DiscoSnpRAD).

  **L4** Librairie C++ GATB-core (2014 - ...) : librairie C++ pour le traitement et l'analyse de données de séquençage haut débit implémentant des structures de données à faible emprunte mémoire.
  https://github.com/GATB/gatb-core
  Ma contribution : je participe à la maintenance, diffusion et formation aux utilisateurs.

- **Logiciels plus légers et diffusés plus récemment (2019-2021)** :

  **L5** SVJedi (2019) : logiciel de génotypage de Variants Structuraux avec des données de séquençage dernière génération, codé en python.
  https://github.com/llecompte/SVJedi

  **L6** MinYS (2019) : logiciel d'assemblage de génomes bactériens dans des données métagénomiques, codé en python.
  https://github.com/cguyomar/MinYS

  **L7** MTG-link (2020) : logiciel de gap-filling dédié aux données de séquençage linked-reads, codé en python.
  https://github.com/anne-gcd/MTG-Link

  **L8** LRez (2021) : librairie C++ et suite d'outils pour traiter les données linked-reads, codé en C++.
  https://github.com/morispi/LRez

  **L9** DRJBreakpointFinder (2019) : logiciel de détection de site d'excision pour des données de séquençage de virus. Langages : bash, perl, R.
  https://github.com/stephanierobin/DrjBreakpointFinder

- **Logiciels encore distribués, peu maintenus ou sans évolution :**

**L9** Leon (2014) : logiciel de compression des fichiers de séquençage d'ADN, codé en C++
Evolution : intégré en 2018 dans la librairie GATB-core, en cours de valorisation industrielle par l'entreprise Enancio (actuellement Illumina) (créée par un des auteurs du logiciel : Guillaume Rizk).
https://github.com/GATB/leon

**L10** Bloocoo (2013) : logiciel de correction des lectures de séquençage à très faible emprunte mémoire, codé en C++.
https://github.com/GATB/bloocoo

**L11** TakeABreak (2014) : logiciel de détection d'inversions sans génome de référence, codé en C++
Preuve de concept, usage essentiellement interne.
https://github.com/GATB/TakeABreak

**L12** Compareads (2012) : logiciel de comparaison de séquences, codé en C++.
http://alcovna.genouest.org/compareads

**L13** Cassis (2008) : logiciel de génomique comparative (codé en perl et R).
http://pbil.univ-lyon1.fr/software/Cassis