Développement de nouvelles méthodes algorithmiques pour le traitement des UMI à partir des données de séquençage haut débit.

Vincent Sater

Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité INFORMATIQUE

Préparée au sein de l'Université de Rouen Normandie

Développement de nouvelles méthodes algorithmiques pour le traitement des UMI à partir des données de séquençage haut débit.

Présentée et soutenue par
VINCENT SATER

Thèse soutenue le 27/09/2021 devant le jury composé de		
MME THERESE COMMES	PROFESSEUR DES UNIVERSITES, UNIVERSITE MONTPELLIER 1	Rapporteur du jury
M. PIERRE PETERLONGO	CHARGE DE RECHERCHE, IRISA RENNES	Rapporteur du jury
M. JEAN-PHILIPPE JAIS	MAITRE DE CONF UNIV. - PRATICIEN HOSP., UNIVERSITE PARIS 5 UNIVERSITE PARIS DESC	Membre du jury
MME ELISE PRIEUR-GASTON	MAITRE DE CONFERENCES, Université de Rouen Normandie	Membre du jury
MME HELENE TOUZET	DIRECTEUR DE RECHERCHE, CNRS NORD, PAS-DE-CALAIS ET PICARDIE	Membre du jury
M. HUGUES ROEST CROLLIUS	DIRECTEUR DE RECHERCHE, CNRS PARIS	Président du jury
M. THIERRY LECROQ	PROFESSEUR DES UNIVERSITES, Université de Rouen Normandie	Directeur de thèse
M. PHILIPPE RUMINY	CHARGE DE RECHERCHE, Université de Rouen Normandie	Co-directeur de thèse

Thèse dirigée par THIERRY LECROQ et PHILIPPE RUMINY, Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes

UNIVERSITÉ DE ROUEN NORMANDIE

MIIS

litis

Résumé

Les objectifs de cette thèse s'inscrivent dans la large problématique du traitement des données issues de séquenceurs à très haut débit, et plus particulièrement des *reads* courts, issus de séquenceurs de deuxième génération. Les aspects abordés dans cette problématique se concentrent principalement sur le développement de nouvelles méthodologies se basant sur des séquences moléculaires uniques appelées UMI utilisées pour étiqueter les fragments d'ADN initiaux et permettant d'améliorer la précision des résultats obtenus.

Tout d'abord, dans le domaine de la transcriptomique, une nouvelle méthode a été développée afin d'améliorer les résultats de mesure de l'expression génique d'une part, et de détecter les transcrits de fusion dans les tumeurs d'autre part. Cette méthode se base sur une RT-MLPA couplée à un séquenceur NGS. Elle permet d'amplifier les fragments d'ARN présents dans un échantillon tumoral et d'obtenir les séquences des fragments récoltés. L'analyse sous-jacente vise à analyser ces séquences une par une pour, dans un premier temps, attribuer chaque séquence à l'échantillon séquencé, et dans un deuxième temps, retrouver le nom du gène qu'il exprime. Pour cela, RT-MiS a été développé. RT-MiS est un outil permettant d'effectuer la totalité de l'analyse commençant par l'extraction et la correction des UMI des séquences jusqu'à la production des résultats sous forme de matrice d'expression par gène pour chaque échantillon. RT-MiS comporte aussi une interface d'analyse dédiée permettant de lancer l'outil facilement par les chercheurs. Cette interface permet d'automatiser le plus possible le processus d'analyse complet et de produire les résultats sous formes de figures et graphiques interactifs rendant l'interprétation biologique plus facile.

Ensuite, dans le domaine de la génomique, un nouvel outil de détection de variants somatiques a été développé. L'outil UMI-VarCal est un *variant caller* basé sur les UMI et donc implémentant une analyse de ces derniers pour appeler efficacement les variants dans les échantillons tumoraux. L'utilité des UMI est mise en évidence par l'amélioration de la précision de détection des variants, surtout quand la fréquence tombe au-dessous de 1%. UMI-VarCal applique un test de Poisson pour filtrer les positions ne présentant pas des variants et puis se sert d'une analyse des UMI et de deux filtres complémentaires pour filtrer les faux positifs. UMI-VarCal a été conçu de façon très optimisée afin d'effectuer son analyse tout en restant plus efficace que les autres outils en termes de détection de variants et de temps d'exécution.

Finalement, et toujours dans le domaine de la détection des variants, un nouveau simulateur de *reads* a été développé. Cet outil appelé UMI-Gen est le premier simulateur de *reads* capable de générer des séquences avec des UMI. De plus, UMI-Gen est capable d'insérer des variants somatiques (SNV) ou des variants structuraux (CNV) dans les fichiers simulés. En outre, en analysant un ensemble de fichiers normaux, il est capable d'estimer le bruit de fond dans ces échantillons pour le reproduire dans les *reads* simulés. Ces fichiers peuvent être utilisés par la suite pour évaluer les *variant callers*, surtout ceux implémentant une analyse UMI dans leur algorithme.

Mots clés : Séquençage à haut débit, UMI, transcriptomique, expression génique, transcrits de fusion, génomique, *variant calling*, simulation de *reads*.

Abstract

The objectives of this thesis fall within the broad issue of processing data from next generation sequencers, and more particularly short reads from second-generation sequencers. The aspects addressed in this issue mainly focus on the development of new methodologies based on unique molecular sequences called UMI used to label the initial DNA fragments and to improve the precision of the results.

First of all, in the field of transcriptomics, a new method has been developed in order to improve the results of measuring gene expression on the one hand, and to detect fusion transcripts in tumors on the other hand. This method is based on an RT-MLPA coupled to an NGS sequencer. It makes it possible to amplify the RNA fragments from a tumor sample and to obtain the sequences of the analyzed fragments. The underlying analysis aims to analyze these sequences one by one in order, first, to assign each sequence to the corresponding sample, and secondly, to find the name of the gene it expresses. For this, RT-MiS has been developed. RT-MiS is a tool that is able to perform the entire analysis starting with the extraction and correction of the UMI from the sequences until the production of the results in the form of an gene expression matrix for each sample. RT-MiS also includes a dedicated analysis interface allowing for the tool to be launched easily by the users. This interface automates the entire analysis process as much as possible and produces the results in the form of interactive figures and graphs making biological interpretation much easier.

Then, in the field of genomics, a new somatic variant detection tool was developed. The UMI-VarCal tool is a UMI-based variant caller that implements a UMI analysis to efficiently call the variants in tumor samples. The utility of using the information from the UMI is highlighted by the improved accuracy of variant detection, especially when the frequency falls below 1%. UMI-VarCal applies a Poisson test to filter out non-variant positions and then applies a UMI analysis and two complementary filters to remove false positives. UMI-VarCal has been designed in a highly optimized manner allowing it to perform its analysis while remaining more efficient than other tools in terms of variant detection and execution time.

Finally, and still in the field of variant detection, a new read simulator was developed. This tool called UMI-Gen is the first read simulator capable of generating sequences with UMI. In addition, UMI-Gen is capable of inserting somatic variants (SNV) or structural variants (CNV) into the simulated files. Furthermore, by analyzing a set of normal files, it is able to estimate the background noise in these samples and reproduce it in the simulated data. These files can be used later to evaluate different variant callers, especially those implementing a UMI analysis in their algorithm.

Keywords : next generation sequencing, UMI, transcriptomics, gene expression, fusion transcripts, genomics, variant calling, read simulation.

Remerciements

Tout d'abord, je tiens à remercier Thierry Lecroq, Élise Prieur-Gaston et Philippe Ruminy de m'avoir accordé leur confiance, et de m'avoir laissé une grande autonomie pour bien mener mes travaux. Leur soutien, leur disponibilité et leurs nombreux conseils m'ont été extrêmement précieux tout au long de cette thèse.

J'adresse également mes remerciements aux membres du jury, Thérèse Commes, Pierre Peterlongo, Hélène Touzet, Jean-Philipe Jais et Hugues Roest Crollius d'avoir accepté d'examiner mon travail. Je remercie aussi le jury du comité de suivi individuel, Bruno Tesson et Abdelaziz Bensrhair, d'avoir suivi mon travail et d'avoir veillé au bon déroulement de ma thèse.

Un grand merci également à l'équipe TIBS, qui m'a accueilli pendant ces trois ans, pour la sympathie qu'ils m'ont témoignée. Laurent, Nicolas, Hélène, Saïd, Lina et Richard : merci pour votre bonne humeur au quotidien. Un grand merci à Caroline qui m'a fait confiance pour enseigner sous sa supervision pendant trois semestres et qui en a fait une expérience très agréable.

Merci à toute l'équipe de l'Inserm 1245 au Centre Henri Becquerel, Fabrice, Philippe, Pierre-Julien, Mathieu, Victor, Vinciane, Shirley, Élodie et "Madeleine" qui m'ont très bien accueilli pendant mon Master et sans qui ce travail de thèse n'aurait pas eu lieu. C'est grâce à vous que j'ai pu développer ma passion pour la bioinformatique et je suis ravi de pouvoir continuer de travailler avec vous pour la suite.

Je tiens également à remercier Youssef, M. Abbas, Alix, Mahmoud, Hussein, Thaer, M.A. Moussawi, Georges, Ronan, Chloé, Fouad, Orel, Sally et tous les autres amis extra-professionnels qui m'ont soutenu et m'ont permis de décompresser durant ces trois années : j'oublierai jamais nos tournois FIFA et nos soirées "jeux de cartes" autour d'une bonne shisha. Un grand merci aussi à toi Pierre, mon frère BEAR d'avoir rendu mes horaires de travail au bureau ainsi que nos missions professionnelles un peu plus amusantes et beaucoup moins solitaires.

Enfin, je tiens évidemment à remercier ma famille pour leur soutien et leurs encouragements, non seulement durant cette thèse, mais également durant tout le parcours qui m'a mené jusqu'ici.

À L'ÂME DE MON PÈRE

ET À MA MÈRE...

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

ADN	**A**cide **D**ésoxyribo**N**ucléique
AITL	**A**ngio **I**mmunoblastic **T**-cell **L**ymphoma
ALCL	**A**naplastic **L**arge **C**ell **L**ymphoma
ARN	**A**cide **R**ibo**N**ucléique
ALCL	**A**dult **T**-cell **L**eukemia **L**ymphoma
BAM	**B**inary **A**lignment/**M**ap
bp	**b**ase **p**airs
BWT	**B**urrows-**W**heeler **T**ransform
CCD	**C**harge-**C**oupled **D**evice
CCS	**C**ircular **C**onsensus **S**equence
cfDNA	**c**ell-**f**ree **DNA**
ctDNA	**c**irculating **t**umor **DNA**
CLR	**C**ontinuous **L**ong **R**ead
CNV	**C**opy **N**umber **V**ariation
CPU	**C**entral **P**rocessing **U**nit
DAG	**D**irected **A**cyclic **G**raph
DLBCL	**D**iffuse **L**arge **B**-**C**ell **L**ymphoma
ddNTP	**d**i**d**ésoxyribo**N**ucléotide **T**ri**P**hosphate
HTS	**H**igh-**T**hroughput **S**equencing
EBV	**E**pstein-**B**arr **V**irus
FL	**F**ollicular **L**ymphoma
FDR	**F**alse **D**iscovery **R**ate
FN	**F**aux **N**égatif
FP	**F**aux **P**ositif
GEP	**G**ene **E**xpression **P**rofiling
HBZ	**H**TLV-1 **B**asic leucine **Z**ipper factor
HMM	**H**idden **M**arkov **M**odel
IGV	**I**ntegrated **G**enome **V**iewer
ISFET	**I**on-**S**ensitive **F**ield-**E**ffect **T**ransistor
LPS	**L**inear **S**core **P**redictor
LNH	**L**ymphome **N**on **H**odgkinien
MCL	**M**antle **C**ell **L**ymphoma
MZL	**M**arginal **Z**one **L**ymphoma
MEM	**M**aximal **E**xact **M**atch
NAS	**N**etwork **A**ttached **S**torage
NGS	**N**ext **G**eneration **S**equencing
ONT	**O**xford **N**anopore **T**echnologies
PacBio	**P**acific **B**iosciences
PCR	**P**olymerase **C**hain **R**eaction
PTCL	**P**eripheral **T**-**C**ell **L**ymphoma
PPV	**P**ositive **P**redictive **V**alue
RAM	**R**andom **A**ccess **M**emory
RF	**R**andom **F**orest

RT-MLPA	**R**everse **T**ranscriptase **- M**ultiplex **L**igation**-**dependent **P**robe **A**mplification
SAM	**S**equence **A**lignment**/M**ap
scRNA-Seq	**s**ingle **c**ell **RNA-Seq**uencing
SLL	**S**mall **L**ymphocytic **L**ymphoma
SMRT	**S**ingle **M**olecule **R**eal **T**ime
SNP	**S**ingle **N**ucleotide **P**olymorphism
SNV	**S**ingle **N**ucleotide **V**ariation
SVM	**S**upport **V**ector **M**achine
TFH	**T F**ollicular **H**elper
UMI	**U**nique **M**olecular **I**dentifier
VAF	**V**ariant **A**llele **F**requency
VCF	**V**ariant **C**all **F**ormat
VN	**V**rai **N**égatif
VP	**V**rai **P**ositif
ZMW	**Z**ero**-M**ode **W**aveguides

Chapitre 1

Introduction

1.1 Préambule

Le traitement des données de séquençage à haut débit est un champ d'étude primordial du domaine de la bioinformatique. Ces données sont couramment utilisées dans plusieurs domaines, notamment la génomique et la transcriptomique. Dans la génomique, elles peuvent servir pour la détection des mutations somatiques SNV (*Single Nucleotide Variant*) ainsi que des variants structuraux CNV (*Copy Number Variation*), ces variations étant des marqueurs par excellence des maladies génétiques. De plus, elles permettent de reconstruire de nouveaux génomes, pour lesquels des références ne sont toujours pas établies. D'un autre côté, ces données sont particulièrement utiles dans la transcriptomique puisqu'elles peuvent être utilisées pour la quantification de l'expression génique ainsi que la détection des transcrits de fusion, anormalités de l'ARN (Acide RiboNucléique) et souvent marqueurs très forts de certains cancers. Cependant, les séquenceurs utilisés dans ce genre d'expériences ont souvent tendance à introduire des erreurs aléatoirement lors du séquençage ce qui crée des artefacts dans les séquences obtenues. L'ADN polymérase, enzyme permettant l'amplification de l'ADN (Acide DésoxyriboNucléique), elle-aussi représente une source supplémentaire d'artefacts dans les fragments d'ADN séquencés. Ces artefacts sont souvent introduits à très faible fréquence et pourraient être facilement confondus avec des vrais variants somatiques. L'utilisation récente des UMI (*Unique Molecular Identifier*) servant comme étiquette unique aux fragments séquencés a offert un solution permettant de filtrer les artefacts des données, facilitant ainsi l'analyse bioinformatique et la rendant plus précise. L'objectif de cette thèse est donc d'étudier les méthodes existantes se servant des UMI dans leurs algorithmes, et de proposer des améliorations, voire de nouveaux outils permettant une utilisation plus efficace des UMI dans les différents domaines d'application.

1.2 Contexte de travail

1.2.1 Le laboratoire LITIS et l'équipe TIBS

Le LITIS (Laboratoire d'Informatique, du Traitement de l'Information et des Systèmes) est une équipe d'accueil (EA 4108) Université de Rouen Normandie, Université du Havre Normandie et INSA Rouen Normandie et dirigée par le Pr. Laurent HEUTTE. Le LITIS est membre de l'école doctorale MIIS (Mathématiques, Information et Ingénierie des Systèmes) et du réseau d'intérêt normand « Normandie Digitale ». Il est partenaire de la fédération CNRS de recherche NormaSTIC. Il est en association avec le Groupe de Recherche en Informatique, Image, Automatique et Instrumentation (GREYC) de Caen, depuis janvier 2014. Le laboratoire comporte 7 équipes de recherche : l'équipe Apprentissage (App), l'équipe Combinatoire et Algorithmes (C&A), l'équipe Quantification en Imagerie Fonctionnelle (QuantIF), l'équipe Multi-agents, Interaction, Décision (MIND),

l'équipe Traitement de l'Information en Biologie Santé (TIBS), l'équipe Réseaux d'Interaction et Intelligence Collective (RI2C) et l'équipe Systèmes de Transport Intelligent (STI). Ainsi, les travaux du laboratoire associent le traitement de l'information bio-médicale, l'intelligence artificielle, l'apprentissage automatique et l'étude combinatoire et algorithmique des modèles. Les travaux menés dans le cadre de cette thèse s'inscrivent dans les thématiques de recherche de l'équipe TIBS, dirigée par le Pr. Thierry LECROQ. Au sein de cette équipe, les thématiques générales sont la modélisation statistique, l'indexation et l'extraction des informations de différents types de données biologiques, en particulier, celles issues des séquenceurs de nouvelle génération à haut débit.

1.2.2 Le Centre Henri Becquerel

Le Centre Henri Becquerel (CHB) est le centre de lutte contre le cancer (CLCC) de Haute-Normandie. Il est situé à Rouen et est actuellement dirigé par le professeur Pierre Vera. Cet établissement privé à but non lucratif assure une triple mission de soins, de recherche et d'enseignement, et constitue avec le CHU de Rouen le pôle de référence régional en cancérologie. Le CHB est particulièrement spécialisé en hématologie et oncologie médicale (sénologie, gynécologie et ORL). Il est de plus centre référent en radiothérapie et médecine nucléaire. Le Centre Henri Becquerel, c'est aujourd'hui :
— plus de 3 500 patients hospitalisés par an ;
— 150 lits d'hospitalisation ;
— plus de 700 médecins, soignants et techniciens ;
— 35 chercheurs.
Le Centre Henri Becquerel développe aussi des activités de recherche fondamentale dont les principaux axes portent sur l'hématologie et l'imagerie médicale. Il assure également des activités de recherche clinique dans différents domaines.

1.2.3 Unité Inserm 1245

L'unité Inserm 1245 a été créée en janvier 2017 et est composée de quatre équipes. L'une d'elles est hébergée principalement au Centre Henri Becquerel à Rouen et se concentre sur la génétique et la clinique des prolifération lymphoïdes, et en particulier des lymphomes. Cette équipe, dirigée par le Pr. Fabrice JARDIN, est monothématique et de nombreuses compétences y sont représentées. Elle est associée à un département d'hématologie clinique particulièrement spécialisé dans la prise en charge des patients atteints de lymphome, avec un recrutement d'environ 150 nouveaux cas par an. Ce recrutement permet notamment la collecte d'échantillons tumoraux, dans le cadre de protocoles de recherche. Ce département travaille en étroite collaboration avec une unité de recherche clinique, qui assure une actualisation en temps réel du suivi des patients. Un laboratoire de pathologie, qui joue un rôle central dans le diagnostic de ces tumeurs et gère la mise en banque des échantillons. À ce jour, plus de 1 000 biopsies congelées de lymphome y sont disponibles. Ce laboratoire a également développé un réseau pour collecter l'ensemble de ces biopsies au niveau régional. Un laboratoire de génétique dont la compétence est largement reconnue. Ses capacités techniques vont de la cytogénétique conventionnelle et moléculaire à la génomique et à l'analyse d'expression génique. Il dispose d'une plateforme d'analyse performante, qui comporte notamment des séquenceurs capillaires et plusieurs appareils de PCR (*Polymerase Chain Reaction*) en temps réel. Deux séquenceurs de nouvelle génération ont également été acquis récemment. Ces outils offrent d'excellentes perspectives dans les domaines fondamentaux et translationnels, comme pour la validation des marqueurs tumoraux.

1.2.4 Le lymphome

Les différents types de lymphome représentent des tumeurs hétérogènes du système lymphatique qui se développent aux dépens des lymphocytes B ou T, cellules jouant un rôle essentiel dans les réactions de défense immunitaire. Selon leur nature, les lymphomes sont dits hodgkiniens ou non hodgkinens, et ont des degrés de gravité variables. Ce sont des cancers relativement fréquents puisqu'ils se placent en France au sixième rang en terme d'incidence (4,8 cas pour 100 000 personnes) et au premier rang des cancers chez les adolescents et jeunes adultes (15-25 ans). On distingue principalement deux grands groupes de lymphome : les lymphomes B diffus à grandes cellules (DLBCL) et les lymphomes T (PTCL).

1.2.5 Le traitement des données de séquençage

Aujourd'hui, le séquençage de nouvelle génération NGS (*Next Generation Sequencing*) est devenu la méthode de référence pour la mesure de l'expression génique ainsi que la détection des anormalités génétiques dans l'ARN (transcrits de fusion) et l'ADN (SNV et CNV) des cellules tumorales. Les années 2000 ont marqué l'apparition des technologies de séquençage de deuxième génération qui produisent des *reads* de quelques centaines de paires de bases. Ces *reads* présentent des taux d'erreurs moyens de l'ordre 1% (la plupart sont des substitutions) ce qui les rend bien adaptés à l'analyse des détections des variants. Ensuite, au début des années 2010 ont suivi les technologies de troisième génération offrant la possibilité de séquencer des *reads* pouvant atteindre des centaines de milliers de paires de bases. Les *reads* produits par ces séquenceurs affichaient des taux d'erreur beaucoup plus élevés (10-30%) les rendant plus compatibles pour la résolution de problèmes d'assemblage. Ces technologies produisent d'énormes quantités de données sous forme de milliards de séquences lues, appelées *reads*, et représentant des régions génétiquement intéressantes dans le génome des tumeurs. Ainsi, vu la quantité importante d'information produite par ces séquenceurs, il est devenu primordial de développer des structures de données ainsi que des algorithmes permettant un traitement efficace et rapide des données produites. Dans tous les cas, l'utilisation de ces technologies nécessite une étape d'amplification par PCR suivie d'une étape de séquençage, pendant lesquelles des artefacts sont introduits dans les *reads* à de très basses fréquences. Ces artefacts sont souvent confondus avec de véritables variants de faible fréquence qui peuvent être trouvés dans les cellules tumorales et dans l'ADN plasmatique. Les UMI sont des séquences nucléotidiques aléatoires et uniques, introduites dans les fragments d'ADN avant l'amplification. L'utilisation récente de ces barcodes moléculaires dans des protocoles de séquençage ciblés a offert une approche fiable pour filtrer les artefacts et appeler avec précision les variants somatiques, même à de très faible fréquences. De plus, l'utilisation des UMI a permis de quantifier avec exactitude l'expression ciblée des gènes ainsi que la détection des anormalités dans l'ARN, sous forme de transcrits de fusion.

1.3 Objectifs

L'intégration de l'analyse des UMI dans les différents types d'analyse bioinformatique secondaire et tertiaire a conduit au développement des outils plus gourmands en mémoire que ceux basés sur des *reads* bruts (sans UMI), augmentant considérablement le temps de l'analyse. De ce fait, le but principal de cette thèse est de développer des outils capables d'intégrer cette analyse des UMI en implémentant des structures de données et des algorithmes spécifiquement conçus pour l'analyse supplémentaire de ces séquences. Ainsi, le premier objectif de la thèse est le développent d'un outil permettant de quantifier

la mesure d'expression génique sur un panel de gènes ciblés. De plus, ce même outil devrait être capable de s'adapter à un autre type d'analyse qui est la recherche de transcrits de fusion. Les données seront issues d'une expérience RT-MLPA (*Reverse Transcriptase - Multiplex Ligation-dependent Probe Amplification*) couplée à un séquenceur NGS. L'outil doit être implémenté dans une interface d'analyse permettant de faciliter et d'automatiser le plus possible le lancement des analyses par les biologistes ainsi que la production de résultats sous forme de fichiers bruts et graphiques facilement interprétables.

Le deuxième objectif de cette thèse est le développement d'un outil pour détecter les variants somatiques de très faible fréquence dans les fragments d'ADN étiquetés par des UMI. En effet, cet outil intégrera des algorithmes spécifiquement conçus pour rendre l'implémentation de l'analyse des UMI le plus efficace possible. L'intégration de cette analyse à l'outil permettra de réduire le taux de faux positifs dans la liste des variants trouvés, surtout pour les variants de très faible fréquence. L'outil doit être comparé à d'autres logiciels actuels pour démontrer son efficacité en termes de temps d'exécution et de consommation mémoire ainsi qu'en termes de sensibilité et spécificité de détection des variants.

La comparaison doit être faite en utilisant des données biologiques réelles mais aussi des données simulées. L'intérêt d'utiliser des données simulées est de pouvoir contrôler exactement la composition et la production des fichiers générés. Ainsi, le troisième objectif de cette thèse est de développer un simulateur de données permettant d'évaluer efficacement différents outils de détection de variants. Les simulateurs de *reads* avec des barcodes UMI permettront de reproduire le bruit de fond du séquenceur estimé à partir de données réelles et d'insérer des mutations déjà connues dans les fichiers produits ce qui rendra la comparaison entre les outils totalement objective et non biaisée. Plusieurs simulateurs de *reads* sont publiquement disponibles actuellement mais aucun d'entre eux n'offre la possibilité d'insérer des UMI dans les séquences produites, d'où l'intérêt de développer un tel outil et le rendre disponible aux autres développeurs pour effectuer leurs propres comparaisons.

1.4 Organisation du manuscrit

Ce manuscrit est composé de 7 chapitres. Les chapitres 1 et 2 décrivent le contexte du travail mené dans le cadre de cette thèse, ses objectifs, ainsi que l'état de l'art et les définitions liées aux technologies de séquençage et à l'utilisation des UMI dans les domaines de la génomique et la transcriptomique. Le Chapitre 3 reprend l'état de l'art en ce qui concerne les UMI. Le Chapitre 4 présente un nouvel outil hybride et puissant permettant d'analyser des expériences de mesure de l'expression génique et de détection de transcrits de fusion dans les tumeurs. Le Chapitre 5 introduit une nouvelle méthode de détection de variants somatiques dans les tumeurs. Cette méthode intègre une analyse des UMI très efficace et produit ainsi des résultats plus précis, surtout pour les variants de très faible fréquence. Le Chapitre 6 propose une nouvelle méthode pour simuler des données de séquençage NGS avec UMI. Cet outil est très utile pour comparer différents logiciels de détection de variants utilisant les UMI. Enfin, le Chapitre 7 propose une conclusion à cette thèse, et ouvre sur ses perspectives.

Chapitre 2

Le séquençage de l'ADN

2.1 Introduction

Dans ce chapitre, une courte description des principes biologiques et biochimiques de l'ADN est présentée pour situer le contexte global des travaux de cette thèse. Des raccourcis et des abréviations sont utilisés dans le but de ne présenter que les informations nécessaires pour la suite de ce manuscrit. Ce chapitre propose également un récapitulatif des technologies de séquençage décrites, par leur ordre chronologique d'apparition. Ensuite, quelques structures de données de base sont présentées. Ces dernières permettent le traitement efficace des données de séquençage d'un point de vue informatique et algorithmique. Enfin, l'exploitation de ces données de séquençage et les problématiques associées à leur utilisation dans des applications différentes sont abordées.

2.2 Codage de l'information dans l'ADN

L'ADN, ou acide désoxyribonucléique, est le matériel héréditaire indispensable au développement, au fonctionnement et à la reproduction de l'homme et presque tous les autres organismes. Presque toutes les cellules du corps d'une personne ont le même ADN. La plupart de l'ADN est située dans le noyau cellulaire (où il est appelé ADN nucléaire), mais une petite quantité d'ADN peut également être trouvée dans les mitochondries (où il est appelé ADN mitochondrial ou ADNmt). Les mitochondries sont des structures à l'intérieur des cellules qui convertissent l'énergie des aliments en une forme que les cellules peuvent utiliser.

Les informations contenues dans l'ADN sont stockées sous forme de code composé de quatre bases chimiques : l'adénine (A), la guanine (G), la cytosine (C) et la thymine (T). L'ADN humain se compose d'environ 3 milliards de bases, et plus de 99% de ces bases sont les mêmes chez toutes les personnes. L'ordre ou la séquence de ces bases détermine les informations disponibles pour la construction et l'entretien d'un organisme, de la même manière que les lettres de l'alphabet apparaissent dans un certain ordre pour former des mots et des phrases.

Les bases d'ADN s'apparient les unes aux autres, A avec T et C avec G, pour former des unités appelées paires de bases (ou bp pour *base pairs*). Ainsi, on dit que A et T et que C et G sont des bases complémentaires. Chaque base est également attachée à une molécule de sucre (le 2-désoxyribose) et une molécule de phosphate. Ensemble, une base, un sucre et un phosphate sont appelés un nucléotide. Les nucléotides sont disposés en deux longs brins qui forment une spirale appelée double hélice. La structure d'une molécule d'ADN est illustrée dans la partie droite de la Figure 2.1. La composition chimique de l'ADN a été découverte pour la première fois en 1869, mais son rôle dans l'héritage génétique n'a été démontré qu'en 1943. En 1953, James Watson et Francis Crick, aidés par les travaux des biophysiciens Rosalind Franklin et Maurice Wilkins, ont déterminé que

la structure de l'ADN est une double hélice polymère [1], une spirale constituée de deux brins d'ADN enroulés l'un autour de l'autre. La structure de la double hélice ressemble un peu à une échelle, les paires de bases formant les échelons de l'échelle et les molécules de sucre et de phosphate formant les parties latérales verticales de l'échelle. Cette percée a conduit à des progrès significatifs dans la compréhension des scientifiques de la réplication de l'ADN et du contrôle héréditaire des activités cellulaires.

L'ADN génomique est emballé de manière serrée et ordonnée suite à un processus appelé condensation de l'ADN, pour s'adapter aux petits volumes disponibles de la cellule. Chez les eucaryotes, l'ADN est situé dans le noyau cellulaire, avec de petites quantités dans les mitochondries et les chloroplastes. Chez les procaryotes, l'ADN est contenu dans un corps de forme irrégulière dans le cytoplasme appelé nucléoïde. L'information génétique dans un génome est contenue dans les gènes, et l'ensemble complet de cette information dans un organisme est appelé son génotype. Un gène est une unité d'hérédité, une région d'ADN qui influence une caractéristique particulière d'un organisme. Dans de nombreuses espèces, seule une petite fraction de la séquence totale du génome code pour la protéine. Par exemple, environ 1,5% seulement du génome humain est constitué d'exons codant pour des protéines, avec plus de 50% de l'ADN humain constitué de séquences répétitives non codantes. Cependant, certaines séquences d'ADN qui ne codent pas de protéine peuvent encore coder pour des molécules d'ARN non codantes fonctionnelles, qui sont impliquées dans la régulation de l'expression génique. Certaines séquences d'ADN non codantes jouent des rôles structurels dans les chromosomes. Les télomères et les centromères contiennent généralement peu de gènes mais sont importants pour la fonction et la stabilité des chromosomes.

Au sein d'un gène, la séquence de bases le long d'un brin d'ADN définit une séquence d'ARN messager, qui définit ensuite une ou plusieurs séquences protéiques. L'acide ribonucléique, ou ARN , est une molécule qui se compose également d'acides nucléiques, et dispose de propriétés très similaires à celles de l'ADN. L'ARN cependant n'est généralement formé que d'un seul brin comportant l'uracile (U) pour remplacer la thymine. La structure d'une molécule d'ARN est illustrée dans la partie gauche de la Figure 2.1. Il existe plusieurs types d'ARN dont les plus importants sont l'ARN messager (ARNm) qui est le produit de la transcription d'une partie codante de l'ADN et l'ARN de transfert (ou ARNt) qui sert à apporter les acides aminés au ribosome.

Lors de la transcription, les codons d'un gène sont copiés de l'ADN à l'ARNm par l'ARN polymérase. Cette copie d'ARN est ensuite décodée par un ribosome qui lit la séquence d'ARN en appariant les bases de l'ARN messager à un ARN de transfert portant les acides aminés aux ribosomes. La relation entre les séquences nucléotidiques de l'ARNm et les séquences d'acides aminés des protéines est déterminée par les règles de traduction, connues collectivement sous le nom de code génétique. Le code génétique se compose de «mots» de trois lettres appelés codons formés à partir d'une séquence de trois nucléotides (Figure 2.2). Puisqu'il y a 4 bases dans des combinaisons de 3 lettres, il y a 64 codons possibles (4^3 combinaisons). Ceux-ci codent pour les vingt acides aminés standards, donnant à la plupart des acides aminés plus d'un codon possible. Il existe également trois codons «stop» signifiant la fin de la région codante : ce sont les codons UAA, UGA et UAG.

2.3 Technologies de séquençage

Le séquençage de l'ADN a été inventé en 1977, par deux équipes de recherche indépendantes : la première dirigée par Frederick Sanger, à l'université de Cambridge [4], et la seconde menée par Allan Maxam et Walter Gilbert, à l'université de Harvard [5]. Les

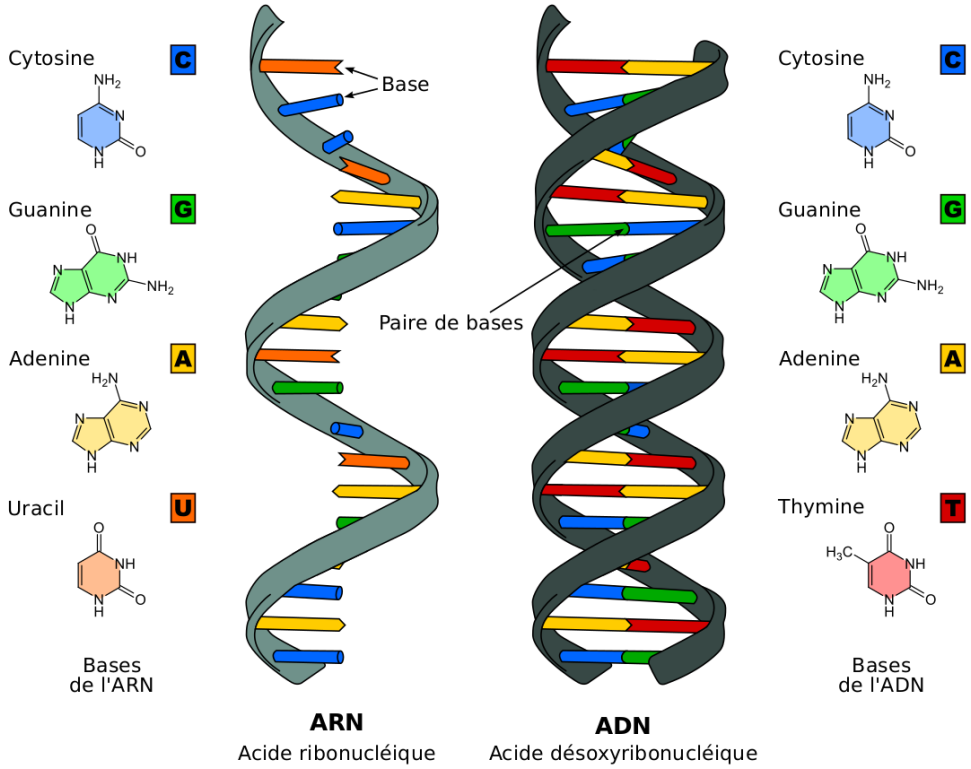


FIGURE 2.1 – Composition et structure d'une molécule d'ADN et d'une molécule d'ARN. Figure adaptée de [2].

travaux de recherche indépendants ont valu à Sanger et Gilbert le prix Nobel de chimie en 1980. Grâce à cette découverte, la composition de l'ADN et le code génétique de toute espèce vivante ont pu être étudiés en détail faisant d'elle une avancée très importante dans le monde de la biologie.

L'objectif principal du séquençage de l'ADN est de convertir l'information présente sur un fragment d'ADN en une séquence composée de 5 lettres appartenant à l'alphabet de l'ADN, $\Sigma = \{A, C, G, T, N\}$. Dans cet alphabet, chaque lettre correspond à une base azotée (ou nucléotide) et la lettre N pouvant être n'importe quelle base (elle sert à dénoter une incertitude de la lecture à une position donnée). Les chaînes de caractères ainsi formées sont appelées *reads* et représentent des fragments de l'ADN de l'échantillon séquencé. Plusieurs technologies de séquençage sont apparues avec le temps permettant d'offrir des solutions à des problématiques et des applications différentes. Selon la technologie utilisée, les *reads* produits vont contenir des erreurs de séquençage (substitution, insertion ou délétion) dont la fréquence peut varier entre 0,001 et 30%.

Les technologies de Sanger et de Maxam-Gilbert sont restées longtemps comme les technologies de séquençage standards utilisées par les laboratoires biologiques. En particulier, grâce à sa faible radioactivité et son efficacité relativement élevée, la technologie de Sanger a été la méthode la plus utilisée. Cependant, les avancées ont continué à arriver et en s'inspirant de ces premières technologies, les chercheurs ont pu développer de nouvelles technologies qui sont à la fois plus efficaces, moins chères et plus rapides. En effet, le séquençage de l'ADN a largement évolué depuis 1977. Dans cette partie, une liste des technologies de séquençage est présentée, commençant par les technologies dites de première génération comme celle de Sanger jusqu'aux technologies de troisième génération, très récentes. La Table 2.1 résume les caractéristiques principales de ces plateformes

LE CODE GÉNÉTIQUE

ARN messager Codon : première base azotée		ARN messager Codon : deuxième base azotée U	C	A	G		ARN messager Codon : troisième base azotée
	U	Phe	Ser	Tyr	Cys	U	
		Phe	Ser	Tyr	Cys	C	
		Leu	Ser	STOP	STOP	A	
		Leu	Ser	STOP	Trp	G	
	C	Leu	Pro	His	Arg	U	
		Leu	Pro	His	Arg	C	
		Leu	Pro	Gln	Arg	A	
		Leu	Pro	Gln	Arg	G	
	A	Ile	Thr	Asn	Ser	U	
		Ile	Thr	Asn	Ser	C	
		Ile	Thr	Lys	Arg	A	
		Met	Thr	Lys	Arg	G	
	G	Val	Ala	Asp	Gly	U	
		Val	Ala	Asp	Gly	C	
		Val	Ala	Glu	Gly	A	
		Val	Ala	Glu	Gly	G	

FIGURE 2.2 – Le code génétique permettant la traduction des codons en acides aminés [3].

de séquençage. Dans ce qui suit, les aspects biochimiques des différentes technologies seront expliqués brièvement vu qu'ils sortent du cadre de cette thèse, et que plusieurs publications existent pour présenter le principe détaillé de chaque technologie [6, 7, 8, 9, 10].

2.3.1 Première génération

2.3.1.1 Sanger

La méthode de Sanger est basée sur un principe de séquençage par synthèse. Ce type de séquençage utilise un des deux brins d'ADN comme modèle et se sert de nucléotides chimiquement modifiés appelés didésoxyribonucléotides(ddNTP). Il existe quatre ddNTP différents, chacun associé à une base azotée spécifique ; ddATP, ddCTP, ddGTP et ddTTP. En incorporant un de ces ddNTP dans le brin d'ADN synthétisé par l'ADN polymérase, l'élongation est immédiatement stoppée. Par exemple, l'élongation s'arrêtera au niveau d'un nucléotide C suite à l'incorporation d'un ddCTP. Ainsi, afin d'assurer l'achèvement du séquençage, la réaction doit obligatoirement être réalisée quatre fois en parallèle, chacune avec un ddNTP différent.

Selon l'endroit où a été incorporé le ddNTP, chaque réaction produira un ensemble de fragments d'ADN de tailles différentes. Ensuite, grâce à une électrophorèse sur une plaque de gel polyacrylamide, les fragments sont séparés en fonction de leurs tailles et chaque réaction aura une ligne indépendante sur la plaque de gel utilisée. Alors, sur cette dernière et sur chaque ligne apparaîtra des bandes distinctes, chacune indiquant les ddNTP qui ont été incorporés dans les séquences ainsi que leur positions respectives. Un système d'imagerie par rayons X ou par lumière ultra-violette assure la visualisation des nucléotides sous forme de bandes. Dans le cas des rayons X, un traceur radioactif doit préalablement être introduit dans l'ADN séquencé. Afin d'éliminer la radioactivité des expériences de séquençage, le traceur a été ultérieurement remplacé par un traceur fluorescent dans des versions plus récentes de la technologie. La Figure 2.3 illustre le processus du séquençage Sanger.

Le séquençage Sanger permet ainsi de produire des *reads* de longueur moyenne de 500 à 600 paires de bases avec un taux d'erreur très faible, de l'ordre de 0,001%. De plus, cette technologie a été utilisée pour séquencer et publier la première version du génome humain [12], étant la méthode de séquençage par défaut à l'époque.

2.3.1.2 Maxam-Gilbert

La méthode de séquençage de Maxam-Gilbert repose sur une réaction de dégradation chimique. À la différence de Sanger qui utilise un seul brin, cette technologie se sert des deux brins de l'ADN à séquencer après l'avoir marqué à son extrémité 5' avec un traceur radioactif. Ensuite, après avoir séparé les deux brins de chaque fragment, des coupures y sont créées en profitant des réactivités différentes des quatre nucléotides. Ainsi, quatre types de réactions différentes existent : une pour les G, une pour les C, une pour les C et les T et une dernière pour les G et les A. Ces réactions sont effectuées en parallèle, sur une partie de chaque brin d'ADN.

Pour s'assurer qu'une modification seulement soit apportée sur chaque fragment d'ADN, les quantités de réactifs utilisés sont bien contrôlées. Ainsi, selon l'endroit où la coupure a été réalisée, ces réactions produisent un ensemble de fragments de tailles différentes. Ces fragments sont ensuite séparés selon leurs tailles, par électrophorèse, sur une plaque de gel de polyacrylamide, comme pour la méthode de Sanger. De la même façon, les positions des différents nucléotides dans la séquence sont repérées par une

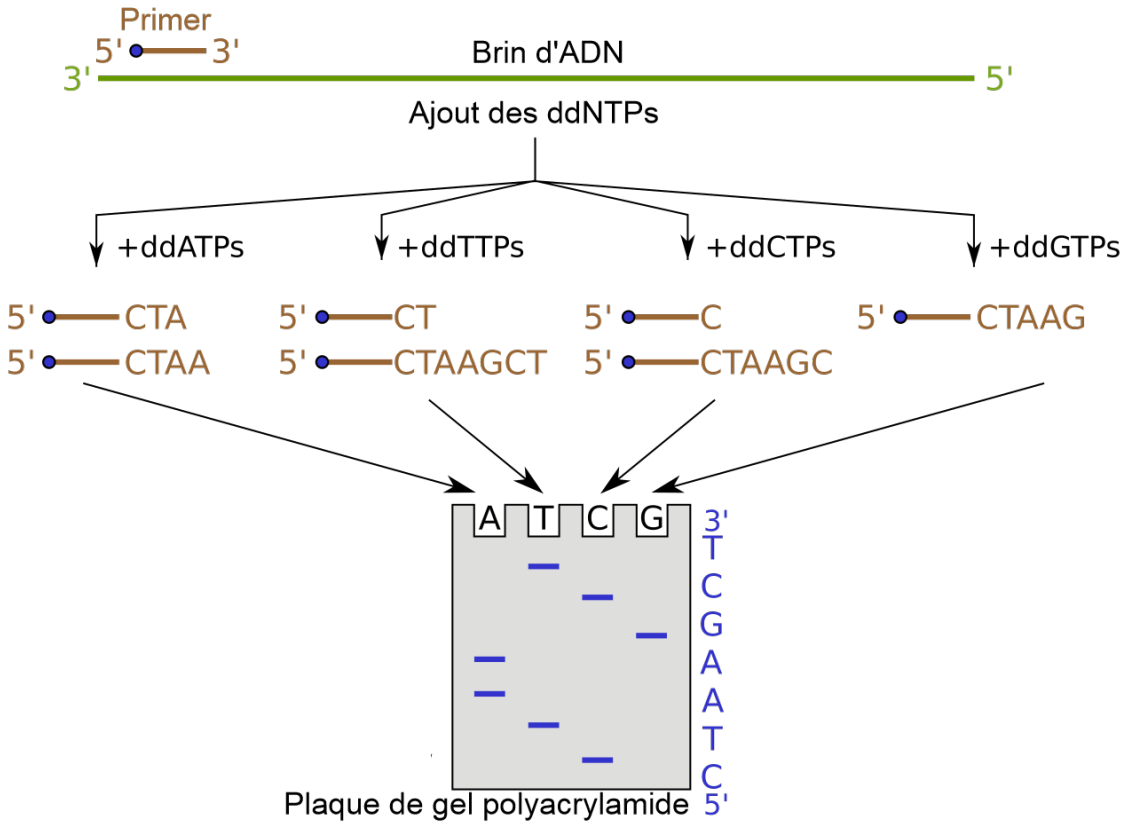


FIGURE 2.3 – Le séquençage de Sanger. Figure adaptée de [11].

bande sur la plaque de gel, à l'aide d'un système d'imagerie. Le processus de séquençage Maxam-Gilbert est illustré dans la Figure 2.4.

Tout comme le séquençage Sanger, le séquençage Maxam-Gilbert permet également de produire des *reads* de longueur moyenne de 500 à 600 paires de bases, affichant un faible taux d'erreurs, de l'ordre de 0,001%.

2.3.2 Deuxième génération

Même si les technologies de première génération (surtout celle de Sanger) ont dominé le marché du séquençage durant près de 30 ans, ces plateformes présentaient toujours deux inconvénients : un temps élevé nécessaire au séquençage et un coût élevé. Ainsi, une nouvelle génération de séquenceurs est apparue à partir de 2005 pour contourner ces limitations. Grâce à ces nouvelles plateformes, dites de deuxième génération, on a pu réduire considérablement le coût et le temps de séquençage puisqu'une expérience devient capable de générer plusieurs millions de *reads* courts en parallèle. De plus, la sortie du séquençage peut ici être directement détectée, sans avoir recours à une électrophorèse ou à un système d'imagerie. Cette génération de technologies de séquençage est généralement qualifiée de NGS, pour *Next Generation Sequencing*, ou de séquençage à haut débit (ou HTS, pour *High-Throughput Sequencing*). Les quatre principaux acteurs de ces technologies de deuxième génération sont présentés ici.

2.3.2.1 Roche/454

La technologie Roche/454 est apparue en 2005 et repose sur une approche de séquençage par synthèse appelée pyroséquençage. Après avoir fragmenté - aléatoirement

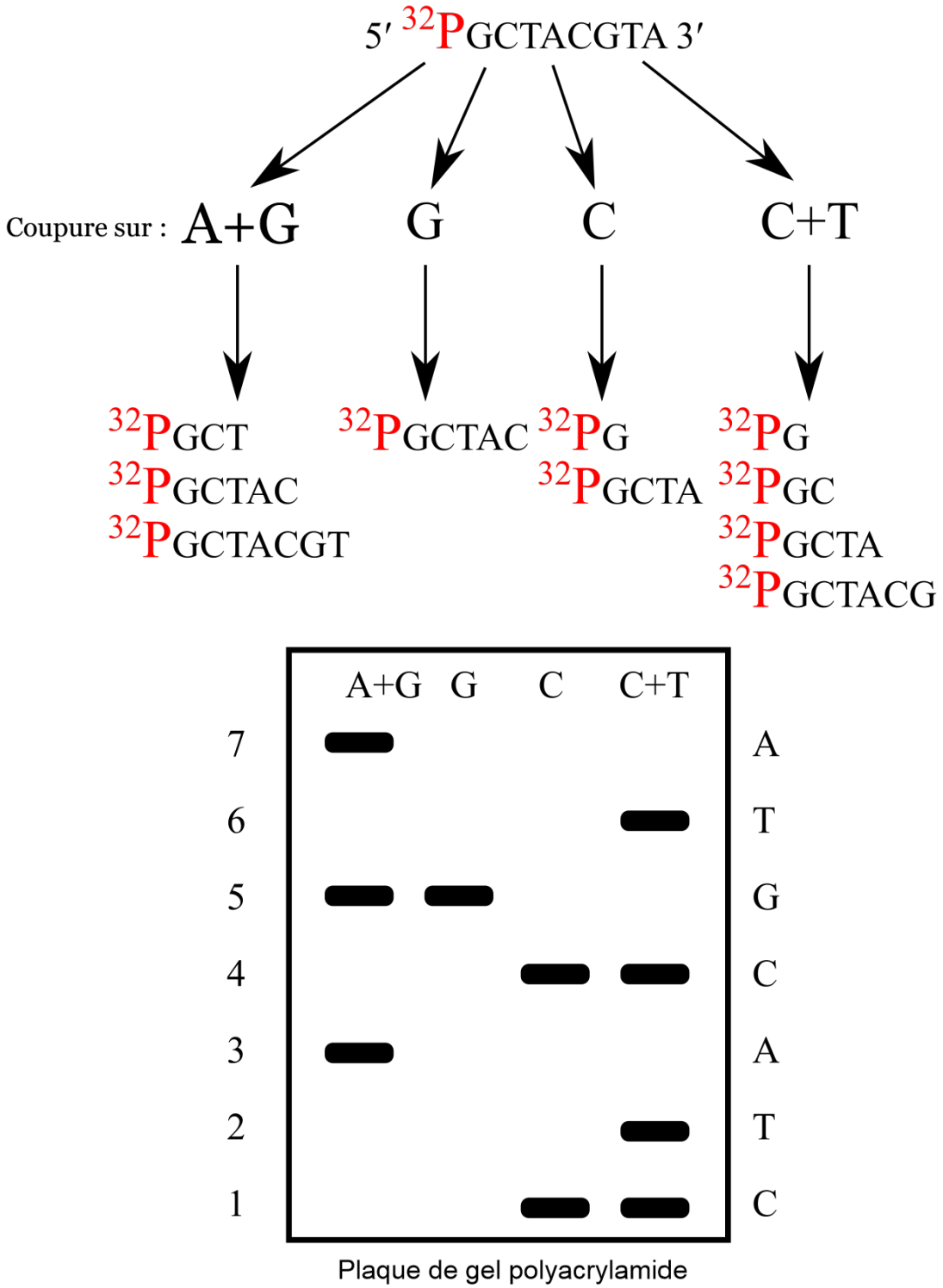


FIGURE 2.4 – Le séquençage de Maxam-Gilbert. Figure adaptée de [13].

- l'échantillon d'ADN à séquencer, on attache aux extrémités des fragments obtenus des adaptateurs dont les amorces complémentaires sont présentes sur des billes. Ceci permettra la fixation des fragments d'ADN aux billes de sorte qu'un seul fragment seulement soit retenu par une bille. Chaque bille est ensuite isolée, et le fragment d'ADN associé est amplifié par PCR en émulsion. Grâce à cette amplification, des millions de copies de chaque fragment d'ADN seront produites à la surface des billes. Ces dernières sont ensuite transférées sur une plaque PTP (*Pico Titer Plate*) contenant un grand nombre de puits dans lesquels les réactions de pyroséquençage auront lieu. Les puits ont un diamètre spécifique afin de ne pouvoir contenir qu'une seule bille.

Ensuite, d'une manière séquentielle et selon un ordre précis, les quatre nucléotides sont ajoutés sur la plaque PTP. Les puits de la plaque contiennent une ADN polymérase qui, suite à l'introduction d'un certain nucléotide, va l'incorporer aux fragments d'ADN sur la surface de la bille. Cette réaction produira un pyrophosphate inorganique qui à son tour, et sous l'action de la luciférase, générera un signal lumineux d'intensité proportionnelle au nombre de nucléotides ajoutés. Donc, l'incorporation d'une région où la même base est répétée plusieurs fois (appelée homopolymère) produira un signal plus fort que celui obtenu suite à l'addition d'un seul nucléotide. Un capteur CCD (*Charge-Couple Device*) est utilisé pour détecter la suite des signaux émis suite à l'incorporation des différents nucléotides. Le processus du séquençage Roche/454 est expliqué dans la Figure 2.5.

La plateforme Roche/454 permet d'obtenir des *reads* d'une longueur supérieure à ceux obtenus par les autres technologies de séquençage de deuxième génération. Les *reads* peuvent atteindre 700 paires de base et présentent des taux d'erreur relativement faible de l'ordre de 1% en moyenne. La plupart des erreurs ont lieu dans des régions présentant des insertions, des délétions et des homopolymères. Ceci s'explique par le fait que l'intensité du signal produit par la luciférase définit la taille de l'homopolymère incorporé et que le capteur du signal puisse parfois sous ou surestimer le nombre de nucléotides à ajouter.

2.3.2.2 Illumina/Solexa

La plateforme de séquençage Solexa est apparue en 2006 et a été renommée ensuite Illumina lors de son achat par Illumina en 2007. Aujourd'hui, cette dernière s'est imposée comme la technologie de séquençage de deuxième génération la plus utilisée. Cette technologie repose sur une technique de séquençage par synthèse. Comme pour la technique Roche, l'échantillon d'ADN commence par subir une fragmentation aléatoire suivie par une fixation d'adaptateurs aux extrémités des fragments obtenus. Grâce à ces adaptateurs, les fragments sont ensuite attachés à des *flow cells* contenant des oligonucléotides complémentaires. Une fois les fragments fixés, des millions de copies identiques sont créées suite à une amplification par PCR *bridge*. On appelle *cluster* un ensemble de fragments provenant d'un même fragment initial. Une fois les *cluster*s produits et l'amplification terminée, on ajoute les nucléotides modifiés, l'ADN polymérase et les amorces aux *flow cells*. Les amorces s'attacheront aux fragments d'ADN et seront étendues par incorporation de nouveaux nucléotides sous l'action de l'ADN polymérase. Cependant, ces nucléotides sont modifiés de façon à ce que chaque type de nucléotide soit attaché à un marqueur fluorescent spécifique. Pour s'assurer que la polymérase ne puisse incorporer qu'un seul ADN à la fois, les nucléotides subissent une deuxième modification : un terminateur réversible est ajouté à chacun d'eux. À chaque cycle, le nucléotide ajouté est déterminé selon la longueur d'onde de son marqueur fluorescent détecté par le capteur CCD. Le terminateur et les nucléotides non incorporés sont ensuite retirés de la *flow cell*

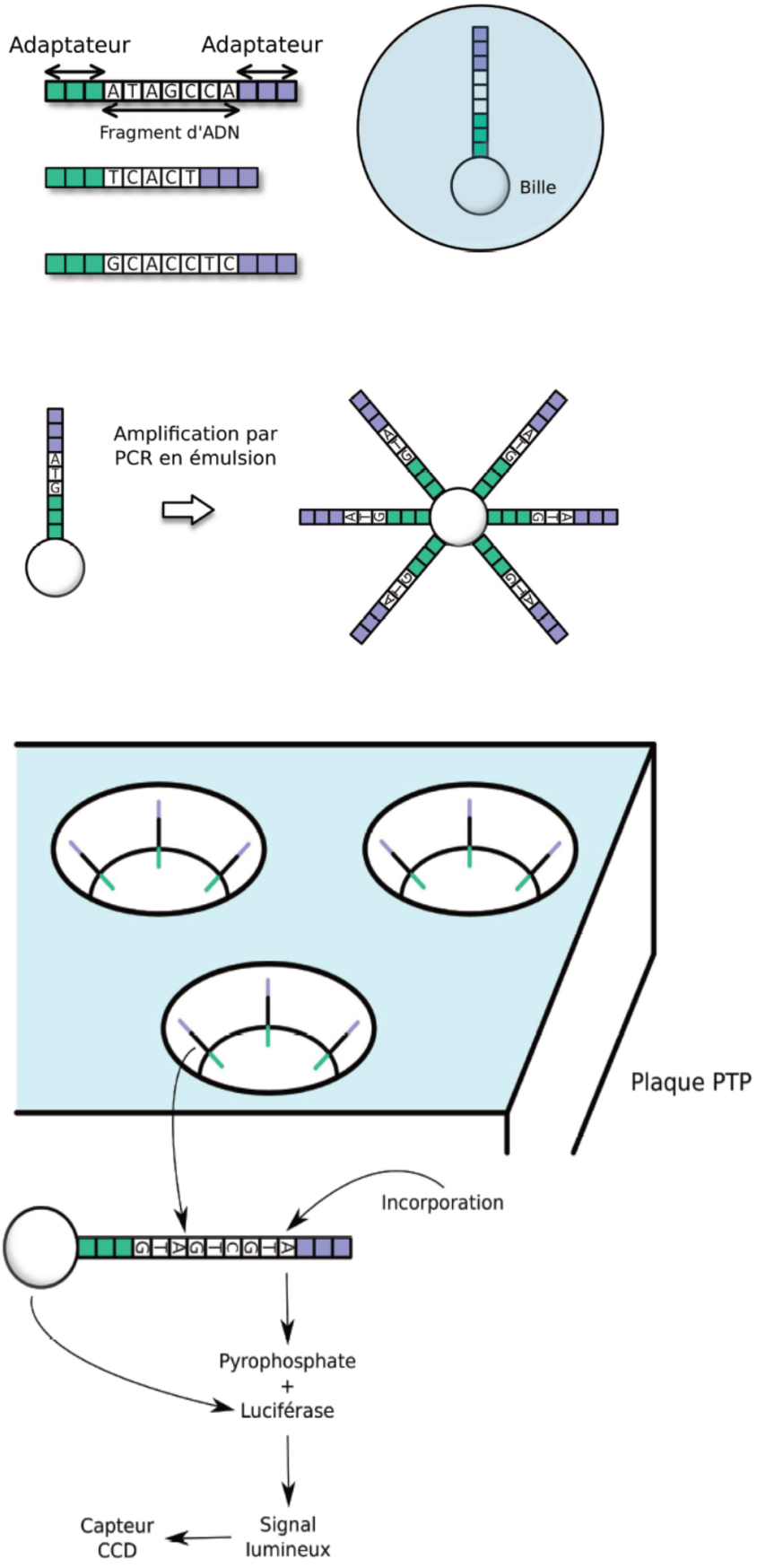


FIGURE 2.5 – Le séquençage de Roche 454. Figure adaptée de [14].

permettant ainsi au processus de continuer avec une nouvelle incorporation à la fois jusqu'à ce que le fragment d'ADN soit complètement séquencé. Le processus de séquençage Illumina est illustré dans la Figure 2.6.

Le séquençage Illumina permet ainsi de produire des *reads* de longueur moyenne de 100 à 300 paires de bases, avec un taux d'erreurs de l'ordre de 1% en moyenne. Les erreurs de séquençage de ces *reads* sont principalement des substitutions, dues à une mauvaise identification du nucléotide incorporé, et sont principalement situées aux extrémités des *reads*.

2.3.2.3 ABI/SOLiD

Apparue en 2007, la technologie ABI/SOLiD repose sur une méthode de séquençage par ligation. L'échantillon d'ADN est tout d'abord fragmenté d'une manière aléatoire et les fragments obtenus sont attachés à des billes grâce à des amorces et des adaptateurs et amplifiés par PCR en émulsion, comme pour la plateforme Roche/454. Ensuite, les billes sont fixées à une lame de verre d'une façon covalente. La réaction de séquençage par ligation peut ainsi commencer. Pour cela, on utilise des 8-mers (des fragments d'ADN de taille 8) spécifiques présentant un marqueur fluorescent spécifique sur l'extrémité 5'. Les 8-mers ont une structure bien déterminée : les deux premières bases de l'extrémité 3' sont complémentaires aux nucléotides en cours de séquençage, les trois bases suivantes sont dégénérées et donc peuvent s'appareiller avec n'importe quels nucléotides et les trois dernières bases sont elles aussi dégénérées mais sont retirées avec le marqueur fluorescent lors de la réaction. Plusieurs passes se produisent durant l'étape de séquençage qui suit, et chaque passe comporte plusieurs cycles. Au début de chaque passe, on ajoute une amorce complémentaire à l'adaptateur utilisé pour fixer le fragment d'ADN à la bille. Lors de chaque cycle, les 8-mers décrits ci-dessus sont ajoutés et ligaturés en tenant compte des deux premières bases de leur extrémité 3'. Le 8-mer qui se lie au fragment d'ADN en cours de séquençage émettra un signal lumineux qui sera détecté, mesuré et enregistré alors que les 8-mers non liés sont retirés de la lame de verre. Les amorces sont choisies de sorte que chaque base soit lue deux fois lors du séquençage. Les données obtenues par cette technique sont représentées par des couleurs, chaque couleur représentant un 2-mer spécifique. L'analyse de ces données permettra finalement de déduire la séquence du fragment d'ADN séquencé. La Figure 2.7 montre le processus de séquençage ABI/SOLiD.

Cette technique de séquençage permet alors d'obtenir des *reads* courts avec une longueur comprise entre 50 et 75 paires de base et présentant un taux d'erreur très faible (environ 0,1%) puisque chaque base est lue deux fois pendant le séquençage. Les erreurs sont généralement des substitutions : elles sont la conséquence du bruit généré pendant les cycles de ligation entraînant une mauvaise identification des bases.

2.3.2.4 Ion Torrent

La plateforme de séquençage Ion Torrent est apparue en 2010 et est basée sur la détection de l'ion hydrogène produit suite à l'incorporation d'un nucléotide. Comme pour les technologies de séquençage précédentes, il faut commencer par fragmenter aléatoirement l'ADN à séquencer. Ensuite, chaque fragment est attaché à une bille grâce aux adaptateurs et aux amorces et amplifié par PCR en émulsion. On utilise une puce composé d'un ensemble de puits qui contiennent chacun une bille.

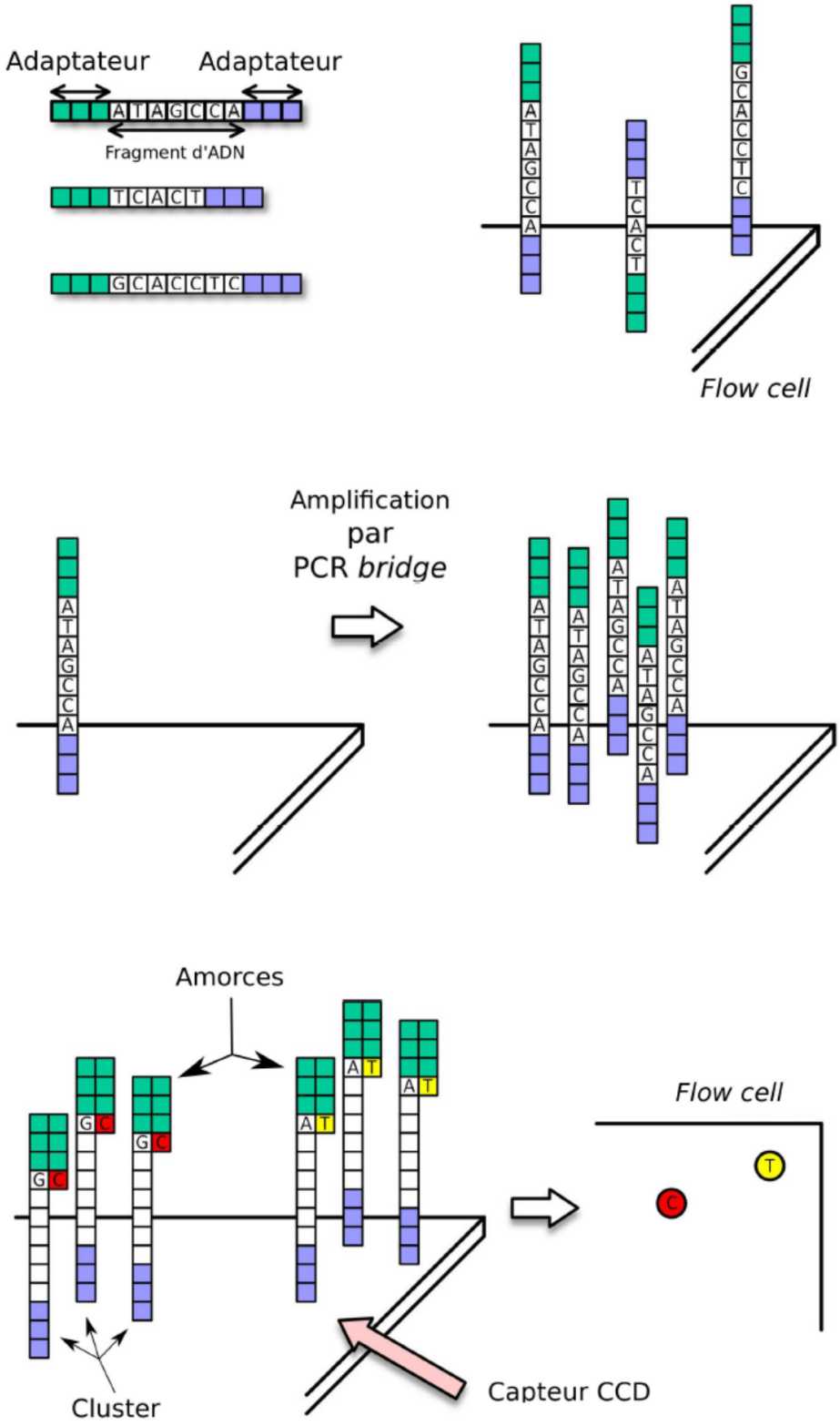


FIGURE 2.6 – La technique de séquençage de la plateforme Illumina. Figure adaptée de [14].

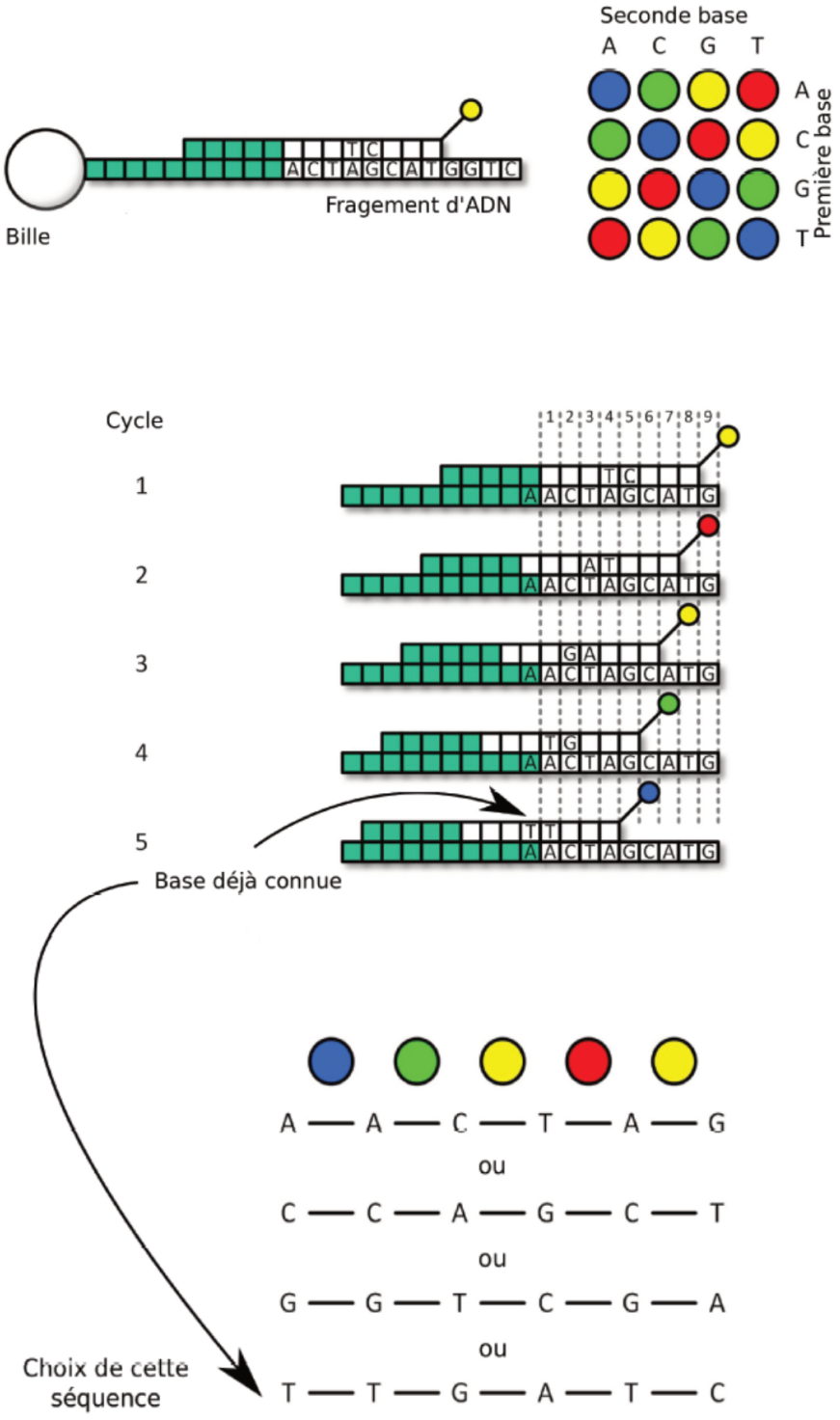


FIGURE 2.7 – Le séquençage de la plateforme ABI/SOLiD. Figure adaptée de [14].

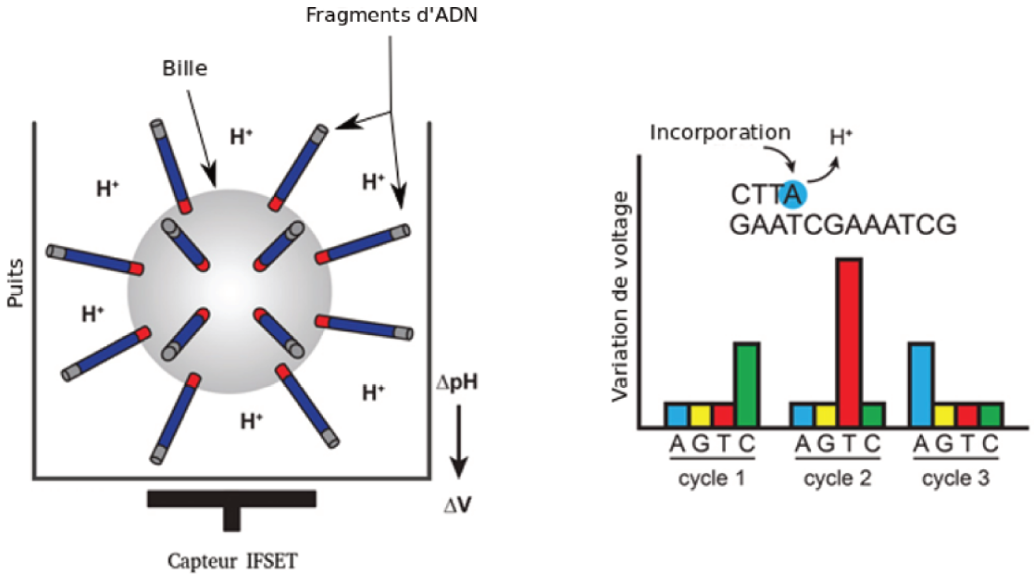


FIGURE 2.8 – Le séquençage de la plateforme Ion Torrent. Figure adaptée de [14].

D'une manière séquentielle, les différents nucléotides sont alors ajoutés pour les incorporer aux fragments d'ADN fixés sur les billes. L'ADN polymérase libère un ion hydrogène en incorporant un nucléotide ce qui entraîne un changement du pH de la solution. Ce changement est converti en un signal électrique et est détecté à l'aide d'un appareil ISFET (*Ion-Sensitive Field-Effect Transistor*) présent au fond du puits. Ce signal est unique à l'incorporation de chaque nucléotide et son intensité est proportionelle à la longueur de l'homopolymère incorporé. La suite des signaux électriques est finalement transmise à un ordinateur et est transduite directement en une séquence d'ADN. Le processus de séquençage Ion Torrent est illustré dans la Figure 2.8.

Cette technique de séquençage permet ainsi de produire des *reads* de taille entre 100 et 400 paires de base avec un taux d'erreur de 1% en moyenne. Les erreurs sont principalement des insertions et des délétions, surtout au niveau des homopolymères, dues à la difficulté de bien interpréter l'intensité du signal électrique enregistré.

2.3.2.5 Les *reads* pairés

Les séquenceurs de deuxième génération permettent aussi de produire des *reads* pairés ou paires de *reads*, dont chacun provient d'une des deux extrémités du fragment d'ADN séquencé. Deux types de *reads* pairés existent : les *reads paired-end* et les *reads mate-pair*. La différence principale entre ces deux reads pairés est la taille du fragment séparant les deux *reads* appariés. Dans le cas des *reads paired-end*, la longueur de l'insert est relativement courte (200-800 pb) alors que dans le cas des *reads mate-pair*, celle-ci dépasse généralement les 1000 pb. Dans les deux cas, le *read* pairé associé à un *read* court donné est généralement appelé *mate*. Pour chacun de ces *reads*, les informations concernant son *read mate* ainsi que la distance les séparant sur le fragment sont alors connues. Ces informations sont très utiles lors de l'alignement des *reads* : elles permettent de résoudre des difficultés lors de l'assemblage surtout dans des régions répétées mais aussi des réarrangements structuraux comme des inversions ou des délétions.

2.3.3 Troisième génération

Même si aujourd'hui les technologies de deuxième génération demeurent les plateformes de séquençage les plus utilisées, elles présentent deux désavantages principaux. Le premier est qu'elles reposent toutes sur une étape essentielle qui est longue et coûteuse : l'amplification par PCR. Le second est la taille des *reads* produits : les *reads* courts générés sont considérés comme facteur bloquant dans l'assemblage et l'analyse des génomes complexes. Pour cela, une nouvelle génération de séquenceurs s'est développée à partir de 2011 ayant comme but principal de résoudre ces deux problèmes. Les séquenceurs de troisième génération ne se servent pas d'une étape d'amplification par PCR rendant ainsi le séquençage plus rapide, plus aisé et moins coûteux. De plus, ces nouveaux séquenceurs sont capables de générer des *reads* avec des longueurs pouvant atteindre des milliers voire des centaines de milliers de paires de bases. Ainsi, ces *reads* sont capables de couvrir des régions beaucoup plus longues et des régions répétées permettant donc de résoudre des problèmes d'assemblage, tâche fastidieuse en utilisant des *reads* courts. Par contre, les *reads* produits sont beaucoup plus bruités que ceux de deuxième génération avec des taux d'erreur entre 10 et 30%. À la différence des *reads* de deuxième génération, les erreurs de troisième génération sont majoritairement des insertions et des délétions. De plus, ces nouvelles plateformes sont plus susceptibles de produire des *reads* formés de séquences qui ne sont pas contiguës au sein du génome de référence, appelés *reads* chimériques.

Contrairement aux technologies de deuxième génération, dominées par Illumina, aucune technologie de troisième génération domine le marché actuellement. Les deux plateformes principales de troisième génération, Pacific Biosciences et Oxford Nanopore Techologies, sont présentées ici. D'autres technologies existent aussi comme 10x Genomics et Illumina True-Seq (anciennement Moleculo) mais elles reposent sur des méthodologies extrêmement différentes, et sont beaucoup moins utilisées que les deux premières et donc ne seront pas présentées.

2.3.3.1 Pacific Biosciences

La technologie Pacific Biosciences (PacBio), apparue en 2011, repose sur une méthode de séquençage moléculaire simple en temps réel ou SMRT (*Single Molecule Real Time*). Cette méthode se base sur un séquençage par synthèse produisant un signal lors de l'incorporation d'un nucléotide et qui est capturé en temps réel. Pour cela, on a recours à des cellules ZMW (*Zero-Mode Waveguides*) sous forme de puits de quelques dizaines de nanomètres contenant au fond un seul fragment d'ADN, l'ADN polymérase et un capteur pour détecter les signaux lumineux émis. Des marqueurs fluorescents uniques sont attachés aux différents nucléotides et donc un signal lumineux spécifique caractérise l'incorporation d'un nucléotide déterminé. Vu la taille des ZMW, on ne peut détecter que le signal émis le plus au fond du puits permettant ainsi de déduire la base qui vient d'être incorporée et en déduire la séquence du fragment d'ADN en temps réel. Le processus de séquençage PacBio est illustré Figure 2.9.

Pour améliorer la précision du séquençage, une nouvelle stratégie a été mise en place. Elle consiste à ajouter des adaptateurs en épingle à cheveux à chaque extrémité des deux brins permettant de le relier afin que l'ADN polymérase puisse boucler autour du fragment et alors le traverser plusieurs fois. À chaque fois que l'ADN polymérase effectue un tour complet, la séquence obtenue est appelée un *subread* et les *subreads* seront utilisés pour déduire une séquence de meilleure qualité, appelée CCS (*Circular Consensus Sequence*). Ainsi, le nombre de *subreads* utilisés pour construire la séquence CCS et donc le nombre de cycles réalisés par l'ADN polymérase déterminera la longueur et la qualité

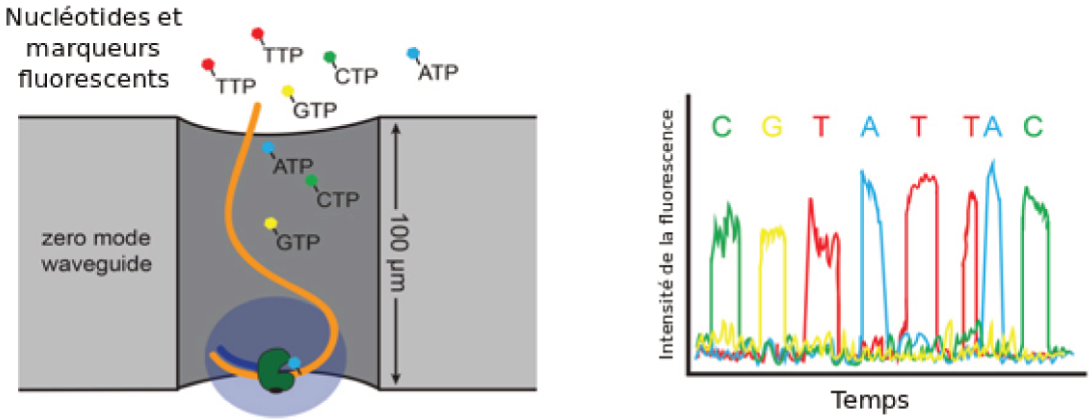


FIGURE 2.9 – Le séquençage de la plateforme Pacific Biosciences. Figure adaptée de [14].

des *reads* CCS. D'autres *reads* peuvent être générés sans consensus et sont appelés des *reads* CLR (*Continuous Long Read*). Le séquençage PacBio permet ainsi de produire des *reads* pouvant atteindre une taille de quelques dizaines de milliers de paires de bases en moyenne. Cependant, ces *reads* affichent un taux d'erreur très élevé situé entre 10 et 15% en moyenne. La grande majorité de ces erreurs sont des insertions et sont réparties d'une manière aléatoire tout au long des *reads*.

2.3.3.1 Oxford Nanopore Technologies

Apparue en 2014, la plateforme de séquençage Oxford Nanopore Technologies (ONT) a été mise sur le marché en 2015. Elle est basée sur l'utilisation des pores constitués de protéine, appelés nanopores, laissant passer le fragment d'ADN. En le traversant, le fragment d'ADN va modifier le courant ionique à l'intérieur du pore. Cette variation est spécifique au nucléotide incorporé et peut être détectée, enregistrée puis analysée pour déterminer la séquence du fragment d'ADN. La stratégie se servant des adaptateurs à épingle à cheveux pour relier les deux brins d'ADN est aussi utilisée ici leur permettant ainsi de traverser le pore l'un après l'autre. La séquence de chaque brin peut facilement être déduite par la suite puisque la séquence - connue - de l'adaptateur les sépare. Les séquences peuvent soit être séparées pour former des *reads* 1D, soit utilisées pour améliorer la qualité de la séquence consensus et obtenir des *reads* 2D. Bien qu'il permet un gain net en terme de précision, dans certains cas, l'adaptateur en épingle à cheveux produisait une structure secondaire qui ralentissait le passage de l'ADN dans le pore et impactait négativement les résultats. Une autre technologie, produisant des *reads* appelées $1D^2$, l'a donc remplacée à partir de 2017. Dans cette nouvelle méthode, des protéines sont attachées à chaque brin leur permettant de passer l'un après l'autre sans que les deux brins soient réellement liés. Malheureusement, il peut arriver qu'un brin passe à travers le nanopore sans l'autre empêchant ainsi l'obtention d'une séquence consensus. Le processus de séquençage ONT est montré dans la Figure 2.10.

Plusieurs plateformes de séquençage sont proposées par Oxford Nanopore technologies. On y distingue notamment le MinION, disponible au prix de 1000$ et pouvant fonctionner à l'aide d'une simple connexion USB à un ordinateur rendant le séquençage plus pratique et peu coûteux. Le MinION ne mesure que quelques dizaines de centimètres mais ONT, afin de rendre le séquençage encore plus accessible, est en train de

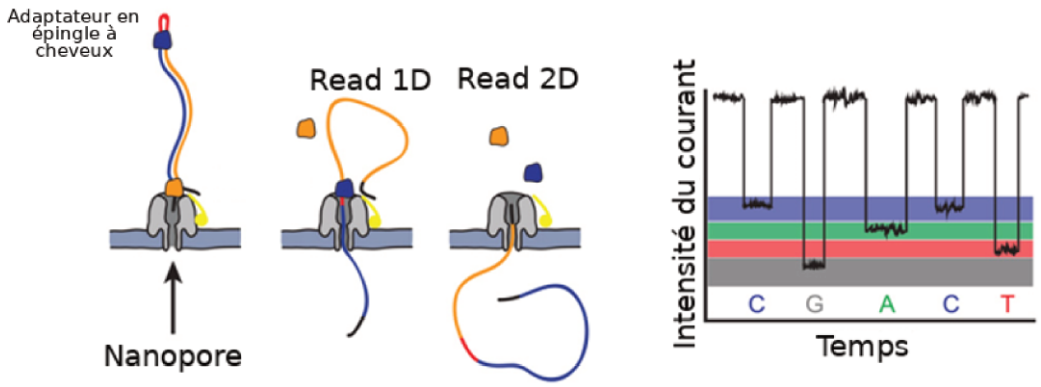


FIGURE 2.10 – Le séquençage de la plateforme Oxford Nanopore Technologies. Figure adaptée de [14].

développer le SmidgION, encore plus petit que le MinION, et qui peut fonctionner en se connectant tout simplement à un *smartphone*. Actuellement, les plateformes de séquençage ONT produisent des *reads* de taille pouvant atteindre les 200 000 paires de bases. En théorie, la longueur des *reads* produits ne dépend que de la librairie préparée ce qui a permis la production des *reads* de longueur supérieure à un million de paires de bases, appelées *ultra-long reads*. Cette technologie affiche aussi des taux d'erreur bien élevés compris entre 12 et 30%. Contrairement aux *reads* PacBio, les erreurs sont majoritairement des délétions et ont tendance à se localiser dans les régions contenant des homopolymères, du à la difficulté de déterminer précisément la longueur exacte de ces derniers quand ils dépassent les 5 ou 6 bases.

2.3.4 Notations en NGS

Bien que le séquençage de nouvelle génération respecte certains protocoles standards, il peut également être personnalisé pour répondre aux besoins de recherche individuels, y compris des facteurs tels que les questions spécifiques auxquelles on cherche à répondre, la profondeur du séquençage, la taille des *reads* et le type du séquençage (*single-end* ou *paired-end*). Lors du séquençage, il est possible de spécifier le nombre de paires de bases lues à la fois. Par exemple, une lecture peut être constituée de 50 paires de bases, 100 paires de bases ou plus. Des lectures plus longues peuvent fournir des informations plus fiables sur les emplacements relatifs de paires de bases spécifiques. C'est la longueur des *reads* ou bien *read length*. La profondeur ou la couverture est une mesure du nombre de fois qu'un site génomique spécifique est séquencé au cours d'un cycle de séquençage. Dans le séquençage d'exome, par exemple, la cible peut avoir une couverture de 60X, ce qui signifie qu'en moyenne, chaque base ciblée est séquencée 60 fois. Par contre, cela ne signifie pas que chaque base ciblée est séquencée 60 fois ; certains segments peuvent être lus 100 fois ou plus, tandis que d'autres ne peuvent être lus qu'une ou deux fois, ou pas du tout. Logiquement, plus le nombre de séquences d'une base est élevé, meilleure est la qualité des données mais plus coûteuse devient l'expérience. Dans la suite du manuscrit, nous ne nous intéresserons qu'aux *reads* de deuxième génération (produits particulièrement par la plateforme Illumina), qualifiés de *reads* courts. Le mot-valise indel sera utilisé pour faire référence aux erreurs d'insertions et de délétions.

```
>G4:B7UI2:1:1203:56387:1476/1
CAGACTTACGGCGGCGCTAATCGCTGCGATATTAACTGTATAACAGCCAA

>G4:B7UI2:1:1203:56387:1476/2
ACGATGATATGGCAGATTTATTAATAAAAAAACCAGTGAGTGAGTTAATC
```

FIGURE 2.11 – Un exemple d'un fichier FASTA comprenant deux *reads* pairés (l'appariement est indiqué par le «/1» et le «/2» à la fin de chaque en-tête). Chaque *read* est décrit sur deux lignes.

2.3.5 Simulation des données

Des outils sont développés afin de produire des *reads* artificiellement, sans avoir besoin de séquencer un vrai échantillon d'ADN. Ce sont des simulateurs de *reads* et ils permettent à l'utilisateur de contrôler les différents aspects d'une expérience de séquençage NGS telles que la profondeur et la longueur des *reads*. De plus, ces outils sont souvent utilisés pour introduire des erreurs dans les *reads* et contrôler le type et le taux d'erreur ajoutée, ceci en se basant sur un génome de référence. Outre leur utilisation pour évaluer l'impact de ces propriétés sur les résultats d'analyse, les données simulées sont particulièrement utiles pour guider le développement de nouveaux outils : elles permettent d'identifier les limitations des différents logiciels ainsi que de les évaluer sans biais en les comparant aux autres programmes. Différents simulateurs existent ainsi, afin de simuler le plus précisément possible le comportement des différentes technologies de séquençage. En ce qui concerne les technologies de troisième génération, d'une part, on distingue SimLORD [15] et NPBSS [16] assurant la simulation des *reads* PacBio, et d'autre part, on note Deep Simulator [17] et NanoSim [18] s'occupant des *reads* ONT. Plusieurs simulateurs existent aussi pour simuler des *reads* courts dont ART [19] permettant de simuler des *reads* ABI/SOLiD, des *reads* Roche/454 et des *reads* Illumina. De plus, on note CuReSim [20] qui permet de simuler des *reads* de toutes les technologies de deuxième génération (Roche/454, ABI/SOLiD, Illumina et Ion Torrent). Un troisième simulateur, MetaSim [21] se distingue des deux précédents par sa capacité de simuler des *reads* Sanger ainsi que des *reads* Illumina et Roche/454.

2.3.6 Format des données

Les séquenceurs NGS produisent les données sous forme de fichiers texte de différents formats. Ces fichiers stockent principalement les séquences nucléotidiques de l'échantillon d'ADN analysé et sont le plus souvent sous format FASTA ou FASTQ. Le format FASTA ne stocke que deux informations indispensables pour la reconnaissance de la séquence : un en-tête (ou *header*), contenant des informations relatives à la séquence formant une sorte d'identifiant unique à la séquence et la séquence du fragment. Le *header* commence généralement par le caractère «>» et est décrit sur une seule ligne. La séquence nucléotidique est sous forme d'une chaîne de caractères sur l'alphabet d'ADN et peut s'étendre sur plusieurs lignes. Un exemple de fichier au format FASTA est décrit dans la Figure 2.11.

Le format FASTQ, quant à lui, stocke une information supplémentaire pour chaque séquence. Un read est décrit sur quatre lignes et est identifié par un *header*, sa séquence nucléotidique et une chaîne de caractères décrivant la qualité du séquençage de chaque base. À la différence du *header* trouvé dans un FASTA, le premier caractère des *headers*

```
@G4:B7UI2:1:1203:56387:1476/1
CAGACTTACGGCGGCGC<b>T</b>AATCGCTGCGATATTAACTGTATAACAGCCAA
+
$!GD28D@3D882GDB3<b>B</b>D237DG23HD7H293HD92HD3D7G2G7D23D

@G4:B7UI2:1:1203:56387:1476/2
ACGATGATATGGCAGATTTATTAATAAAAAAACCAGTGAGTGAGTTAATC
+
;?98HG7938H8#H98H8H8JJAHJHJ9GD8D8ED993HJHGH3H52H99
```

FIGURE 2.12 – Un exemple d'un fichier FASTQ comprenant deux *reads* pairés (l'appariement est indiqué par le «/1» et le «/2» à la fin de chaque entête). Chaque *read* est décrit sur quatre lignes. Les scores Phred du premier et du deuxième *read* sont indiqués au niveau de la quatrième et la huitième ligne respectivement. Par exemple, la lettre T, représentée en gras dans la séquence nucléotidique (deuxième ligne) lui est attribuée le caractère «B». Ce caractère correspond à un score de qualité Phred de 32 et donc à une probabilité d'erreur d'identification de 0,00063.

FASTQ est un «@». Le *header* est suivi par la séquence au niveau de la deuxième ligne. La troisième ligne comporte le caractère «+» permettant de séparer la séquence de la quatrième ligne comportant les scores de qualité. En effet, ce score de qualité, appelé score Phred [22, 23], est calculé pour chaque base de la séquence. Le score Phred, noté Q, est stocké sous forme d'un encodage ASCII et sert à estimer la probabilité P de commettre une erreur lors de l'identification d'une base. En fait, ces deux valeurs sont interdépendantes, et respectent les équations $Q = -10log_{10}P$ et $P = 10^{\frac{-Q}{10}}$. Un exemple de fichier au format FASTQ est illustré dans la Figure 2.12.

2.3.7 Récapitulatif

La Table 2.1 résume les technologies et les plateformes de séquençage les plus répandues et les plus utilisées. Elle décrit notamment la longueur, le taux d'erreur et le nombre de *reads* générés par ces plateformes ainsi que le débit, le coût, l'année de sortie et le temps nécessaire au séquençage de chacune d'elles.

2.4 Problématiques

Aujourd'hui, suite à la grande diffusion du séquençage ainsi que la grande quantité de données qu'il génère, on a pu réaliser des avancées majeures pour répondre à plusieurs questions dans le domaine de la biologie. Tout d'abord, les génomes de référence ne sont pas disponibles pour toutes les espèces et donc une autre problématique liée à cet aspect est l'assemblage. En fait, les *reads* générés même par les séquenceurs de troisième génération sont plus courts que les chromosomes dont ils proviennent pour la plupart des espèces. Ainsi, l'assemblage est l'étape qui sert à fusionner les *reads* obtenus afin de reconstruire les chromosomes originaux et éventuellement déterminer le génome de référence des espèces. Ensuite, on note la recherche de similarités entre les séquences et leur alignement afin de trouver des mutations somatiques appelées SNV, des variations

Technologie	Plateforme	Année	Taille (pb)	Reads	Temps	Débit (Go)	Erreur (%)	Prix ($)	Prix/Go ($)
Première génération									
Sanger	-	1997	500-600	-	-	-	0,001	-	-
Maxam-Gilbert	-	1997	500-600	-	-	-	0,001	-	-
Deuxième génération									
454	GS Junior	2010	400-600	0,1 M	10 h	0,035	1	108 000	40 000
454	GS Junior+	2014	700-1000	0,1 M	18 h	0,07	1	108 000	19 500
454	GS FLX Titanium XLR70	2009	450-600	1 M	18 h	0,45	1	450 000	15 500
454	GS FLX Titanium XL+	2011	700-1000	1 M	23 h	0,7	1	450 000	9 500
Illumina	Genome Analyzer	2006	100	320 M	11 j	600	1	250 000	400
Illumina	MiniSeq (Débit moyen)	2013	150	14 - 16 M	17 h	2,1 - 2,4	1	50 000	200 - 300
Illumina	MiniSeq (Haut débit)	2013	150	44 - 50 M	24 h	6,6 - 7,5	1	50 000	200 - 300
Illumina	MiSeq v2	2011	250	24 - 30 M	39 h	7,5 - 8,5	0,1	99 000	142
Illumina	MiSeq v3	2011	300	44 - 50 M	21 - 56 h	13,2 - 15	0,1	99 000	110
Illumina	NextSeq 500/550 (Débit moyen)	2014	150	260 M	26 h	32 - 40	1	250 000	40
Illumina	NextSeq 500/550 (Haut débit)	2014	150	800 M	29 h	100 - 120	1	250 000	33
Illumina	HiSeq2500 v2	2012	250	600 M	60 h	125 - 160	0,1	690 000	40
Illumina	HiSeq2500 v3	2012	100	3 G	11 j	270 - 300	0,1	690 000	45
Illumina	HiSeq2500 v4	2012	125	4 G	6 j	450 - 500	0,1	690 000	30
Illumina	HiSeq4000	2015	150	2,5 G	1 - 3,5 j	650 - 750	0,1	900 000	22
Illumina	HiSeq X	2016	150	2,6 - 3 G	< 3 j	800 - 900	0,1	1 000 000	7
SOLiD	5500 Wilfire	2011	50-75	700 M	6 j	160	0,1	349 000	130
SOLiD	5500xl	2013	50-75	1,4 G	10 j	320	0,1	595 000	70
Ion Torrent	PGM 314	2011	200	400 - 550 K	3,7 h	0,06 - 0,1	1	49 000	25 - 1000
Ion Torrent	PGM 316	2011	400	2 - 3 M	4,9 h	0,6 - 1	1	49 000	700 - 1000
Ion Torrent	PGM 318	2013	400	4 - 5,5 M	7,3 h	1 - 2	1	49 000	450 - 800
Ion Torrent	Proton	2012	200	60 - 80 M	2 - 4 h	10	1	224 000	80
Ion Torrent	S5 520	2015	400	3 - 5 M	4 h	1,2 - 2	1	65 000	1200 - 2400
Ion Torrent	S5 530	2015	400	15 - 20 M	4 h	6 - 8	1	65 000	475 - 950
Ion Torrent	S5 540	2015	200	60 - 800 M	2,5 h	10 - 15	1	65 000	300
Troisième génération									
PacBio	RS II	2013	20 K	55 K	4 h	0,5 - 1	10 - 15	695 000	1 000
PacBio	Sequel	2016	8 - 12 K	350 K	0,5 - 6 h	3,5 - 7	10 - 15	350 000	N/A
ONT	MinION	2014	10 K - 1 M	> 100 000	< 48 h	1,5	12 - 30	1 000	750
ONT	PromethION	2014	10 K - 1 M	N/A	N/A	4 000	12 - 30	75 000	N/A

TABLE 2.1 – Résumé des caractéristiques principales des différentes technologies de séquençage. Tableau adapté de [14].

structurales appelées CNV ou encore du polymorphisme ou SNP (*Single Nucleotide Polymorphism*). Ceci est possible lors d'une étape appelée *Variant Calling*. Lors de cette étape, les *reads* alignés au génome de référence sont comparés fragment par fragment ou base

par base pour détecter et caractériser les modifications génétiques présentes. Finalement, toutes les technologies de séquençage présentent un taux d'erreur assez variable qui peut aller de 0,1 à 30% selon la plateforme utilisée. La détection et la correction potentielle de ces erreurs de séquençage est une problématique essentielle surtout lorsqu'elles peuvent être confondues avec d'autres mutations présentes à de faibles fréquences. Cette section présente plus en détail ces problématiques, qui seront abordées dans ce manuscrit, et plus particulièrement l'alignement de séquences, le variant calling et la correction d'erreurs, constituant le cœur des travaux menés durant cette thèse.

2.4.1 Correction des *reads*

Les erreurs de séquençage présentes dans les *reads* rendent l'analyse bioinformatique qui suit plus compliquée. En fait, ces erreurs peuvent, d'une part, augmenter le temps nécessaire pour l'alignement des *reads* et d'autre part, causer un mauvais alignement de ces derniers. De plus, même si ces erreurs sont présentes à faible fréquence, leur présence reste problématique dans les domaines de recherche de variants rares. Lors de l'assemblage également, la présence de telles erreurs est problématique pouvant mener à un résultat avec des contigs plus courts et plus nombreux. Pour cela, dans la plupart des expériences de séquençage de NGS, la première étape vise à essayer de corriger le plus d'erreurs possible dans les données. Plusieurs méthodologies ont été développées afin de corriger rapidement et efficacement un maximum d'erreurs dans les *reads*. Ces méthodologies diffèrent largement selon le taux et le type d'erreurs introduites et donc selon la technologie de séquençage utilisée. Les *reads* courts ont un taux d'erreur relativement faible (environ 1%) avec une majorité de substitution alors que les *reads* longs présentent une majorité d'indels avec des fréquences pouvant atteindre les 30%. Ainsi, de nombreux outils ont dû être développés implémentant des méthodes de correction différentes et spécifiques à un type de *read*, ou encore, à une seule plateforme.

Tout d'abord, pour les *reads* courts, quatre méthodes différentes existent pour les corriger. La première est appelée *k-mer spectrum* et repose sur une analyse de la fréquence des *k*-mers des *reads*. Elle part du principe qu'au sein des *reads*, les *k*-mers comportant des erreurs doivent théoriquement être moins observés que les *k*-mers génomiques. Après avoir fixé un seuil, on peut séparer les *k*-mers entre *k*-mers faibles et *k*-mers solides. Les *k*-mers faibles sont ceux représentés un nombre de fois inférieur au seuil fixé (et donc comportant des erreurs) et les *k*-mers solides sont ceux observés au-delà de ce seuil. Ces derniers serviront par la suite à corriger les *k*-mers faibles. De nombreux outils se basent sur cette approche comme BLESS [24] et Quake [25]. La deuxième méthode est basée sur la *k-mer spectrum* mais au lieu de fixer un *k*, elle utilise une table de suffixes pour pouvoir traiter différentes valeurs de *k* en même temps. Les outils HiTEC [26] et HybridSHREC [27] appliquent cette approche dans leur *workflow*. La troisième méthode se sert d'un alignement multiple pour corriger les *reads*. Ainsi, en premier lieu, des alignements multiples sont établis entre les *reads* et une séquence consensus en est déduite pour chaque *read*. Cette approche est implémentée dans les outils Coral [28] et ECHO [29] par exemple. Finalement, la quatrième approche cherche à établir des modèles de Markov cachés HMM (*Hidden Markov Model*) entre les *reads* afin d'en déterminer une séquence consensus pour chaque *read*. Cette approche est principalement appliquée dans le correcteur PREMIER Turbo [30].

En ce qui concerne les *reads* longs, deux stratégies essentielles sont souvent utilisées : la correction hybride et l'auto-correction. La correction hybride se sert des *reads* courts pour corriger les *reads* longs. Elle profite de la meilleure qualité des *reads* courts (leur taux d'erreur est faible) ainsi que de leur profondeur de couverture généralement supérieure à celle des *reads* longs. Des outils comme LSCplus [31] et HECIL [32] se basent sur

FIGURE 2.13 – Alignement de deux séquences S1 = GTAGTAC et S2 = GCACGTC. Les positions 0, 2, 4, 5 et 7 (en bleu) représentent des correspondances, ou *matches*, entre les séquences. La position 1 représente une substitution entre les séquences. La position 3 représente une délétion dans S1, ou l'insertion d'un C dans S2. La position 6 représente une insertion d'un A dans S1 ou une délétion dans S2.

cette méthode. L'auto-correction, quant à elle, se base uniquement sur les informations extraites des séquences des *reads* longs mais nécessite des profondeurs de séquençage plus importantes ($\geq$ 30x). Cette approche est utilisée dans des outils tels que MECAT [33] et Daccord [34].

La correction des *reads* inclut aussi la correction des UMI dans les *reads*. En effet, la grande utilité des UMI réside dans leur capacité de permettre le regroupement des *reads* originaires d'un même fragment initial. Cependant, une erreur de séquençage touchant la séquence d'UMI d'un *read* conduira à l'apparition d'un nouvel UMI unique et menant ainsi à une surestimation du nombre de fragments d'ADN initialement séquencés. De nombreux outils pour corriger les erreurs spécifiques des UMI existent dont UMI-tools [35]. La correction des UMI sera davantage détaillée dans les chapitres suivants, constituant le cœur des travaux menés durant cette thèse.

2.4.2 Alignement

Aligner deux séquences consiste à trouver le nombre minimum d'opérations d'édition (insertion, substitution ou délétion) permettant d'obtenir l'une à partir de l'autre. L'alignement des séquences est une étape primordiale pour toute analyse bioinformatique permettant par exemple de remonter aux ancêtres communs de différentes espèces en détectant des similarités ou des gènes identiques au sein de leur génome. On distingue deux méthodes d'alignement différentes : l'alignement local visant à trouver des régions assez similaires entre deux ou plusieurs séquences différentes, et l'alignement global qui détermine la similarité entre les séquences sur la totalité de leur longueur. Dans les deux cas, afin de quantifier la similarité entre les séquences, un score est calculé. La Figure 2.13 présente un exemple d'alignement entre deux séquences.

L'alignement est effectué en utilisant une matrice à deux dimensions, chaque séquence étant représentée sur une dimension. En se basant sur la programmation dynamique, le premier algorithme permettant d'effectuer un alignement local a été décrit par Smith et Waterman [36] alors que Needleman et Wunsch ont développé le premier algorithme pour réaliser un alignement global [37]. Dans les deux cas, le résultat de l'alignement est obtenu en temps quadratique $\mathcal{O}(MN)$, M et N représentant la longueur des séquences alignées. Depuis leur description initiale, ces algorithmes ont subi des optimisations visant à réduire le nombre d'opérations nécessaires les rendant ainsi plus efficaces (par exemple, l'algorithme d'alignement global initialement décrit par Needleman et Wunsch produisait les résultats en temps cubique mais plusieurs améliorations lui ont été apportées pour passer finalement à un temps quadratique). Cependant, ils restent

très coûteux en pratique, surtout dans les domaines d'application où les séquences sont divergentes et longues ainsi que lorsque des alignements multiples sont indispensables.

En effet, afin de réduire le temps de calcul, les outils d'alignement, appelés aligneurs ou *mappeurs*, utilisent donc des méthodes heuristiques au sein des algorithmes d'alignement implémentés. Les termes alignement et *mapping* sont en réalité différents. Le *mapping* ne fournit que l'information concernant les correspondances, ou *matches* (nombre et position) entre les séquences alors qu'un alignement calcule un score décrivant la similarité entre les séquences ainsi que les opérations d'édition nécessaires pour passer d'une séquence à l'autre. Malgré cette différence, nous utiliserons dans la suite le terme alignement pour parler de la comparaison des séquences en général. En addition, vu que certaines technologies, notamment Illumina, ont tendance à effectuer des erreurs de séquençage au niveau des deux extrémités des *reads*, certains aligneurs ont été conçus de façon à appliquer moins de contraintes dans ces régions en appliquant un procédé appelé, le *clipping*.

Ces méthodes, afin d'être plus efficaces, ont souvent recours à une structure d'indexation construite à partir de la cible de l'alignement, appelée *target*, et dans laquelle la recherche de la séquence, désignée par *query*, s'effectuera. La *query* est en général la séquence d'un *read* déterminé alors que la *target* peut être la séquence d'un autre *read*, un ensemble de *reads* ou même un génome de référence complet. En effet, une méthode appelée *seed and extract* est souvent utilisée dans ce type d'applications. Tout d'abord, on essaie d'identifier des régions appelées graines, ou *seeds* qui sont des sous-séquences fortement similaires retrouvées dans les deux séquences à aligner. En général, les *seeds* sont soit des chaînes de *k*-mers partagées entre les séquences, soit des *Maximal Exact Matches* (MEM) qui sont des régions identiques entre les séquences impossibles d'être étendues ni à gauche ni à droite. Finalement, on se sert des méthodes basées sur la programmation dynamique pour étendre l'alignement à partir des *seeds*.

Plusieurs outils d'alignement existent aujourd'hui dont on cite notamment BLAST [38], Bowtie [39], Bowtie2 [40] ainsi que BWA [41] et ses dérivés BWA-SW [42] et BWA-MEM [43]. D'autres outils sont développés spécifiquement pour l'alignement des *reads* de troisième génération comme BLASR [44], DALIGNER [45] ou encore Minimap [46] et Minimap2 [47]. Ces derniers prennent en compte la longueur supérieure ainsi que le taux d'erreur plus élevé de ces *reads* et donc utilisent des structures d'indexation et des types de graines différents. Plusieurs formats peuvent être utilisés pour représenter les alignements produits par ces aligneurs dont on distingue principalement les formats BAM (*Binary Alignment/Map*) et SAM (*Sequence Alignment/Map)* [48]. Le format BAM représente les alignements en version binaire alors que le format SAM est sa version équivalente mais en format texte. Le format SAM est composé de onze colonnes obligatoires, séparées par des tabulations et contenant des informations sur les séquences alignées telles que le nombre et le type des opérations d'édition (code CIGAR), les positions de début et de fin de l'alignement ainsi que le sens de ce dernier. Le format SAM est décrit en détail dans la Table 2.2.

Un alignement n'est pas forcément limité à une comparaison entre deux séquences mais peut être aussi réalisé entre plusieurs séquences. Ce type d'alignement est connu sous le terme alignement multiple et il permet de comparer un ensemble de séquences, deux à deux. L'utilisation principale de ce type d'alignement est dans le but d'extraire une séquence consensus à partir de plusieurs séquences. Pour ce faire, à chaque position de l'alignement multiple, on choisit la base majoritaire, et donc la plus fréquemment observée, au sein de l'ensemble de séquences. Ce genre d'alignement est souvent représenté par une matrice de taille $M \times N$, N étant le nombre de séquences à aligner et M la longueur de la séquence consensus (donc de l'alignement).

Colonne	Description	Type
1	*Header* de séquence *query*	Chaîne de caractères
2	Drapeau décrivant l'alignement (orphelin, orientation, etc)	Entier
3	*Header* de séquence *target*	Chaîne de caractères
4	Position de début de l'alignement sur la séquence *target*	Entier
5	Qualité de l'alignement	Entier
6	Code CIGAR	Chaîne de caractères
7	*Header* du *mate*	Chaîne de caractères
8	Position d'alignement du *mate*	Entier
9	Longueur de la séquence *target*	Entier
10	Facteur de la séquence *query* alignée	Chaîne de caractères
11	Score de qualité Phred	Chaîne de caractères

TABLE 2.2 – Présentation des onze colonnes obligatoires d'un fichier du format SAM. Tableau adapté de [14].

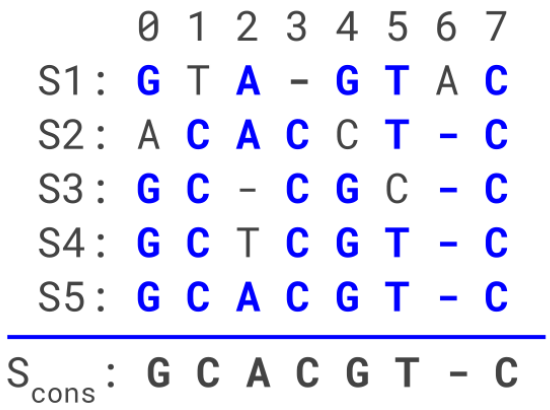


FIGURE 2.14 – Représentation par matrice de l'alignement multiple de 5 séquences et de la séquence consensus S_{cons}. La séquence consensus est ici obtenue par un vote majoritaire à chaque position. La base la plus représentée à chaque position est représentée en bleu. La séquence consensus ainsi obtenue est alors GCACGTC.

Finalement, il est complètement possible et fréquent d'utiliser un graphe orienté acyclique DAG (*Directed Acyclic Graph*) pour schématiser un alignement multiple. Comme son nom l'indique, un tel graphe est un graphe orienté qui ne comporte aucun circuit et, dans le cas des alignements multiples, aura les différentes bases des séquences représentées chacune par un sommet et des arcs, reliant les sommets entre eux, et représentant le nombre de fois qu'un passage d'une base spécifique à une autre a été observé au sein des différentes séquences. Un exemple d'alignement multiple, et de séquence consensus en résultant, est illustré en Figure 2.14, pour la représentation par matrice et en Figure 2.15 pour la représentation par DAG. Ces méthodes d'alignement multiple sont couramment utilisées par les algorithmes de correction d'erreurs de séquençage qui seront présentés dans la section 2.5.1 et expliqués en détail dans le Chapitre 4.

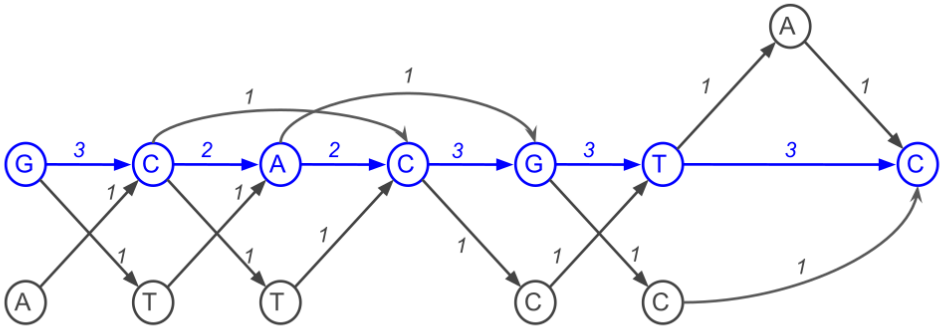


FIGURE 2.15 – Représentation par DAG de l'alignement multiple de 5 séquences et de la séquence consensus. La séquence consensus est ici obtenue en recherchant le chemin de poids maximal au sein du graphe. Ce chemin est reporté en bleu. La séquence consensus ainsi obtenue est alors GCACGTC.

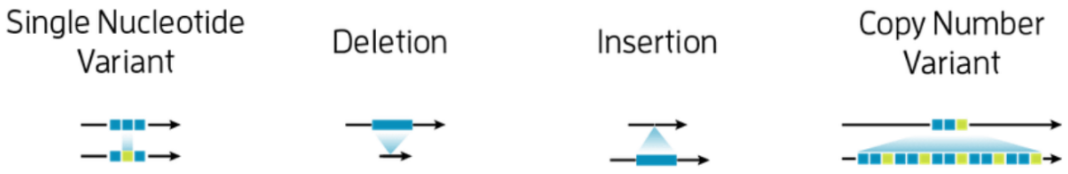


FIGURE 2.16 – Les différents types de variants détectés par les expériences de séquençage NGS.

2.4.3 Variant Calling

L'appel de variants, ou *Variant Calling* consiste à analyser une séquence d'échantillon, qui peut être une séquence de gène unique, un exome entier ou un génome entier, en la comparant à une séquence de référence. Les différences entre l'échantillon et la référence sont identifiées : ces dernières peuvent être des changements de base unique, tels que des SNP, SNV et des indels, ou encore des variants structuraux à plus grande échelle telles que les CNV (Figure 2.16).

La détection des variants que ce soit des substitutions ou de petites insertions/délétions (indels) représente une étape primordiale dans l'analyse des données de séquençage et constitue une partie très importante de l'analyse bioinformatique secondaire. Grâce au débit très élevé des nouveaux séquenceurs NGS, il est aujourd'hui possible de séquencer des exomes voire des génomes complets d'un grand nombre d'individus pour un prix relativement faible et un temps très court [49]. Pourtant, l'appel précis des variants reste difficile en raison d'un nombre de problèmes tels que les erreurs de séquençage et la couverture variable [50].

Les erreurs de séquençage introduites, créant des faux variants ou artefacts, ont souvent des fréquences très basses (inférieures à 3%) et donc sont facilement filtrées lors de la recherche des mutations à des fréquences élevées. Cependant, les nouvelles avancées dans le domaine du cancer ont mis en évidence la présence d'ADN tumoral dans le sang

[51]. Cet ADN est appelé *ctDNA* (*circulating tumor DNA*) et contient des mutations somatiques à de très faibles fréquences. Ces dernières peuvent être très facilement confondues avec les artefacts et donc mener à un taux élevé de faux positifs (variants détectés comme somatiques mais qui sont en réalité des erreurs de séquençage). Pour répondre à ces problèmes, de nombreuses stratégies ont été déployées notamment l'utilisation des UMI dans les expériences de séquençage, l'utilisation d'un échantillon normal apparié (*matched normal sample*) mais aussi l'implémentation d'un nombre de filtres complémentaires permettant de faire la différence entre vrai et faux variant. Un *variant caller* en général prend en entrée un fichier d'alignement BAM/SAM. Cependant, des *pipelines* complets ont été développés effectuant alignement, calibration et *variant calling* et donc partant d'un fichier FASTQ. Pour mieux repérer les artefacts, des *variant callers* nécessitent un échantillon normal apparié permettant de filtrer les artefacts en effectuant une comparaison entre un échantillon normal et un échantillon tumoral de la même personne. Ceci est effectué en se basant sur le principe que la plupart des erreurs introduites lors du séquençage ne sont pas aléatoires et donc devraient apparaître au même endroit pour les deux échantillons. Ainsi, seules les positions mutées dans l'échantillon tumoral seront considérées comme de vraies mutations somatiques. Cependant, l'obtention d'une biopsie normale de la même personne est dans la plupart des cas très compliquée et donc les *variant callers* proposent aussi un mode de détection sans se comparer à un *matched normal sample*. Ce mode de détection est certainement moins précis puisque la comparaison se fait contre le génome de référence et donc ne prend pas en compte la variabilité inter-individuelle de l'ADN dûe à la présence des SNP par exemple. D'autres stratégies sont utilisées pour contourner ce problème dont notamment l'utilisation des UMI qui sera détaillée dans le Chapitre 3.

Actuellement, un très grand nombre de *variant callers* existent : ils diffèrent principalement selon la loi statistique utilisée, les filtres complémentaires incorporés et l'implémentation d'une analyse UMI. De ces outils, on cite LoFreq [52] et MuTect [53] qui reposent sur une analyse de fréquence allélique et qui peuvent fonctionner sans ou avec un échantillon normal apparié. MutationSeq [54] quant à lui repose sur de l'apprentissage automatique et nécessite forcément un *matched normal sample* pour effectuer son analyse. De plus, les deux *variant callers* OutLyzer [55] et SiNVICT [56] sont conçus pour du séquençage profond (*Deep Sequencing*) et permettent la détection des variants à très faible fréquence. OutLyzer se base sur une estimation du bruit de fond par position alors que SiNVICT repose sur une approche probabiliste en appliquant un modèle de Poisson sur les *reads*. Finalement, les outils implémentant une analyse des UMI sont peu nombreux vu que cette utilisation est encore très récente dans le monde du séquençage NGS. On distingue principalement MAGERI [57] et DeepSNVMiner [58] pouvant détecter des mutations à des fréquences minimales de 0,1%. Tous ces outils produisent leurs résultats sous forme d'un fichier au format VCF (*Variant Call Format*), un fichier texte formé par au moins huit colonnes obligatoires, séparées par des tabulations et contenant des informations sur les variants détectés telles que leur fréquence, leur qualité et leur type. Le format VCF est décrit en détail dans la Table 2.3.

Les *copy number variations* (CNV) sont des variations génétiques structurales pouvant représenter des réarrangements, des gains (duplications) ou des pertes (délétions) d'un segment, d'un gène ou même d'un chromosome entier. Un exemple est donné dans la Figure 2.18. Plusieurs mécanismes peuvent expliquer leur apparition : cependant aujourd'hui, ces mécanismes restent très complexes et mal compris. Les CNV sont estimés à participer de 18% à la variabilité héritée au niveau de l'expression génique. La détection des CNV par séquençage nécessite des approches particulières, différentes de la détection des mutations qui s'intéresse uniquement à la séquence des *reads*. Il existe à l'heure actuelle plusieurs stratégies pour procéder à la détection de CNV à partir de données

Colonne	Nom	Description	Type
1	CHROM	Identifiant du chromosome	Chaîne de caractères
2	POS	Position du variant	Entier
3	ID	Identifiant du variant	Chaîne de caractères
4	REF	La base de référence	Entier
5	ALT	La base alternative	Entier
6	QUAL	Score de qualité du variant	Entier
7	FILTER	Les filtres que le variant a passé	Chaîne de caractères
8	INFO	Une liste des paires <clé>=<valeur> décrivant le variant	Entier

TABLE 2.3 – Présentation des huit colonnes obligatoires d'un fichier du format VCF.

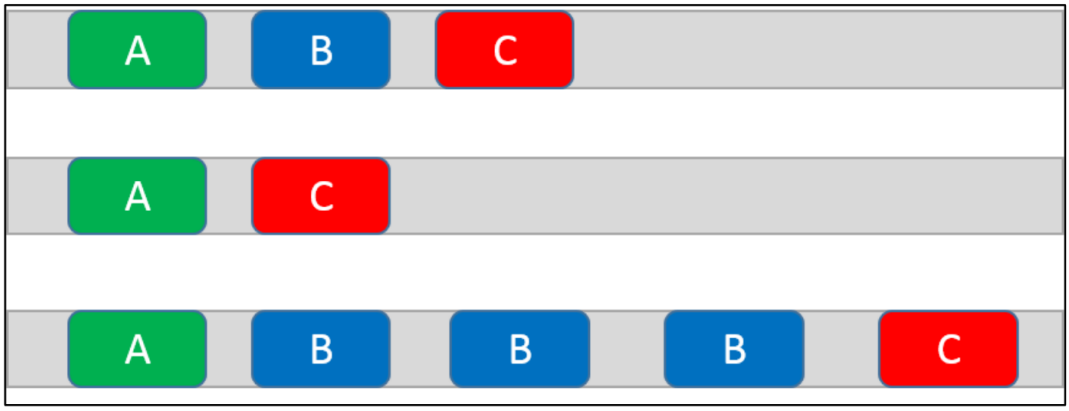


FIGURE 2.17 – Les différents types de CNV. La première séquence présente un fragment d'ADN normal contenant trois segments A, B et C. La deuxième séquence représente une délétion du segment B et la dernière séquence met en évidence le cas d'une duplication du segment B présent en 3 exemplaires. Figure adaptée de [62].

NGS ce qui a conduit au développement de nombreux outils tels que ONCOCNV [59], CONTRA [60] ou encore VarScan 2 [61].

La détection des CNV diffère de la détection des SNV, cette dernière ne s'intéressant principalement qu'à la séquence des *reads*. Pour cela, des méthodes particulières ont dû être développées pour la détection des variants structuraux à partir des données de NGS. La première est la méthode de l'alignement des paires de *reads* [63] : elle repose sur la distance séparant un *read* de son *mate* lors de l'alignement et donc ne peut fonctionner qu'avec des *reads* pairés. Une autre approche, appelée méthode des *reads* tronqués, repose sur l'identification du point de cassure entre un *read* et son *mate* dans le cas où les deux *reads* ne s'alignent pas correctement sur le génome [64]. Cette méthode présente les mêmes limitations que celle au-dessus en plus de la nécessité que les *reads* aient une longueur assez importante. Ces deux approches ne prennent pas en compte les informations concernant la position des *reads* et la couverture d'une région déterminée. Une approche visant à contourner cette limitation est la méthode de la profondeur des *reads* qui se sert de cette information pour détecter les CNV. Elle se base sur le principe que la profondeur moyenne d'un segment déterminé est proportionnelle à la quantité d'ADN séquencé et donc moins la quantité d'ADN présente, moins la profondeur. D'autres méthodes existent également telles que la méthode de l'assemblage *de novo* et la méthode

par le séquençage de longs fragments qui ne seront pas présentées ici vu que les travaux principaux effectués pendant cette thèse se concentrent sur la détection des CNV en utilisant les UMI.

2.5 Synthèse

Dans ce chapitre, les caractéristiques principales de l'ADN et de l'ARN ont été présentées d'un point de vue biochimique ainsi que leur composition et leur structure. De plus, nous avons détaillé le mécanisme permettant le passage de l'ADN à l'ARN, appelé transcription, et celui permettant la formation des protéines à partir de l'ARN appelé traduction. Ensuite, nous avons introduit les différentes technologies de séquençage en décrivant les principes biochimiques sur lesquels elles sont fondées ainsi que les caractéristiques des données produites (longueur des *reads*, taux d'erreur, ...) par chacune commençant par les technologies de première génération, passant par les plateformes de deuxième génération et finissant par les séquenceurs de troisième génération. Enfin, nous avons présenté trois problématiques essentielles de l'analyse des données NGS. Tout d'abord, la correction des erreurs dans les *reads* visant à diminuer leur impact sur la qualité des résultats d'analyse. La deuxième est l'alignement des séquences, étape primordiale pour toute analyse des données et la troisième étant le *variant calling* permettant de détecter des nouveaux variants de type SNV ou CNV afin de déterminer leur rôle dans différentes maladies telles que les cancer.

Chapitre 3

Utilisation des UMI en NGS

3.1 Introduction

Comme décrites dans le Chapitre 2, les techniques de séquençage de deuxième génération permettent de séquencer de l'ADN en produisant des *reads* courts, beaucoup moins bruités que les *reads* longs de troisième génération. Néanmoins, les erreurs de séquençage rencontrées peuvent compliquer, voire fausser les résultats de certaines applications de NGS, notamment la quantification et la recherche des variants rares dans lesquelles il faut être le plus précis possible. Pour cela, une nouvelle approche visant à étiqueter les fragments d'ADN par des barcodes moléculaires appelés UMI (*Unique Molecular Identifiers*) a permis de détecter et même corriger les erreurs dans les *reads* ainsi que de quantifier avec précision le nombre des molécules initiales d'ADN. Cette méthode a été utilisée dans plusieurs domaines comme le DNA-Seq, le RNA-Seq et aussi le scRNA-Seq (*single cell RNA-Sequencing*). Ceci a mené au développement de différents outils afin de profiter de l'information additionnelle apportée par la présence des UMI dans les *reads*.

Ce chapitre présente donc l'état de l'art de certaines applications où les UMI ont été efficacement utilisés pour améliorer les résultats obtenus ainsi que les principaux outils développés permettant l'extraction et la correction des UMI dans les *reads* mais aussi leur utilisation efficace pour détecter les variants somatiques avec plus de précision à des fréquences très basses.

3.2 Utilisation

Les UMI sont un type de *barcodes* moléculaires et donc, de courtes séquences utilisées pour marquer de manière unique chaque molécule dans une librairie de fragments, fournissant ainsi une correction d'erreur et une précision accrue lors du séquençage. Les UMI sont utilisées pour une large gamme d'applications de séquençage, surtout dans l'analyse de l'expression des gènes RNA-seq et d'autres méthodes de séquençage quantitatif dans lesquelles l'identification et la suppression des doublons de PCR est essentielle. En effet, dans tout scénario où la profondeur du séquençage est un facteur important, les doublons de PCR font augmenter à tort la couverture et, s'ils ne sont pas supprimés, peuvent donner l'illusion d'une confiance élevée. Les doublons de PCR sont expliqués dans les Figures 3.1 et 3.2. La Figure 3.1 montre 9 *reads* alignés sur un génome de référence à une certaine position dont 4 présentent un variant C. Ces 4 *reads* ont des coordonnées d'alignement différentes et donc représentent des *reads* provenant de fragments différents. Dans ce cas de figure, la confiance accordée à ce variant est assez élevée. D'une autre part, la Figure 3.2 présente un exemple reprenant la même position, le même nombre de *reads* qui y sont alignés et le même nombre de *reads* mutés. Cependant, les *reads* mutés ont tous les mêmes coordonnées d'alignement ce qui montre que les 4 *reads* proviennent de la duplication d'une même molécule initiale. Ces derniers, appelés doublons de PCR,

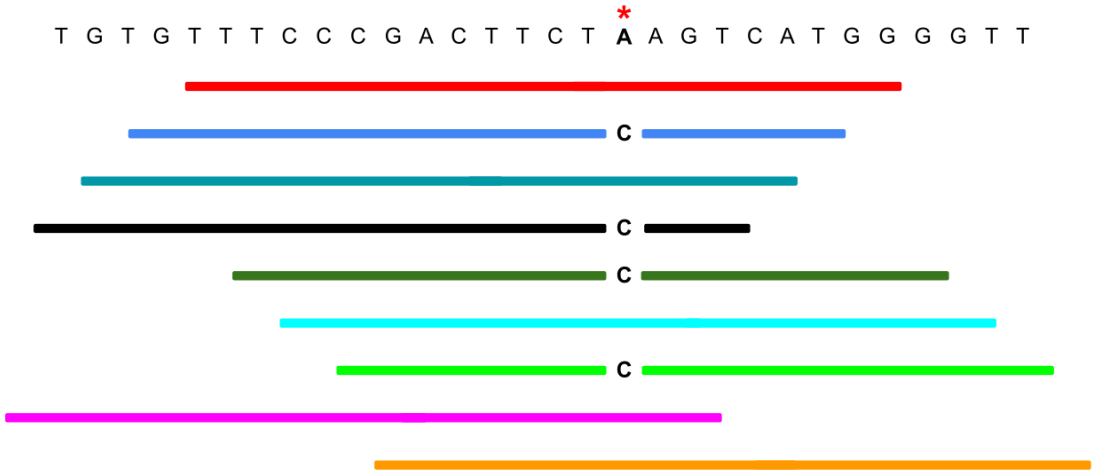


FIGURE 3.1 – Schéma montrant l'alignement de reads mutés (A>C) avec des coordonnées d'alignement différentes et donc issus de fragments différents.

conduisent souvent à une diminution de précision dans les analyses en surestimant le nombre de *reads* s'alignant à une certaine position, surtout dans le cas où un biais de PCR est présent lors de l'amplification. L'amplification par PCR est généralement une étape nécessaire dans le séquençage car elle augmente le nombre de fragments d'ADN en dupliquant chaque molécule plusieurs fois. En ligaturant une courte séquence UMI aléatoire sur chaque brin de fragment d'ADN avant l'amplification par PCR, les *reads* séquencés avec un même UMI peuvent être facilement identifiés comme des doublons PCR. En effet, chaque UMI aléatoire représente une étiquette unique pour chaque molécule d'ADN, ce qui signifie que tous les doublons de PCR de la même molécule d'ADN présenteront le même UMI. Les UMI sont utiles dans de nombreuses expériences car ils permettent de réduire des inexactitudes dans le nombre réel de molécules d'ADN/ARN avant une analyse plus en aval. Dans ce qui suit, nous allons présenter les différents domaines d'applications où les UMI ont été utilisés pour obtenir de meilleurs résultats.

3.2.1 DNA-Seq

3.2.1.1 Détection d'une trisomie 21 par caryotypage digital

La première utilisation des UMI a été réalisée en 2012 par Kivioja *et al.* [65]. Dans cette expérience, les auteurs ont prélevé l'ADN d'une femme normale ainsi que celui de son fils atteint du syndrome de Down caractérisé par la présence de trois chromosomes 21 au lieu de deux. Les deux échantillons d'ADN ont été mélangés (ratio 1 : 1), fragmentés et puis étiquetés pas des barcodes nucléotidiques uniques, les UMI (dans cette expérience, les UMI utilisés avaient une taille de 5 pb). Ensuite, l'ADN des deux personnes a été amplifié et séquencé de façon à obtenir des *reads* courts avec les UMI au début.

Les résultats sont présents dans la Figure 3.3. En comparant le nombre total des *reads*, la différence n'est pas claire au niveau du chromosome 21 entre l'échantillon mère-fils (Figure 3.3a) et le caryotype masculin normal (Figure 3.3b). En effet, une ambiguïté existe au niveau du comptage des *reads* s'alignant sur le chromosome 21 ce qui empêche de

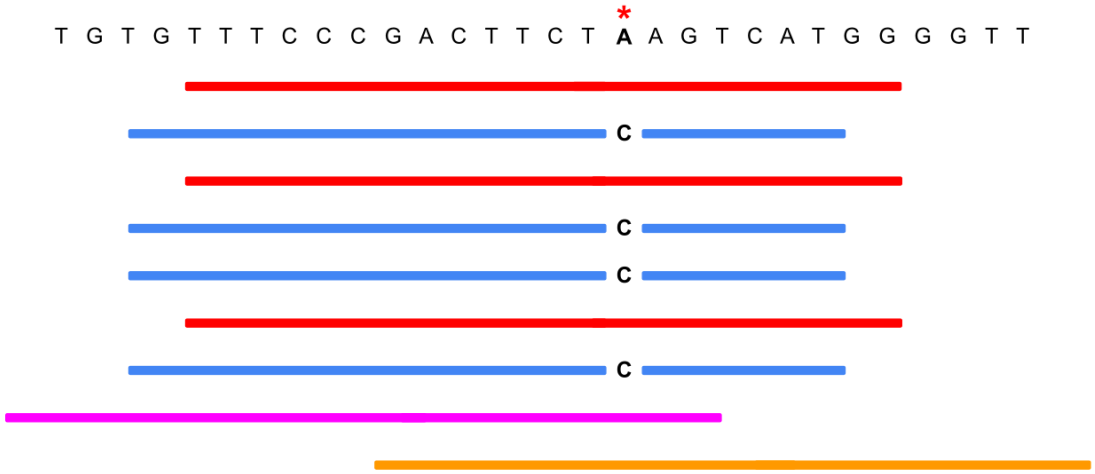


FIGURE 3.2 – Schéma montrant l'alignement de reads mutés (A>C) avec des coordonnées d'alignement identiques et donc présentant des doublons de PCR.

conclure précisément sur le nombre de copies présentes dans l'échantillon (a). En réanalysant les résultats et en tenant compte des UMI et non pas du nombre total des *reads*, ils ont pu quantifier le nombre de molécules initiales et donc présent dans le tube avant l'amplification.

La Figure 3.3c présente ces nouveaux résultats. En comptant les molécules initiales au lieu du nombre total des *reads*, et en le comparant à un génome humain féminin simulé *in silico*, l'ADN séquencé de l'échantillon mère-fils montre clairement un chromosome 21 en plus, mettant alors en évidence la trisomie 21 du fils. Ainsi, on peut conclure que, contrairement aux anciennes approches, la nouvelle méthode qui se base sur les UMI peut être utilisée pour estimer avec précision le nombre de molécules initiales sans forcément avoir à les observer.

3.2.1.2 Détection des mutations *de novo* dans du *cfDNA* (*cell-free DNA*)

Les auteurs de cette étude ont développé un nouveau système permettant l'amélioration de détection de nouvelles mutations somatiques à de très faibles fréquences dans l'ADN plasmatique, ou *cfDNA* [66]. Ce système est nommé NOIR (*Non-Overlapping Integrated Reads*) et comprend les deux étapes suivantes :

1. Une nouvelle méthode de séquençage ciblé qui ajoute des séquences de barcodes moléculaires, les UMI, par ligature d'adaptateur : cette méthode utilise l'amplification linéaire pour éliminer les erreurs introduites au cours des premiers cycles de PCR.
2. La surveillance et la suppression des barcodes erronés. Ce processus implique l'identification de molécules individuelles qui ont été séquencées et pour lesquelles le nombre de mutations a été absolument quantifié.

Ce système a été testé notamment sur une série de patients atteints d'un cancer gastrique et pour lesquels la mutation c.747G>C dans le gène *TP53* a été validée. L'ADN

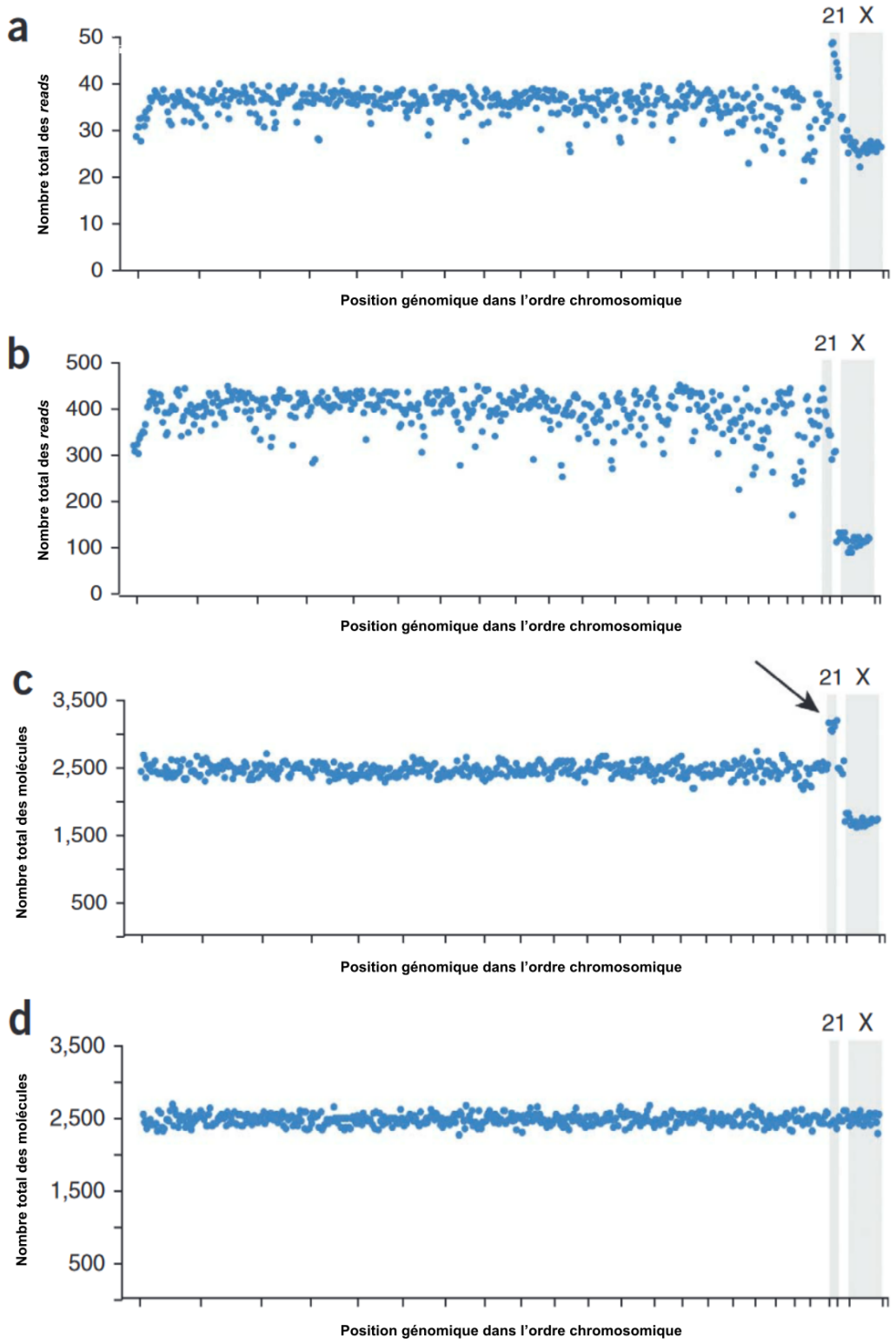


FIGURE 3.3 – Caryotypage digital en comptant le nombre absolu de molécules. (a) Caryotype numérique standard basé sur l'ADN génomique d'un garçon atteint de trisomie 21 et de sa mère, mixte 1 :1. (b) Caryotype numérique standard d'un échantillon d'un individu masculin avec un nombre normal de chromosomes. (c) Le même échantillon que dans (a) mais analysé par comptage UMI. La flèche met en évidence le nombre de copies uniformément élevé des régions du chromosome 21. (d) Échantillon simulé par échantillonnage aléatoire uniforme d'un génome humain normal féminin. Les chromosomes 21 et X sont indiqués par les deux zones ombrées à la fin. Figure adaptée de [65].

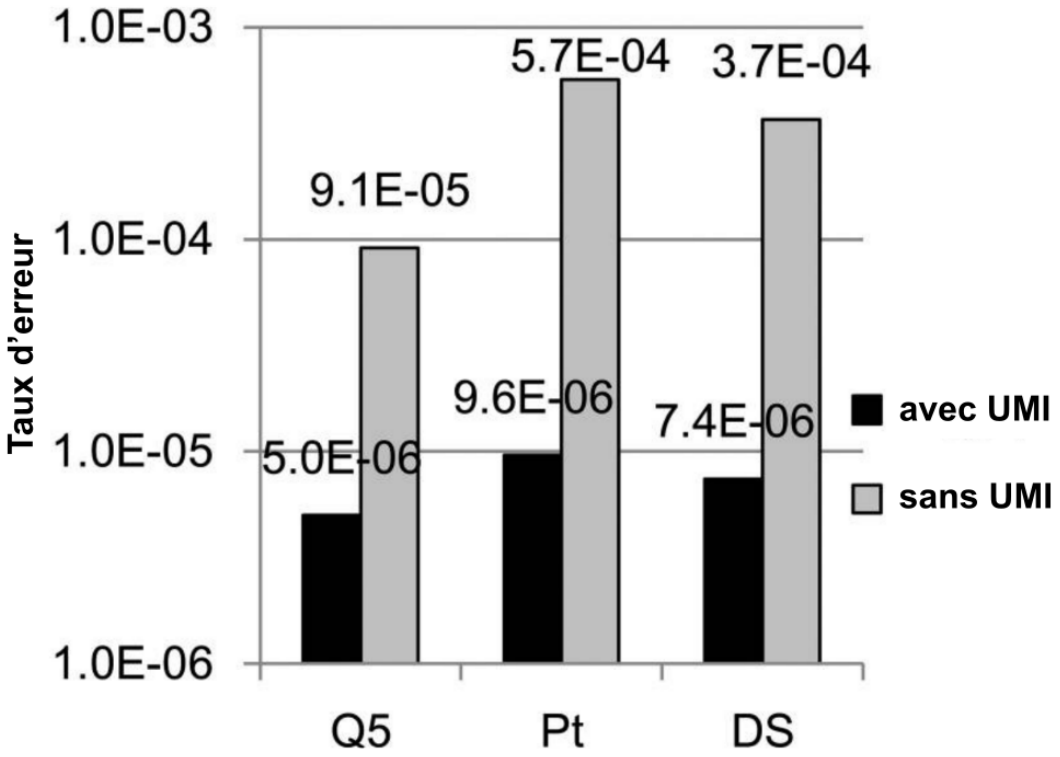


FIGURE 3.4 – Taux d'erreur de séquençage pour les régions cibles avec (noir) et sans UMI (grises). Q5, étiquetage simple brin avec l'ADN polymérase Q5 pour l'amplification par PCR ; Pt, étiquetage simple brin avec l'ADN polymérase Platinum Taq pour l'amplification par PCR ; DS (*double strand*), étiquetage double brin. Figure adaptée de [66].

extrait de ces patients a été séquencé en utilisant un Ion Proton Sequencer (Life Technologies). D'abord, pour démontrer la meilleure précision d'un séquençage avec UMI, les auteurs ont comparé les taux d'erreur obtenus avec trois ADN polymérases différentes : la Q5 qui fait de l'étiquetage simple brin, la Platinum Tag (Pt) faisant aussi de l'étiquetage simple brin et la *double strand* (DS) permettant un étiquetage double brin. Les comparaisons ont été réalisées pour chaque enzyme, avec et sans utilisation des UMI. La Figure 3.4 présente les résultats obtenus. Ces derniers montrent clairement que pour les trois ADN poylmérases, le taux d'erreur obtenu avec des UMI est significativement inférieur à celui obtenu sans leur utilisation. De plus, on note que la différence entre un étiquetage simple brin et un étiquetage double brin n'est pas significative, permettant ainsi de conclure qu'un étiquetage simple brin est suffisant pour améliorer les résultats obtenus.

Ensuite, les auteurs ont testé le système NOIR pour détecter une mutation d'un patient atteint d'un cancer gastrique. La tumeur principale présentait la mutation c.747G>C dans le gène *TP53*. Un séquençage avec et sans UMI a été effectué chez ce patient et le résultat est présenté dans la Figure 3.5. Dans cette figure, on voit clairement que l'utilisation des UMI lors du séquençage a permis d'éliminer la plupart, voire tous les faux positifs (erreurs de séquençage détectées comme variants par le *variant caller*) détectés lors d'un séquençage sans UMI. Le séquençage avec UMI détecte un seul variant seulement : la mutation c.747G>C.

Finalement, l'utilisation du système NOIR a permis aux auteurs de mettre en évidence des nouvelles mutations dans le gène *KRAS* chez des patients atteints d'un cancer pulmonaire. Trois groupes de personnes ont été séquencés : un premier groupe représentant

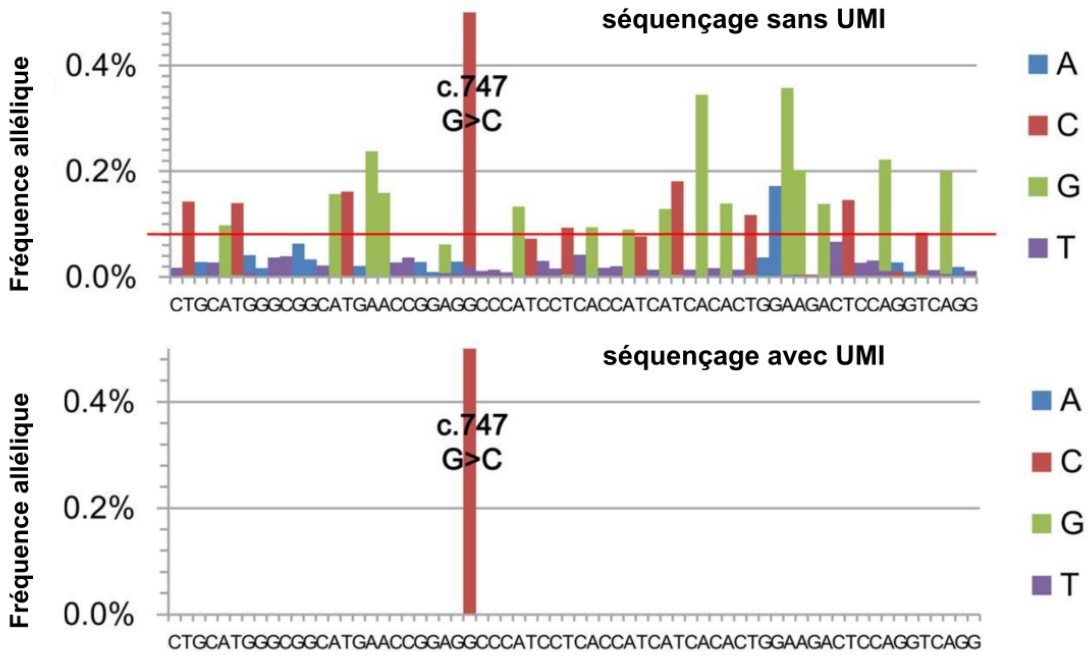


FIGURE 3.5 – Les variants détectés par séquençage profond avec et sans UMI. Le séquençage avec UMI ne détecte qu'un seul variant. Plus de 20 faux positifs ont été détectés en séquençage sans UMI. La ligne rouge indique le critère de détection des variants proposé par Couraud *et al*. [67]. Figure adaptée de [66].

le groupe normal (normal) ayant des leucocytes sains, un deuxième groupe atteint d'un cancer pulmonaire sans mutation du gène *EGFR* (Pwt) et un troisième groupe atteint d'un cancer pulmonaire avec mutation du gène *EGFR* (Pmt). Les résultats sont présentés dans la Figure 3.6. Dans les deux cas (Figure 3.6A, B), tous les variants présents au-dessous de la ligne grise sont des faux variants et tous ceux au-dessus de la ligne sont des vraies mutations. Dans la Figure 3.6A présentant les variants détectés en utilisant les UMI, aucun faux variant ne ressort pour les individus normaux, huit nouveaux variants sont détectés pour les patients Pwt et un nouveau variant chez le groupe Pmt. D'autre part, l'absence des UMI (Figure 3.6B) rend la détection des variants plus compliquée. Chez les individus normaux, cinq variants sont détectés (faux positifs), quatre variants de plus sont présents chez le groupe Pwt et un variant de plus est trouvé chez les patients Pmt. Ces variants sont très probablement des faux positifs démontrant ainsi l'efficacité de l'utilisation des UMI pour filtrer efficacement les faux positifs et pour détecter des nouvelles mutations avec précision.

3.2.1.3 Comparaison entre différents fabricants de kits NGS avec et sans UMI

Afin de définir les bonnes pratiques concernant le choix approprié pour un séquençage ciblé basé sur la PCR, les auteurs de cette étude [68] ont cherché à comparer les performances obtenues lors d'un séquençage NGS ciblé avec ou sans UMI, en utilisant les kits de quatre fabricants différents : Archer® Reveal ctDNA^{TM} 28, NEBNext Direct® Cancer HotSpot Panel, Nugen Ovation® Custom Target Enrichment System, Qiagen Human Comprehensive Cancer Panel (HCCP) et Qiagen Human Actionable Solid Tumor Panel (HASTP). Pour chaque kit, les *reads* bruts ont été analysés et séparés en *reads* finaux

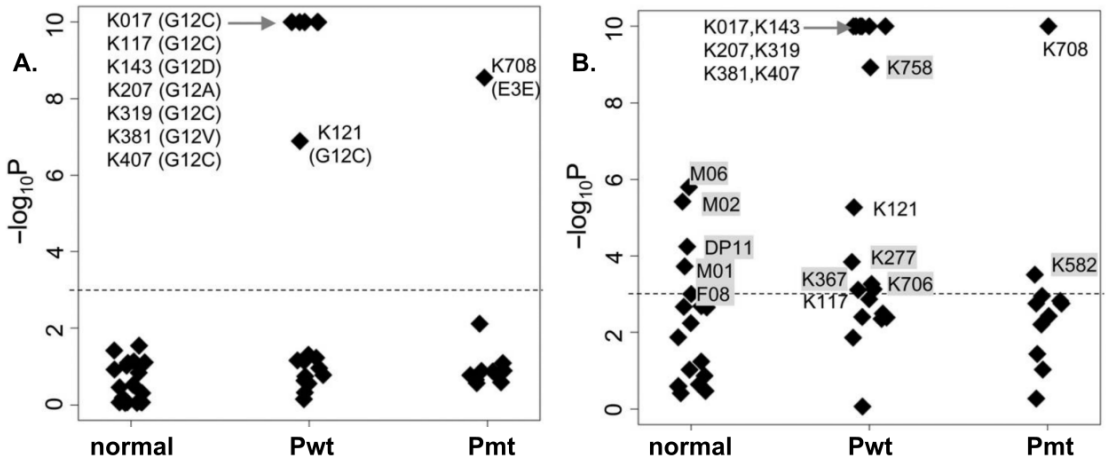


FIGURE 3.6 – Détection *de novo* des mutations *KRAS* dans du *cfDNA* prélevé chez des patients atteints d'un cancer du poumon. En ordonnée, la probabilité qu'un variant soit un faux positif, notée P. En abscisse : ADN normal provenant d'individus sains ; Pwt, patients négatifs à la mutation EGFR ; Pmt, patients porteurs d'une mutation EGFR. Le graphique A montre les résultats du système de séquençage NOIR. Le graphique B montre les résultats des *reads* sans l'utilisation des UMI. Les lignes pointillées indiquent le seuil de détection des variants (P = 0,001). Les variants faux positifs en B sont marqués d'un fond gris. Figure adaptée de [66].

et filtrés. Les *reads* finaux sont ceux qui s'alignent correctement sur les positions ciblées par le kit (*on-target*). Les *reads* filtrés représentent l'ensemble des séquences non alignées, les doublons de PCR et les séquences s'alignant sur des positions non ciblées par le kit (*off-target*). En effet, les doublons de PCR sont une gêne quotidienne dans le séquençage. Ils sont obtenus lors de l'amplification par PCR et peuvent représenter entre 30% et 70% des *reads* obtenus. Ce sont des copies identiques d'une même molécule initiale d'ADN et peuvent donc entraîner une sur ou sous-représentation d'un ou de plusieurs fragments d'ADN. La plupart des *pipelines* de séquençage recommandent de les supprimer ou au moins de les marquer avec des UMI.

En comparant les performances des cinq kits dans la Figure 3.7, on voit clairement que tous ont du mal à produire une grande quantité de *reads on-target* sans utilisation des UMI. Le kit Qiagen HASTP est le plus efficace avec 10,1% de *reads* alignés, un pourcentage relativement faible. Cependant, en utilisant les UMI, on voit que ce pourcentage passe à 52%. Même pour les autres kits, les résultats avec UMI ont montré une augmentation de *reads on-target* entre 1,7% et 41,9%. Cette augmentation du nombre de *reads* alignés est surtout due à une diminution du nombre des doublons de PCR puisqu'ils sont plus faciles à filtrer en utilisant les UMI.

La Figure 3.8 montre la variation du nombre de molécules uniques en fonction de la profondeur du séquençage pour les cinq kits, avec et sans UMI. De même, les deux kits Qiagen avec UMI obtiennent les meilleurs résultats et pour tous les kits, la version avec UMI présente un plus grand nombre de molécules uniques en la comparant avec la version sans UMI. Ces résultats mettent en évidence l'efficacité de l'utilisation des UMI pour identifier et supprimer les doublons de PCR dans les expériences de séquençage NGS ciblé basées sur de l'amplification par PCR.

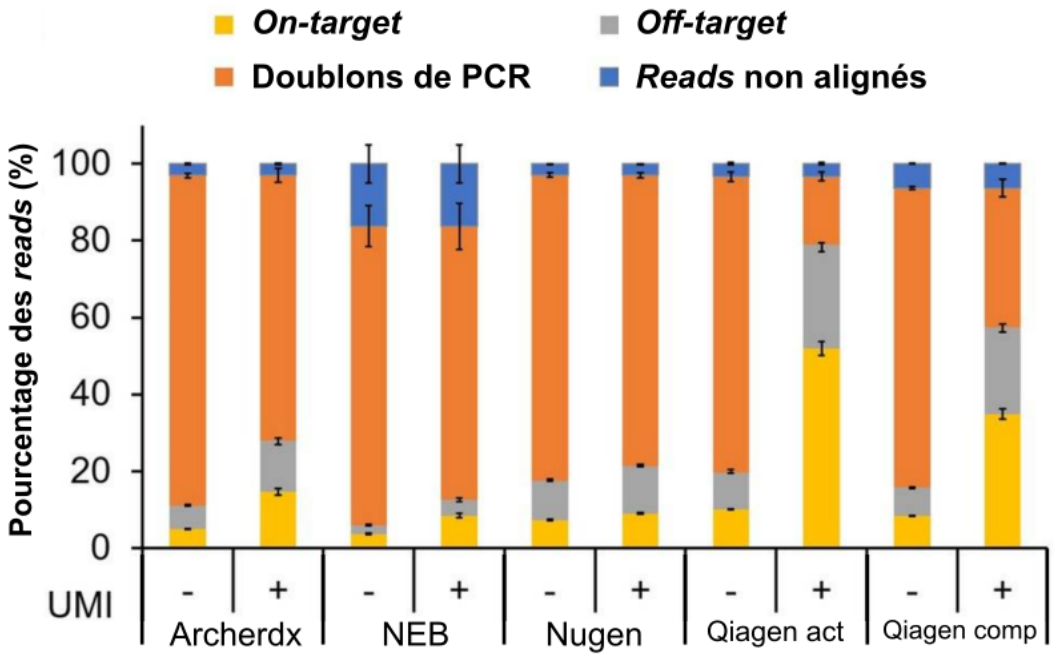


FIGURE 3.7 – Diagramme à barres empilées montrant les fractions de *reads* (*reads* non alignés, doublons de PCR et *reads off-target*) et *reads* finaux après filtrage (*reads on-target*) pendant le traitement des données brutes pour cinq kits commerciaux avec et sans UMI. Figure adaptée de [68].

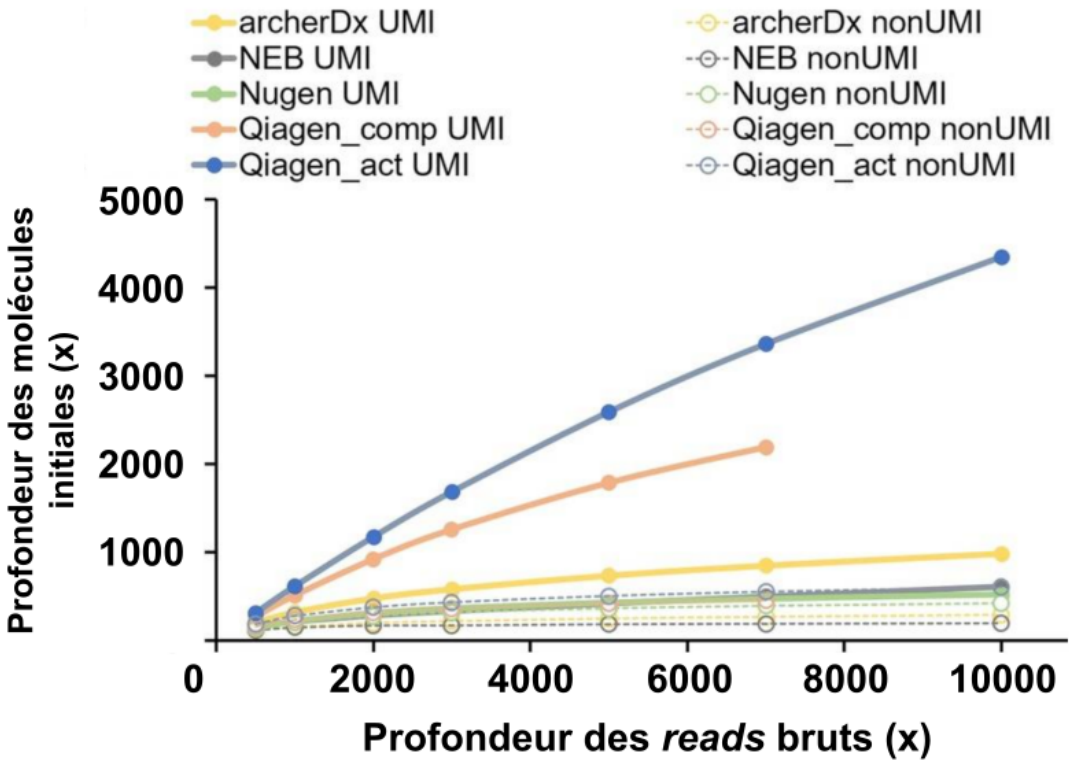


FIGURE 3.8 – Variation du nombre des molécules initiales après filtrage en fonction de la profondeur des *reads* bruts pour cinq kits commerciaux, avec et sans UMI. Figure adaptée de [68].

3.2.2 RNA-Seq

3.2.2.1 La découverte d'un nouvel artefact de séquençage dans l'analyse de l'expression génique basée sur du RNA-Seq [69]

L'utilisation des UMI dans cette expérience a mené les auteurs à mettre en évidence un nouvel artefact de séquençage pouvant causer une surestimation du nombre de transcrits identifiés pour certains gènes et alors fausser les résultats d'une étude d'expression génique. Typiquement, les UMI sont introduits avant l'étape de PCR : ils sont attachés aux fragments d'ADN et sont amplifiés avec ces derniers lors de la PCR. En théorie, les *reads* obtenus partageant un même UMI sont supposés être dérivés de la même molécule initiale. En se basant sur ce principe, les UMI ont été utilisés pour étudier et modéliser la nature du biais d'amplification par PCR afin de pouvoir le corriger en aval. Cela a eu un impact significatif sur les études de transcriptomique, dans lesquelles le comptage des UMI par gène offre des résultats supérieurs au comptage des *reads* bruts, et fournit donc des estimations plus précises de l'expression quantitative des gènes.

Sena *et al.* [69] ont développé des méthodes ciblées RNA-Seq afin d'étudier différents phénomènes chez *Physcomitrella patens* avec précision. Au cours de cette étude, ils ont observé qu'une fraction considérable des *reads* ayant le même UMI, correspond à des positions différentes mais très rapprochées. Pour un UMI donné, ces positions forment des regroupements, appelés *clusters*, de sorte que la majorité des *reads* s'alignent à la bonne position, avec d'autres *reads*, moins nombreux, s'alignant étroitement en amont et en aval, formant une distribution en forme de cloche. Le groupe des *reads* étroitement espacés et partageant un même UMI est alors appelé *cluster*. La taille d'un *cluster* lu par UMI est le nombre de coordonnées adjacentes partageant ce même UMI. Cet artefact est visible dans toutes les librairies contenant des UMI que les auteurs ont créées, et dans tous les ensembles de données RNA-Seq contenant des UMI collectés dans le domaine public. Cet artefact a aussi été obtenu en utilisant différents algorithmes d'alignement STAR [70], GSNAP [71] et HISAT2 [72]. Les différentes observations sont présentées dans les Figures 3.9, 3.10 et 3.11.

Pour évaluer l'impact de cet artefact sur les estimations de l'expression génique du comptage des UMI à des différentes positions dans le même gène en tant qu'événements distincts, les auteurs ont compté le nombre d'UMI par gène, avec et sans regroupement des *reads* d'un même *cluster*. Ensuite, ils ont déterminé le rapport du nombre des UMI non regroupés aux nombres des UMI regroupés représentant le niveau d'expression relatif de chaque gène. Le nombre de gènes et leurs rapports sont représentés sur la Figure 3.12. Dans tous les cas, une fraction considérable des gènes montre des niveaux d'expression artificiellement élevés. Encore pire, dans le cas de l'ensemble de données Yanai1, les gènes ayant une surexpression apparente sont majoritaires.

Finalement, les auteurs conclurent que les séquences portant l'artefact ont tendance à contenir de simples répétitions en tandem (*short tandem repeats*) et que les méthodes de prédiction de ces répétitions arrivent à bien prédire l'occurrence de l'artefact mais pas dans tous les ensembles de données. Ainsi, la présence de ces *short tandem repeats* est une cause potentielle de l'apparition de l'artefact, mais qui ne peut que partiellement l'expliquer.

3.2.2.2 Utilisation des UMI pour éliminer les doublons de PCR en RNA-seq

En premier lieu, les auteurs de cette étude [73] ont cherché à étudier l'impact de la correction des UMI sur l'identification des doublons de PCR. Pour cela, ils ont simulé un ensemble de données contenant un nombre suffisant de molécules d'ARN afin d'obtenir une grande diversité en terme d'UMI. Le moyen le plus simple de déterminer des *reads*

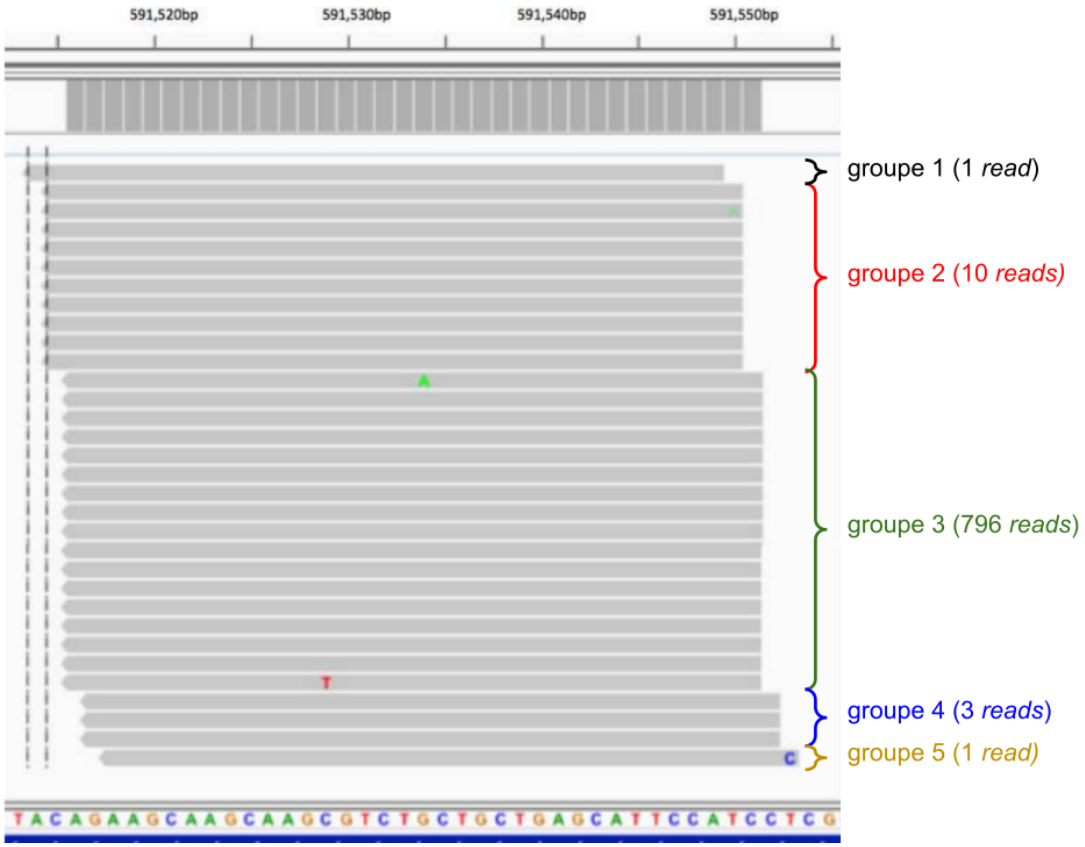


FIGURE 3.9 – Un exemple de *reads* partageant un même UMI et s'alignant à des positions adjacentes sur le génome. Le nombres des *reads* correspondant aux cinq coordonnées indiquées sont 1, 10, 796, 3 et 1, respectivement. Les 796 alignements du milieu ont été modifiés pour permettre de voir les *reads* à chaque position sur une seule figure. Figure adaptée de [69].

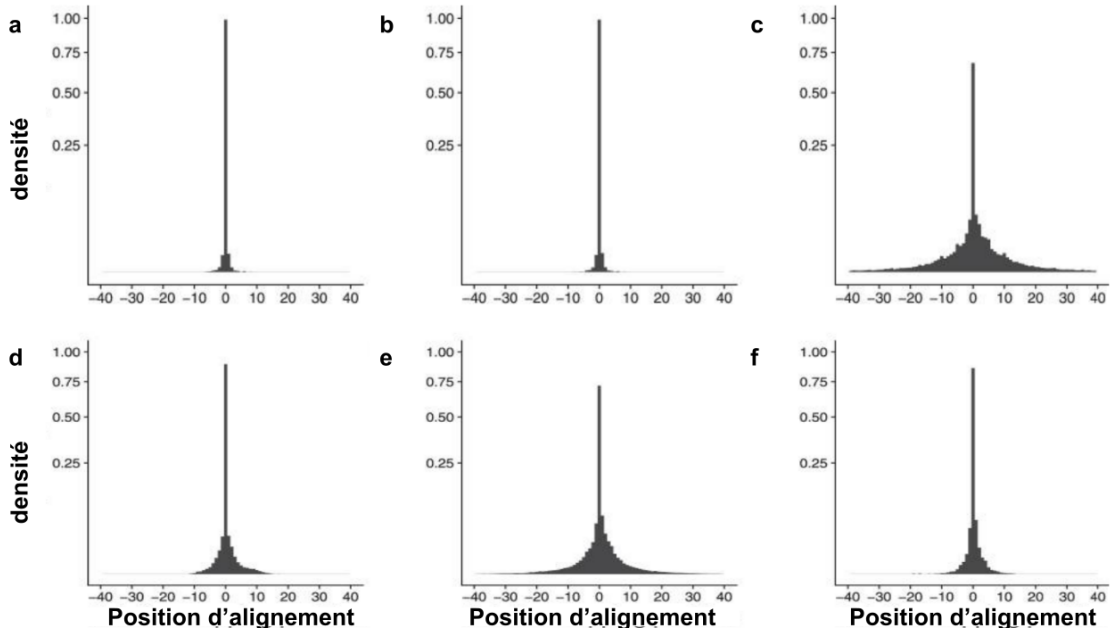


FIGURE 3.10 – Décalages des *reads* partageant un UMI dans six ensembles de données. Les décalages de toutes les tailles sont affichés. L'axe Y représente la densité de probabilité. (a) run_171108. (b) run_170420. (c) SCRB. (d) La Manno. (e) Yanai1. (f) Yanai2. Figure adaptée de [69].

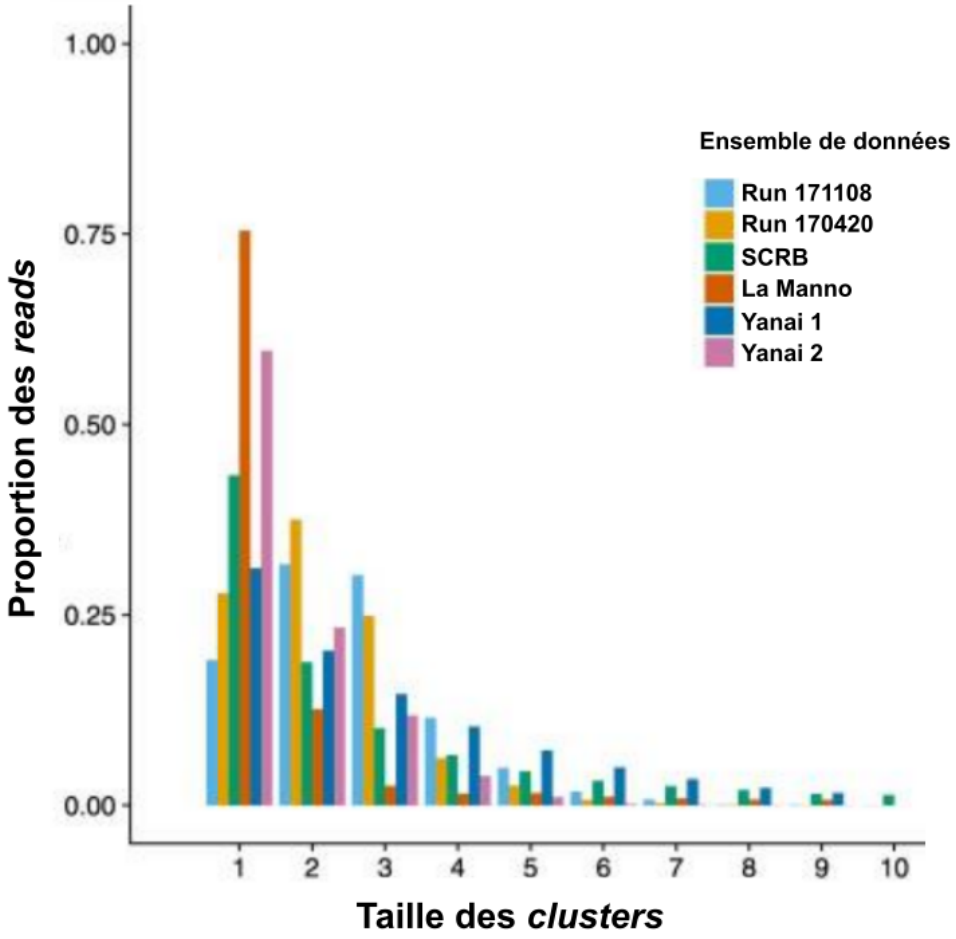


FIGURE 3.11 – Proportions de *reads* trouvés dans des *clusters* de tailles différentes pour les six ensembles de données. Figure adaptée de [69].

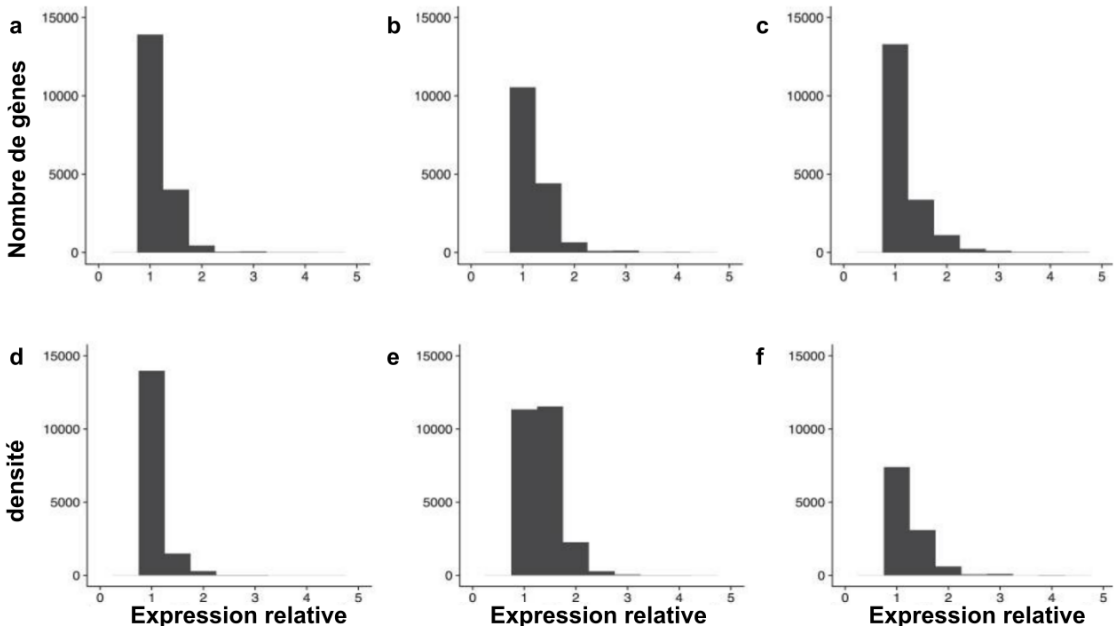


FIGURE 3.12 – Expression relative apparente des gènes si les *reads* décalés ne sont pas pris en compte. L'alignement des UMI aux gènes a été compté avec ou sans regroupement des *clusters*. Le nombre de gènes et leurs rapports du nombre des UMI non regroupés aux nombres des UMI regroupés sont tracés pour les six ensembles de données. (a) run_171108. (b) run_170420. (c) SCRB. (d) La Manno. (e) Yanai1. (f) Yanai2. Figure adaptée de [69].

biologiquement identiques est de rechercher ceux qui ont la même séquence nucléotidique mais marqués par des UMI différents. Cette approche suppose qu'il n'y a pas d'erreur dans la réplication ou la lecture des UMI, car de telles erreurs pourraient rendre des séquences UMI identiques différentes et vice versa, provoquant une mauvaise identification des doublons de PCR. Cependant, des erreurs dans les séquences des UMI peuvent survenir lors du séquençage et de l'amplification par PCR, et il a été démontré que la correction informatique de ces erreurs améliore l'identification des doublons de PCR ([74], [75], [76]).

Pour évaluer la précision de l'identification des doublons PCR à l'aide des UMI, ils ont calculé la différence entre le nombre de *reads* après élimination des doublons de PCR (estimation) et la vraie valeur (vérité) par rapport à la valeur vraie (vérité) : (estimation - vérité) / vérité. Cette métrique reflète la mesure dans laquelle les UMI surestiment ou sous-estiment la vérité en tant que fraction de la valeur vraie. La Figure 3.13 illustre les résultats obtenus.

Ensuite, les chercheurs ont comparé l'identification des doublons par PCR à l'aide des UMI avec les coordonnées des *reads*, à l'approche conventionnelle consistant à utiliser les coordonnées seules. Lorsque seules les coordonnées sont utilisées, 16,4 à 44,5% des *reads* sont jugés comme étant des doublons de PCR, alors qu'en utilisant les informations UMI en conjonction avec les coordonnées d'alignement, seulement 1,89-10,67% des *reads* sont identifiés comme doublons (Figure 3.14). Autrement dit, la majorité des *reads* alignés à des coordonnées identiques n'étaient en fait pas des doublons de PCR mais plutôt des molécules initiales bien distinctes.

Finalement, les auteurs de l'étude ont cherché à caractériser les paramètres pouvant affecter le nombre de doublons obtenus lors d'une expérience RNA-seq. Ils ont d'abord

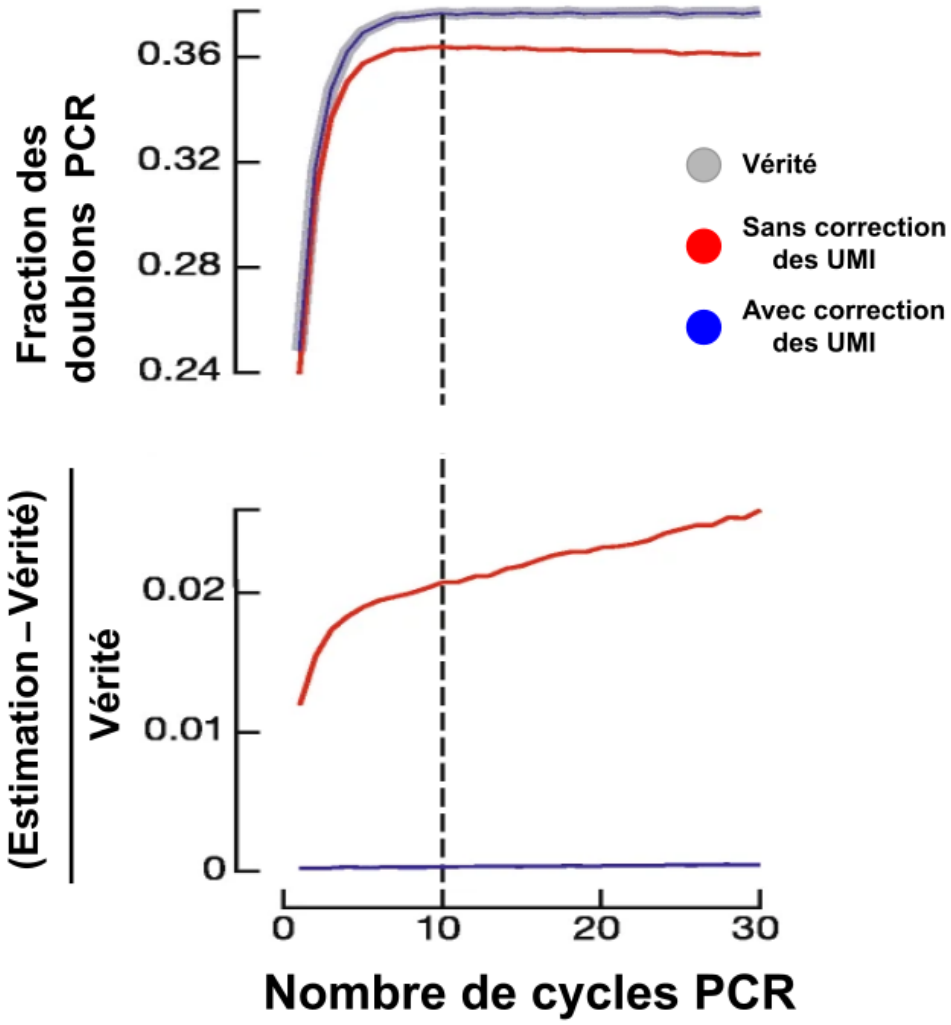


FIGURE 3.13 – Simulation de la suppression des doublons PCR avec ou sans correction d'erreur pour les UMI. Le nombre de cycles PCR a été varié tandis que la quantité initiale d'ARN et la profondeur de séquençage sont restées constantes. Le graphique supérieur montre la fraction de doublons de PCR et celui du bas montre la précision de la détection des doublons. Figure adaptée de [73].

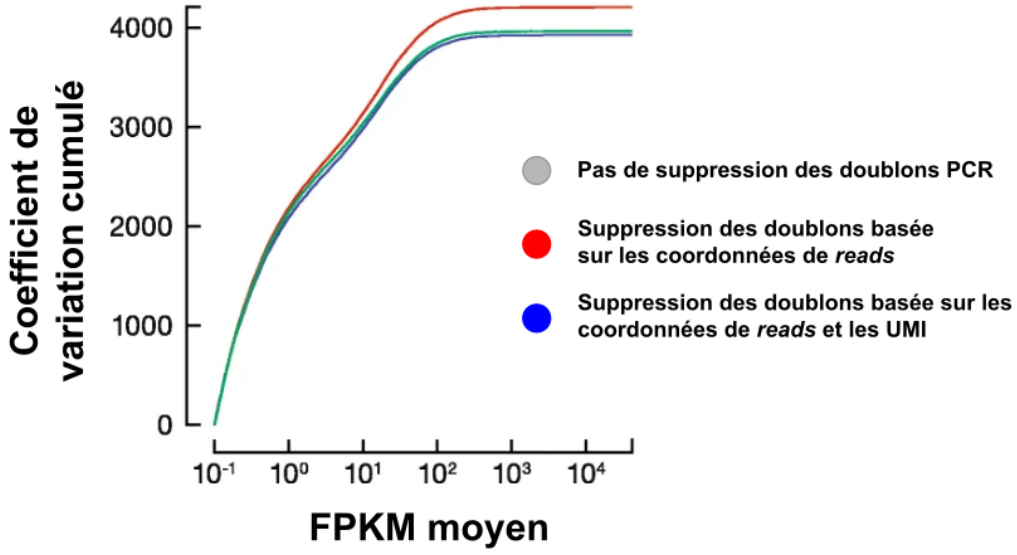


FIGURE 3.14 – Relation entre le coefficient de variation cumulé et l'abondance des transcrits mesurée en FPKM (*Fragments Per Kilobase Million*). Figure adaptée de [73].

analysé un ensemble de cinq librairies avec UMI, faites avec des quantités progressivement décroissantes d'ARN initial et un nombre croissant de cycles de PCR : 4 μg (8 cycles), 2 μg (9 cycles), 1 μg (10 cycles), 500 ng (11 cycles), 125 ng (13 cycles). Ils ont observé qu'une plus petite quantité d'ARN initial (et par conséquent un plus grand nombre de cycles PCR) produit des fractions plus élevées de doublons de PCR (Figure 3.15A). Par exemple, la librairie de 125 ng à 13 cycles a donné 10,7% des doublons PCR, tandis que la librairie de 4 μg à 8 cycles réalisée par la même procédure ne contenait que 1,79% de doublons PCR.

Pour savoir la vraie cause de l'augmentation du nombre de doublons (une amplification plus importante ou un ARN initial plus petit), un deuxième ensemble de neuf librairies UMI RNA-seq a été analysé. Cette fois-ci, toutes les librairies sont générées à partir d'une même quantité d'ARN initial (5 μg) mais amplifiées en utilisant 14 à 30 cycles de PCR. Conformément aux simulations, ces librairies n'ont pas montré de tendance discernable entre la fraction des doublons de PCR et le nombre de cycles de PCR (Figure 3.15B). Ainsi, on peut conclure que ce n'est pas le nombre de cycles PCR réalisés qui influence le nombre de doublons PCR obtenus mais plutôt la quantité initiale d'ARN analysé.

Les auteurs finissent par conclure que l'utilisation des UMI pour identifier et éliminer les doublons de PCR reste indispensable pour ne pas les surestimer et alors diminuer le nombre de transcrits retenus. Ils recommandent aussi que la correction des UMI soit utilisée pour toutes les expériences de RNA-seq puisqu'elle représente un moyen efficace pour améliorer la précision de détection, même si le gain reste minime.

3.3 Outils

Les UMI ont été initialement proposés comme méthode pour compter le nombre de molécules d'ARNm dans un échantillon et ont depuis été utilisés pour marquer explicitement les doublons de PCR. Plus récemment, ils ont servi à identifier en toute confiance les doublons de PCR dans des expériences de séquençage à haut débit. Depuis leur introduction dans les expériences de NGS, de nombreux outils ont été d'abord adaptés pour

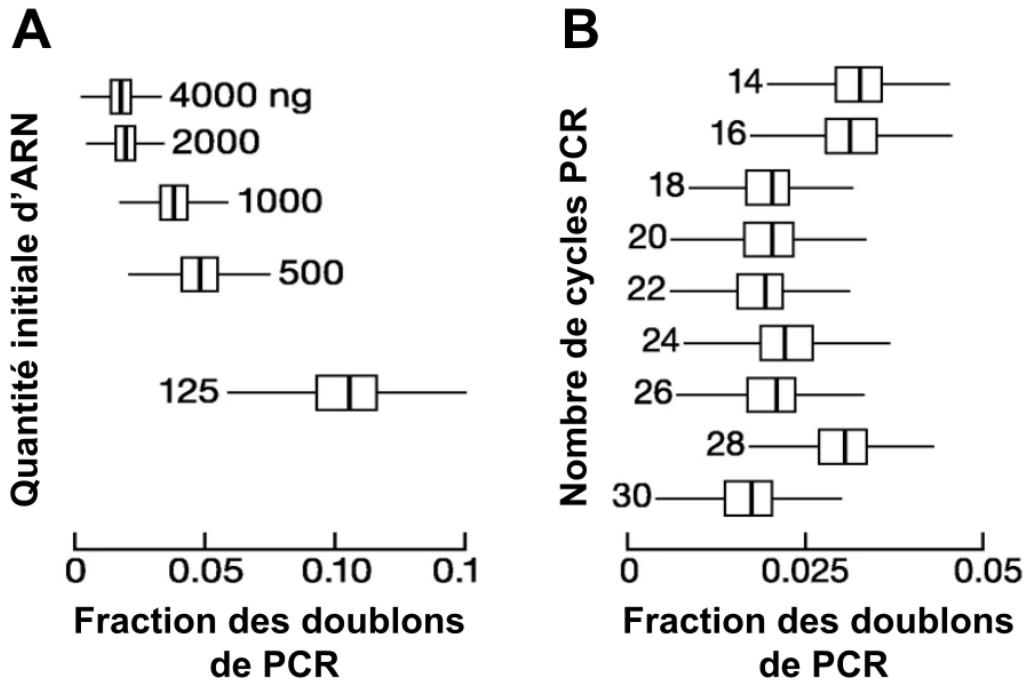


FIGURE 3.15 – Fraction des doublons PCR pour tous les gènes pour (A) une série de librairies RNA-seq fabriquées avec différentes quantités d'ARN initial, et (B) une série de librairies RNA-seq avec UMI toutes faites avec 5 μg d'ARN initial mais avec un nombre croissant de cycles de PCR. Figure adaptée de [73].

tenir compte de la présence de ces nouvelles séquences dans les *reads* comme SAMtools [48], Picard [77] et GATK [78]. Ensuite, de nouveaux outils ont commencé à apparaître, des outils spécialement conçus pour résoudre des problématiques apportées par la présence des UMI dans les *reads*, tout en restant efficace en terme de temps d'exécution et de consommation en mémoire (*RAM*). De ces outils, on cite notamment UMI-tools [35] qui sert à extraire les UMI des *reads* et propose plusieurs méthodes pour les corriger, et gencore [79] permettant de construire des *reads* consensus afin de supprimer les erreurs et les doublons PCR. De même, des *variants callers* implémentant une analyse des UMI ont été développés dont les principaux sont DeepSNVMiner [58], MAGERI [57] et smCounter2 [80].

3.3.1 UMI-tools

Les développeurs de l'outil UMI-tools ont d'abord montré que les erreurs dans les séquences UMI sont courantes et par conséquent, ont introduit des méthodes basées sur des graphes pour tenir compte de ces erreurs et les corriger lors de l'identification des doublons PCR. En utilisant ces méthodes, ils ont démontré une précision de quantification améliorée à la fois dans des conditions simulées et dans des ensembles de données réelles. Dans les deux cas, la reproductibilité entre les réplicats est améliorée à l'aide de la méthode proposée basée sur les graphes, ce qui démontre l'intérêt de tenir correctement compte des erreurs dans les UMI. Ces méthodes sont implémentées dans le logiciel open source UMI-tools.

Tout d'abord, les auteurs ont supposé que les erreurs dans les UMI créent des groupes d'UMI similaires à un locus génomique donné. Pour confirmer cela, ils ont calculé le

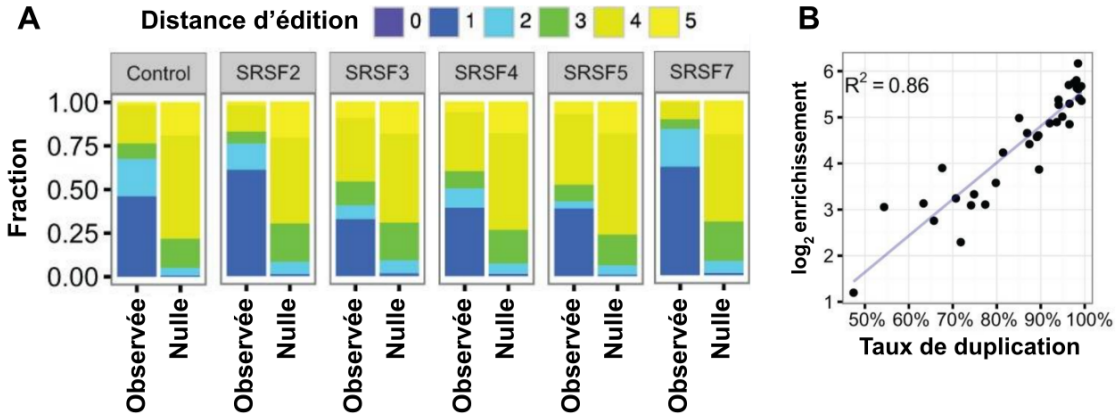


FIGURE 3.16 – Modélisation des erreurs dans les UMI. (A) Distances moyennes d'édition (arrondies aux nombres entiers) entre les UMI avec les mêmes coordonnées d'alignement. Les positions génomiques avec un seul UMI ne sont pas affichées. (Nulle) Espérance nulle de l'échantillonnage aléatoire des UMI, en tenant compte de la distribution à l'échelle du génome des UMI. (B) Corrélation entre le taux de duplication et l'enrichissement des positions ayant une distance d'édition moyenne de 1. Figure adaptée de [35].

nombre moyen de bases différentes, appelé distance d'édition (*edit distance*), entre les UMI s'alignant aux mêmes coordonnées génomiques et ont comparé la distribution des ces distances moyennes à une distribution nulle générée par échantillonnage aléatoire (donc qui représente le hasard). Les résultats obtenus dans la Figure 3.16A confirment que les UMI sont plus similaires les uns aux autres que ce que la distribution nulle prévoit. Ceci suggère fortement que les erreurs de séquençage et/ou de PCR génèrent des UMI artéfactuels. En outre, l'enrichissement des faibles distances d'édition est bien corrélé avec le taux de duplication par PCR. La Figure 3.16B met en évidence un enrichissement de 25 fois pour les positions avec une distance d'édition moyenne de 1, par rapport à la distribution nulle.

En ce qui concerne la correction des UMI, les auteurs ont comparé cinq méthodes différentes présentées dans la Figure 3.17. La première méthode est appelée *unique* et suppose que chaque UMI à un locus génomique donné représente une molécule unique différente. La deuxième méthode a déjà été utilisée par Islam *et al.* [75] et propose de supprimer les UMI ayant une fréquence inférieure à 1% de la moyenne de tous les UMI. Cette méthode est appelée *percentile*. Les trois dernières méthodes ont toutes été développées par les auteurs de cet article et se basent sur des graphes reliant des UMI séparés par une distance d'édition de 1. Dans tous les cas, l'objectif est de réduire le graphe à un ou plusieurs UMI représentatifs de l'ensemble des UMI.

La méthode la plus simple est appelée *cluster* et consiste à fusionner tous les UMI du graphe, en ne conservant que l'UMI avec le plus grand nombre. Pour cette méthode, le nombre de réseaux formés à un locus donné équivaut au nombre estimé de molécules uniques. En revanche, cette méthode pourrait sous-estimer le nombre de molécules uniques, en particulier pour les graphes complexes.

Pour résoudre correctement les graphes complexes, une deuxième méthode, appelée *adjacency*, a été développée en se basant sur le nombre de nœuds dans le graphe. Le nœud le plus abondant et tous les nœuds qui y sont connectés sont sélectionnés d'abord. Si cela ne tient pas compte de tous les nœuds du graphe, le nœud suivant le plus abondant et ses voisins sont également sélectionnés. Ceci est répété jusqu'à ce que tous les nœuds

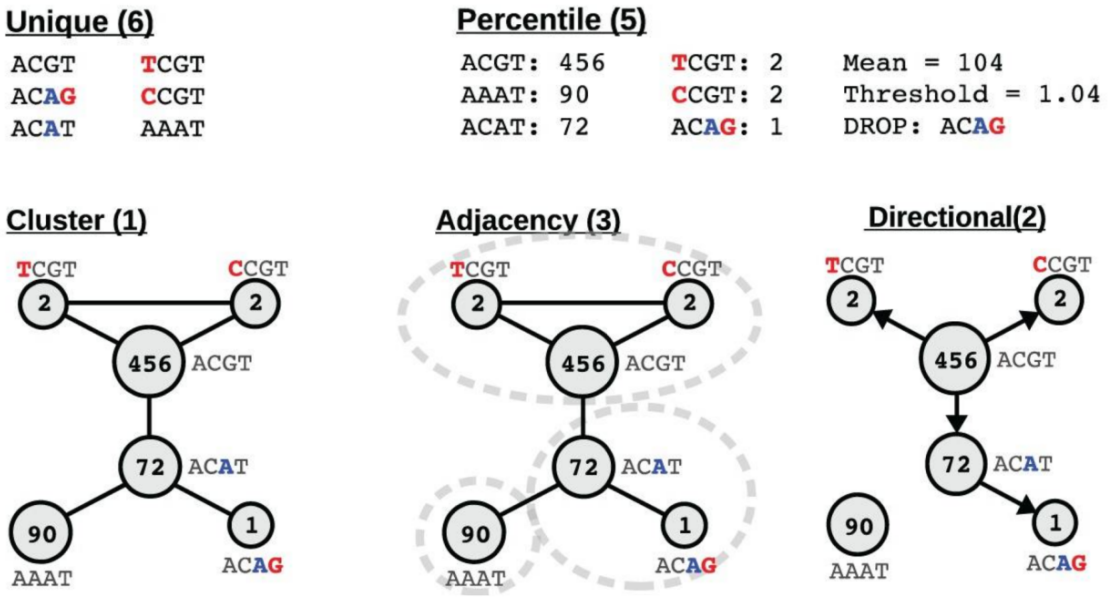


FIGURE 3.17 – Les cinq méthodes d'estimation et de correction de molécules uniques à partir des séquences UMI alignées à un même locus génomique. Les bases en rouge sont supposées être des erreurs de séquençage et les bases en bleu sont des erreurs de PCR. Le nombre estimé de molécules uniques pour chaque méthode est indiqué entre parenthèses. Figure adaptée de [35].

du graphe soient pris en compte. Par conséquent, le nombre total d'étapes nécessaires pour résoudre le graphe formé à un locus donné est équivalent au nombre de molécules uniques estimées.

La dernière méthode se base sur le raisonnement que (1) les UMI générés par une seule erreur de séquençage doivent être plus abondants que ceux générés par plus que deux erreurs et que (2) les UMI résultant d'erreurs de PCR devraient avoir des comptes plus élevés que les UMI résultant d'erreurs de séquençage. La méthode s'appelle *directional* et utilise des liens orientés pour relier les nœuds du graphe. Les nœuds à une distance d'édition de 1 sont reliés lorsque la distance d'édition les séparant est de 1 et lorsque $n_a \geq 2n_b - 1$, où n_a et n_b sont les comptes UMI du nœud a et du nœud b. Dans ce cas, le lien a une orientation précise et est dirigé, dans notre exemple, du nœud a au nœud b. L'ensemble du graphe orienté est alors considéré comme provenant du nœud le plus abondant. La composante -1 a été incluse pour tenir compte des UMI ayant des comptes faibles, chacun séparé par une distance d'édition de 1 et pour lesquels le seuil $2n$ seul est trop restrictif.

Pour comparer la précision des méthodes proposées, un ensemble de données simulées a été produit tout en variant des paramètres différents. Pour chaque méthode, la variation de la métrique log_2 [(estimation - vérité) / vérité] en fonction du paramètre étudié a été analysée. L'augmentation de la longueur UMI ou de la profondeur de séquençage entraîne une augmentation linéaire du degré de surestimation des méthodes *unique* et *percentile* (Figure 3.18A, B). Ceci est prévu puisque l'augmentation de l'un ou l'autre augmente linéairement la quantité totale des UMI et donc le nombre des erreurs potentielles. En revanche, les estimations des méthodes basées sur les graphes restent relativement stables, la méthode *directional* montrant la plus grande précision et la plus faible variance. D'autre part, l'augmentation du taux d'erreur de séquençage conduit à une surestimation exponentielle pour les méthodes *unique* et *percentile* (Figure 3.18C), avec

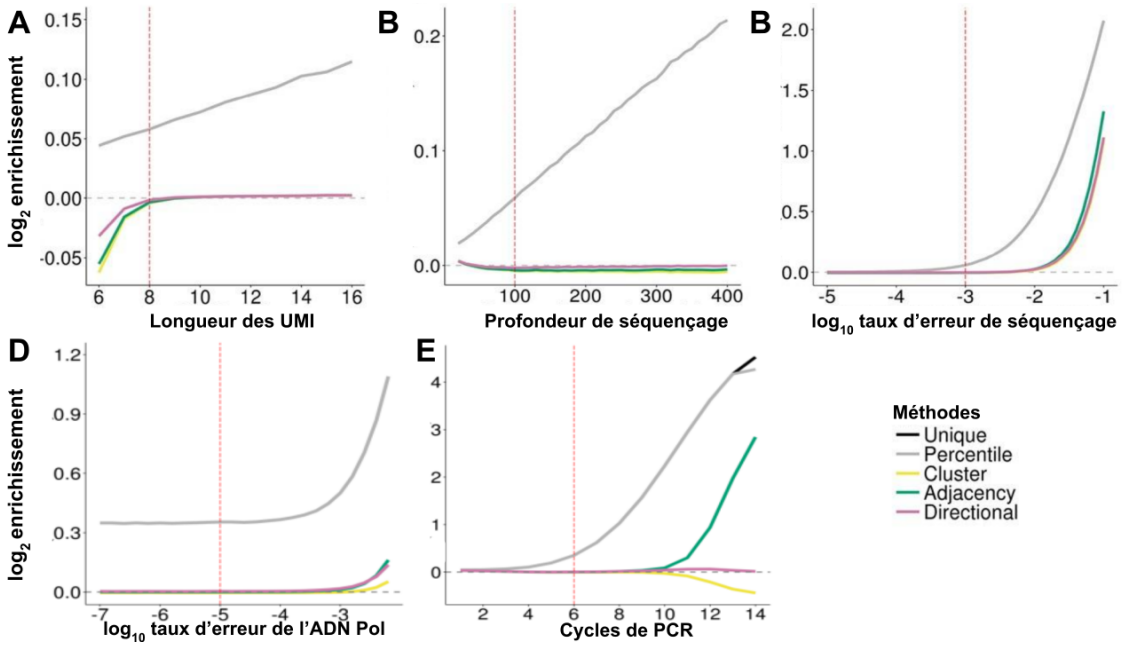


FIGURE 3.18 – Comparaison des cinq méthodes avec des données simulées. Pour chaque graphique, tous les paramètres de simulation sauf un sont maintenus constants, le paramètre restant variant est indiqué sur l'axe des abscisses. (A) Longueur des UMI. (B) Profondeur de séquençage. (C) log_{10} du taux d'erreur de séquençage. (D) log_{10} du taux d'erreur de l'ADN Polymérase. (E) Nombre de cycles de PCR réalisés. Les graphiques montrent la précision de la quantification, présentée par le log_2 de l'enrichissement, ce dernier étant la différence normalisée entre l'estimation et la vérité (log_2 [(estimation - vérité) / vérité]). La ligne rouge en pointillés représente la valeur utilisée pour ce paramètre dans toutes les autres simulations. La ligne grise en pointillés représente une précision parfaite. Les méthodes *unique* et *percentile* donnent des résultats identiques avec les paramètres indiqués ici et sont donc superposées. Figure adaptée de [35].

une surestimation de 1,3 fois observée pour un taux d'erreur de 0,01 contre < 1,05 fois pour les méthodes basées sur les graphes. L'augmentation du taux d'erreur au cours de l'étape PCR a eu un impact similaire (Figure 3.18D). Cependant, cela n'a été observé que lorsque le taux d'erreur de l'ADN polymérase est simulé à > 0,001, ce qui est considérablement plus élevé que les taux d'erreur rapportés même pour les ADN polymérases les moins fiables. Ainsi, cela confirme que les erreurs de séquençage sont susceptibles d'être la principale source d'erreurs dans les UMI. La Figure 3.18E montre que l'augmentation du nombre de cycles PCR a un impact minime sur la précision relative des méthodes. Bien que les trois méthodes basées sur des graphes aient fonctionné de manière très similaire, la méthode *directional* a systématiquement donné des estimations plus précises et moins variables. Par exemple, lorsque la profondeur de séquençage a été augmentée à 400x, la méthode *cluster* a estimé une moyenne de 19,92 molécules distinctes, 19,94 pour la méthode *adjacency* tandis que la méthode *directional* en a trouvé 19,99 et ceci pour une vérité de 20 molécules distinctes. Cependant, aucune différence n'a été observée entre la méthode *percentile* et *unique* dans la plupart des conditions testées.

Pour implémenter ces méthodes dans le cadre de la suppression des doublons PCR des *reads* bruts, UMI-tools a été développé avec deux commandes, *extract* et *dedup*. La fonction *extract* prend l'UMI de la séquence du *read* d'un fichier FASTQ et l'ajoute à l'identificateur du *read* afin qu'il soit conservé pour les analyses en aval. Cette fonction s'attend

à ce que les UMI soient contenus au même emplacement pour chaque *read*. La fonction *dedup* prend en entrée un fichier BAM, identifie les *reads* avec les mêmes coordonnées génomiques comme des doublons potentiels et supprime les doublons de PCR en analysant les UMI selon la méthode choisie. Le temps requis et les besoins en mémoire pour exécuter la déduplication dépend du nombre de *reads* dans le fichier d'entrée, de la longueur de l'UMI et du niveau de duplication. Il faut environ 220 secondes et 100 Mo de *RAM* pour traiter un fichier d'entrée à une extrémité (*single-end*) de 32 millions de *reads* avec des UMI de 5 pb pour environ 700 000 molécules initiales. UMI-tools est disponible en tant qu'un module Python depuis PyPI ou conda (umi_tools) mais aussi en tant que logiciel indépendant depuis GitHub (UMI-tools).

3.3.2 Gencore

L'outil gencore [79] a été développé pour répondre à deux problèmes principaux rencontrés dans les expériences NGS : la correction des erreurs et la déduplication des *reads*. Cet objectif est atteint en se basant sur la méthode de construction d'un *read* consensus pour un groupe des *reads* partageant un même UMI et s'alignant à un même locus génomique. Les développeurs de cet outil assurent que gencore accomplit ses fonctions tout en restant beaucoup moins exigent et plus rapide que les autres outils. Sa particularité est qu'il rapporte les résultats statistiques aux formats HTML et JSON. Le rapport au format HTML contient de nombreuses figures interactives indiquant la couverture statistique et les informations de duplication et de déduplication. Le rapport au format JSON contient tous les résultats statistiques et est interprétable pour les programmes en aval.

Le logiciel gencore prend en entrée un fichier BAM trié par position et un fichier FASTA du génome de référence. Si les données ont des UMI, elles peuvent être prétraitées en utilisant la fonction *extract* de UMI-tools pour extraire les UMI des *reads* et les ajouter à leurs identificateurs. L'outil gencore se sert du génome de référence FASTA pour aider à la génération des *reads* consensus. Si les données proviennent d'un séquençage ciblé, un fichier BED peut également être fourni pour décrire les régions de capture. Dans ce cas, les statistiques de couverture dans les régions BED seront également rapportées dans les rapports HTML/JSON. Le *workflow* de l'outil est expliqué en détail et est illustré dans la Figure 3.19. Il peut être décomposé en six étapes principales :

1. Regroupement par position : toutes les paires de *reads* mappées sont d'abord regroupées par position d'alignement. Les *reads* s'alignant sur un même chromosome, une même position de départ et une même position finale seront regroupés.
2. Regroupement par UMI : pour chaque groupe de *reads* regroupés par position, les paires lues sont ensuite regroupées par leurs UMI avec une tolérance d'une distance d'édition de 1. Si les données n'ont pas d'UMI, cette étape est ignorée.
3. Filtrage des *clusters* : chaque *cluster* sera filtré en comparant le nombre de *reads* qu'il contient avec un seuil choisi par l'utilisateur.
4. Notation des paires : un score par défaut sera initialement attribué à chaque base dans les *reads*. Pour chaque paire de *reads* dans un *cluster*, la région superposée des *reads* appariés est calculée. Le score de chaque base dans la région de chevauchement est ajusté en fonction de sa cohérence avec sa base appariée, en tenant compte de leurs scores de qualité.
5. Notation des *clusters* : à cette étape, les scores totaux sont calculés en résumant les scores calculés dans l'étape précédente.
6. Génération du *read* consensus : pour chaque position dans un *cluster*, sa diversité de base est calculée en fonction des scores des différentes bases calculés dans

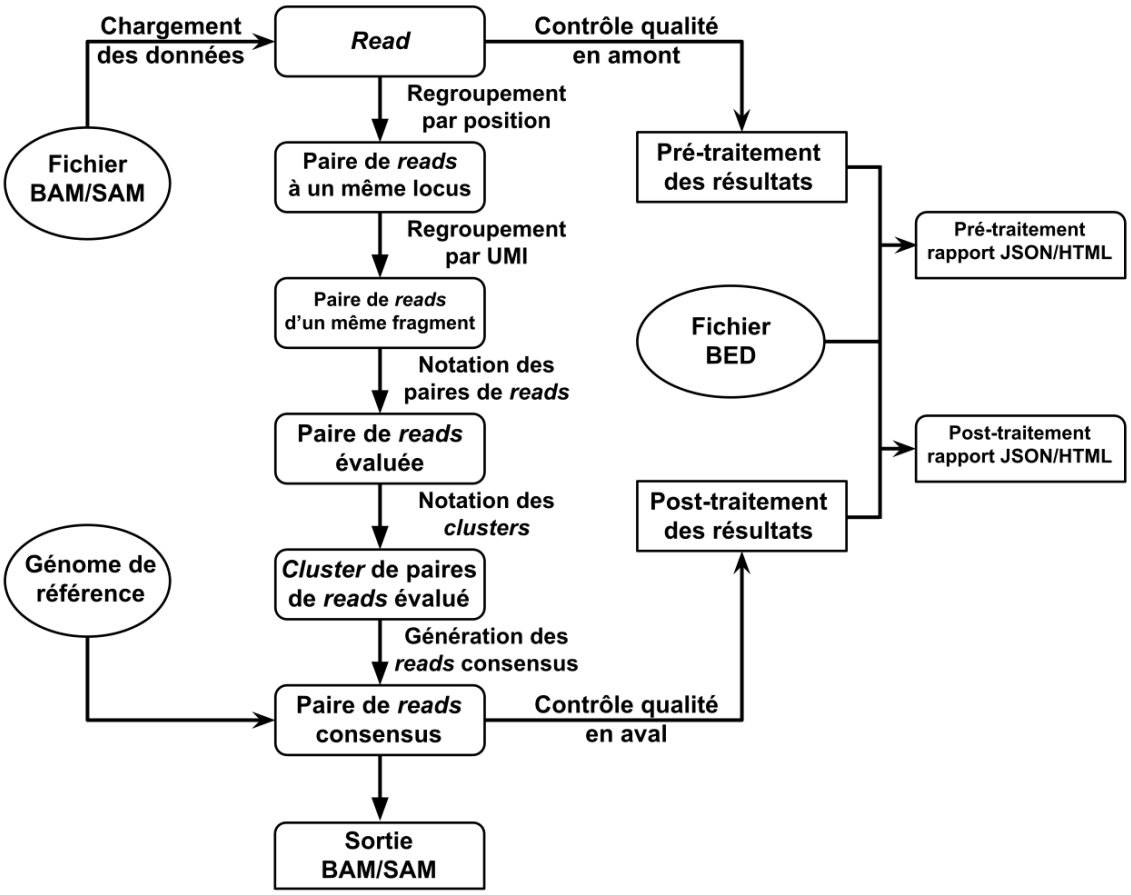
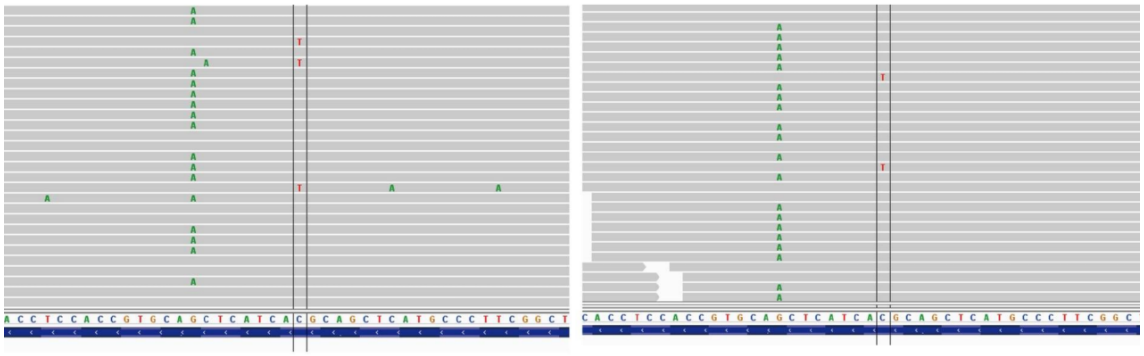


FIGURE 3.19 – *Workflow* de l'outil gencore. Figure adaptée de [79].

l'étape précédente. Si gencore trouve une base dominante, cette base sera présentée dans le consensus. Également, si tous les *reads* à cette position présentent des bases avec de faibles scores, la base correspondante dans le génome de référence sera utilisée. L'utilisation du génome de référence est l'une des principales différences entre gencore et d'autres outils.

7. Production des résultats : une fois le traitement terminé, gencore génère un résumé des données avant et après le traitement. Certaines mesures telles que la couverture, l'histogramme de duplication, le taux d'alignement, le taux de duplication et le taux de *reads* non-alignés sont signalées dans des rapports au format HTML/JSON. Le rapport HTML contient des figures interactives très pratiques et plus faciles à interpréter.

Pour explorer comment gencore élimine les erreurs de séquençage, les auteurs ont séquencé huit échantillons d'ADN. Les échantillons 1801, 1802, 1803, 1811, 1812 et 1813 sont des échantillons prélevés chez des individus malades alors que les deux échantillons 180N et 181N sont obtenus d'individus sains. Le fichier d'alignement de l'échantillon 1802 a été analysé manuellement avant et après traitement. Cet individu porte la mutation c.2369C>T dans le gène EGFR et donc le variant en résultant (p.T790M) est un vrai variant positif. La Figure 3.20 montre la visualisation d'alignement réalisée par Integrated Genome Viewer (IGV) [81] pour les fichiers avant et après traitement. Sur la Figure 3.20a, qui est le fichier d'alignement original généré suite à l'alignement avec BWA [41], la base mésappariée T marquée en rouge est la vraie mutation positive EGFR p.T790M. Cependant, il existe également d'autres bases mésappariées, représentant des faux positifs



a. *reads* avant traitement avec gencore b. *reads* après traitement avec gencore

FIGURE 3.20 – Comparaison des fichiers d'alignement avant et après traitement avec gencore. Dans cette figure, la position marquée par des lignes doubles est (EGFR) c.2369C>T, donnant le variant p.T790M. (a) montre les *reads* alignés du fichier original, (b) montre les *reads* alignés après le traitement gencore. On constate que les faux positifs qui apparaissent aléatoirement dans le fichier d'alignement d'origine, sont corrigés par gencore. Figure adaptée de [79].

causés par des erreurs de séquençage. Sur la Figure 3.20b, qui est le fichier d'alignement après le traitement par gencore, on remarque que les faux positifs ont disparu, tandis que la vraie mutation positive est conservée. Ce résultat suggère que gencore supprime non seulement les doublons, mais également les erreurs de séquençage.

Finalement, les auteurs ont voulu démontrer que gencore était plus optimisé que les autres outils pour la déduplication et la correction des erreurs de séquençage. Ainsi, ils ont comparé le temps d'exécution de l'outil, sa consommation en RAM et le temps nécessaire de préparation contre des outils bien connus comme Picard, SAMtools et UMI-tools. L'outil gencore peut fonctionner sous deux modes, avec ou sans UMI et la comparaison de performance a été effectuée pour les deux modes. En ce qui concerne les besoins en mémoire, gencore utilise beaucoup moins de mémoire que Picard et UMI-tools. Cependant, vu que gencore consomme de la mémoire supplémentaire pour charger le génome de référence et effectue plus de traitement, il utilise plus de mémoire que SAMtools. Mais, comme le montre la Figure 3.21, son pic de mémoire est toujours inférieur à 8 Go. Ce résultat montre que gencore est un outil léger, très rapide et bien adapté pour une utilisation sur le *cloud*. L'outil gencore a été utilisé dans le laboratoire des auteurs pour analyser environ 10 000 échantillons et est prêt à être adopté par la communauté bioinformatique. Il est écrit en C++ et est disponible depuis GitHub (gencore) sous une licence MIT.

3.3.3 DeepSNVMiner [58]

Bien que l'utilité du séquençage avec UMI soit claire, l'analyse des données et le *variant calling* réalisé ne sont pas triviaux. Le défi technique de travailler avec de telles données est en grande partie dû à la grande variété de méthodes pour attacher des UMI, des méthodes qui génèrent des UMI très différents en ce qui concerne la longueur totale de la séquence et leur position sur les molécules par rapport à la séquence d'intérêt et/ou aux adaptateurs. La capacité de travailler avec de telles données nécessite un logiciel permettant d'abord aux utilisateurs de définir la nature de l'UMI dans leur expérience, suivi d'un flux de travail d'analyse où les UMI sont temporairement supprimés des données

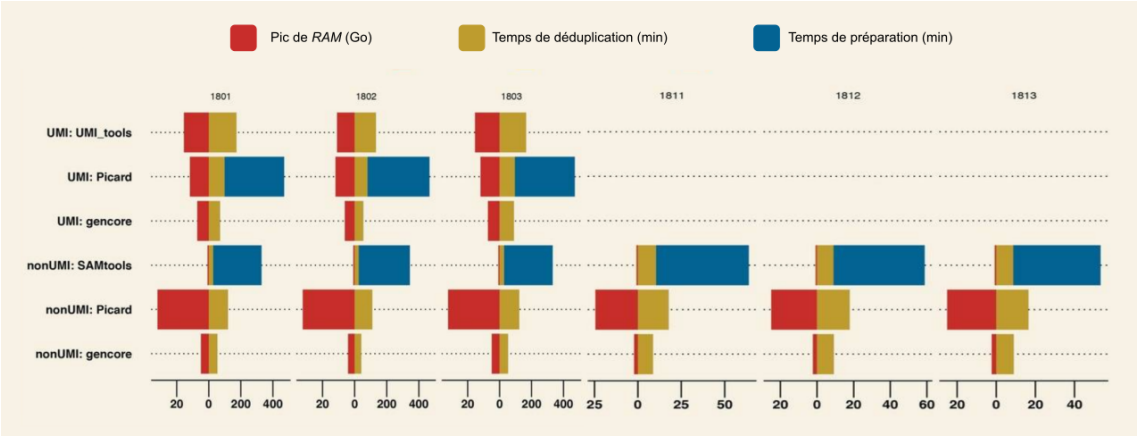


FIGURE 3.21 – Comparaison de la vitesse, du pic de mémoire (*RAM*) de différents outils en mode UMI et non UMI. SAMtools et Picard (en mode UMI) doivent préparer les données avant d'effectuer la déduplication, contrairement à gencore et UMI-tools. Figure adaptée de [79].

de séquence brutes pour l'étape d'alignement, puis restaurés comme moyen de regroupement. Enfin, les variants doivent être appelés au sein de chaque groupe de molécules d'entrée partageant un UMI commun, une tâche intensive en calcul étant donné le grand nombre de groupes d'UMI souvent générés dans une seule expérience. Pour répondre à ce besoin, T. Daniel Andrews *et al.* ont développé DeepSNVMiner, le premier *variant caller* capable de détecter de rares variants (SNV) en analysant les UMI attachés aux *reads*. DeepSNVMiner est un outil autonome et automatisé qui s'exécute dans un environnement Linux ou Macintosh et a été utilisé avec succès même sur du matériel de bureau modeste.

Le *workflow* de DeepSNVMiner consiste à regrouper les *reads* par UMI, suivi par un *variant calling* par groupe d'UMI, identifiant ainsi les mutations qui existaient dans des molécules uniques à partir de l'ADN d'origine hétérogène. La Figure 3.22 montre en détail le *workflow* de DeepSNVMiner pouvant faire appel à d'autres outils externes. Premièrement, l'ensemble des *reads* est soumis à un contrôle de qualité préliminaire pour supprimer les *reads* de faible qualité. Les données sont ensuite interrogées pour la présence d'une séquence d'adaptateur pouvant contaminer les UMI si elles ne sont pas supprimées. Chaque UMI est ensuite identifié sur la base de l'entrée définie par l'utilisateur, supprimé de la ligne de séquence FASTQ et ajouté à l'en-tête du *read* correspondant. Ces nouveaux *reads* et en-têtes sont écrits dans de nouveaux fichiers FASTQ avec les informations d'en-tête et d'UMI utilisées ultérieurement pour détecter des variants spécifiques aux groupes partageant un même UMI. DeepSNVMiner est flexible en ce qui concerne la structure de la balise UMI vu que sa longueur et son emplacement varient généralement en fonction du protocole de marquage et/ou de la technologie de séquençage utilisés. Par exemple, fréquemment l'UMI est ajouté uniquement à l'extrémité 5 'de la région amplifiée, mais dans d'autres protocoles, la séquence de l'UMI est présente à la fois sur les extrémités 5' et 3' et donc doit être concaténée pour déterminer l'UMI final. Ensuite, les *reads* modifiés sont alignés contre un génome de référence avec BWA en utilisant un ensemble de paramètres d'alignement qui permettent les mésappariements mais qui pénalisent l'ouverture d'un espace dans l'alignement, en particulier à la fin des *reads*. Les variants sont ensuite identifiés base par base à l'aide de la commande SAMtools *calmd* dans une région ciblée dont les coordonnées sont précisées dans le fichier BED fourni par l'utilisateur. La sortie de *calmd* est ensuite analysée et les positions des variants et des

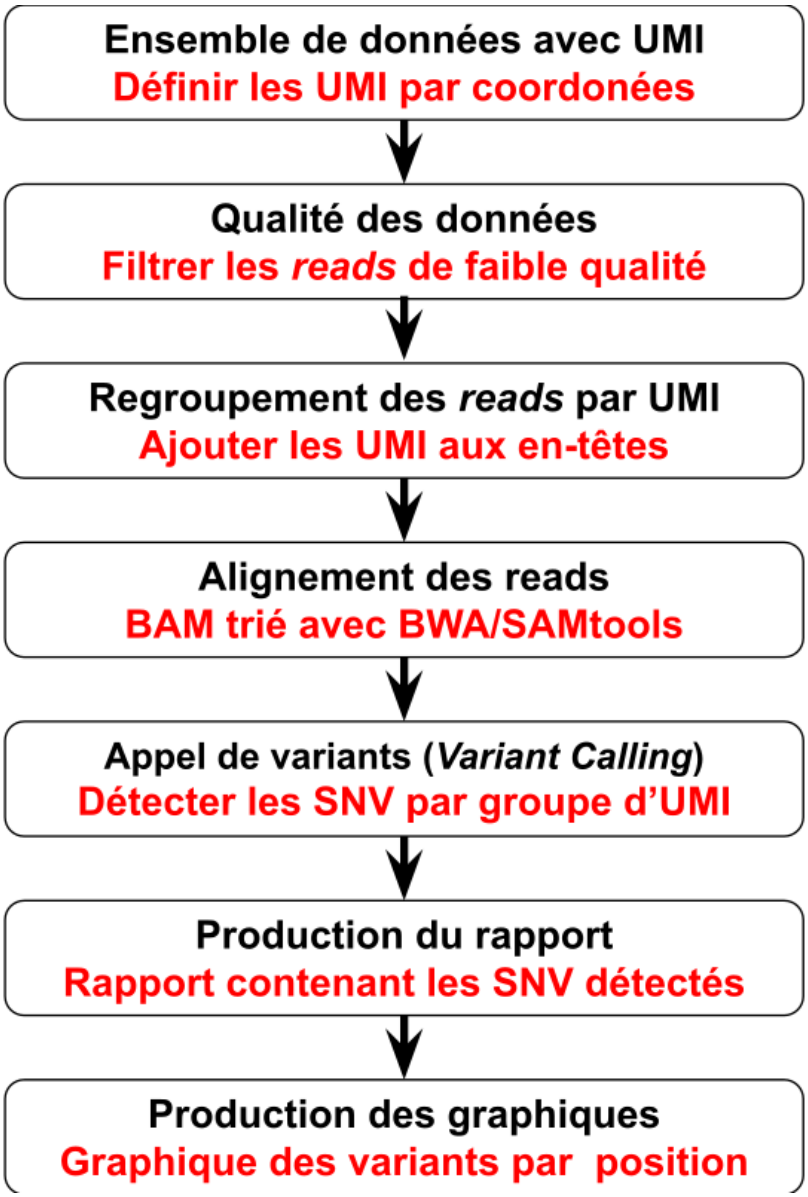


FIGURE 3.22 – *Workflow* de l'outil DeepSNVMiner. Figure adaptée de [58].

reads dans lesquels ils se produisent sont comptées et regroupées en fonction des UMI. Finalement, des graphiques récapitulatifs facultatifs des variants visualisant leur position chromosomique et leur fréquence allélique, ou VAF, (*Variant Allele Frequency*) sont créés à l'aide du logiciel R.

Pour évaluer les performances de DeepSNVMiner, les auteurs l'ont comparé d'abord à des *variant callers* bien connus comme FreeBayes [82], GATK [78], SAMTools [83] et LoFreq [52] en utilisant des données simulées à des niveaux de dilution variables croissants. Deux ensembles de données contenant des *reads paired-end* de 100 pb ont été simulés, chaque paire de *reads* ayant un UMI de 10 pb généré aléatoirement et attaché à l'extrémité 5'. Le premier ensemble de données ne contient aucune mutation tandis que le second ensemble de données d'entrée contient des variants (SNV) générés aléatoirement. Le mélange des deux ensembles de données à des concentrations appropriées simule des niveaux de dilution de 0%, 50%, 90%, 99%, 99,9%, 99,99%, 99,999% et 99,9999% avec un total de 4 000 000 de *reads paired-end*. Pour chaque niveau de dilution, ces *reads* ont

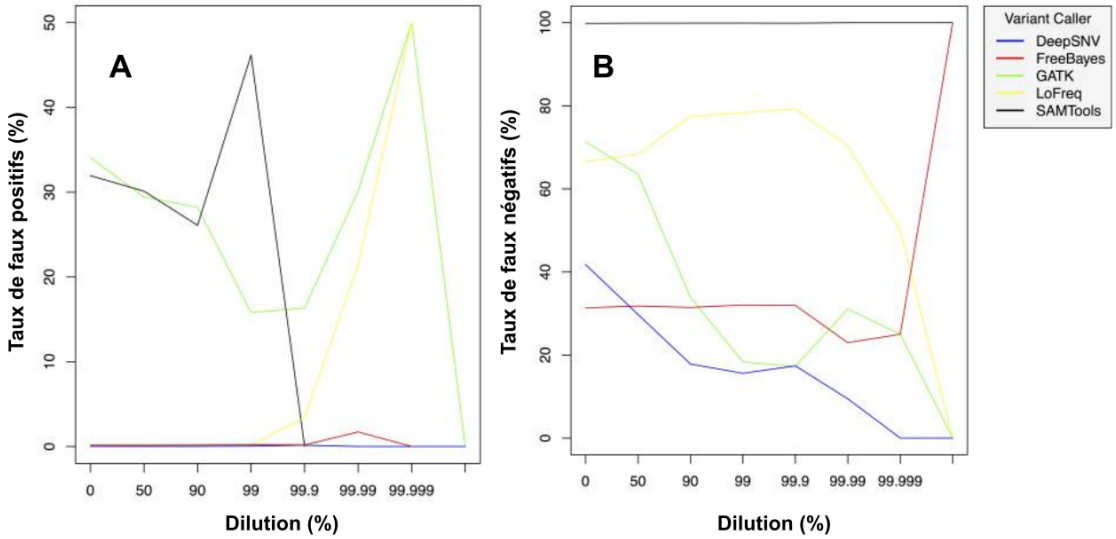


FIGURE 3.23 – Comparaison des performances (taux des faux positifs (A) et (B) taux des faux négatifs) des *variant callers* DeepSNVMiner, FreeBayes, SAMtools, GATK et LoFreq. Figure adaptée de [58].

d'abord été alignés sur le chromosome 22 et les variants appelés à l'aide de DeepSNVMiner, FreeBayes, GATK, SAMTools et LoFreq. Les taux de faux positifs et de faux négatifs ont ensuite été calculés et sont reportés dans la Figure 3.23. Que ce soit pour le taux de faux positifs (Figure 3.23A) ou faux négatifs (Figure 3.23B), on voit clairement que DeepSNVMiner affiche un taux toujours inférieur aux autres *variant callers* et ceci pour toutes les dilutions.

Enfin, DeepSNVMiner a été testé en exécutant une série de dilutions avec de l'ADN génomique de deux lignées cellulaires dont l'une contient un variant somatique hétérozygote connu. Cette mutation est un SNV connu au sein du gène MYD88 à L265P ou chr3 : 38172641, une mutation somatique se produisant fréquemment dans le lymphome non hodgkinien (LNH). Il serait cliniquement utile de disposer d'une méthode pour détecter et dénombrer les cellules rares porteuses de cette mutation dans des échantillons de sang ou de moelle osseuse. De la même façon que pour le test précédent, les deux lignées ont été mélangées à des concentrations appropriées pour obtenir des niveaux de dilution de 0%, 90%, 99%, 99,9%, 99,99%, 99,999% et 99,9999% correspondant à des VAF de 100%, 10%, 1%, 0,1%, 0,01%, 0,001% et 0,0001% respectivement. Les échantillons obtenus ont été séquencés à l'aide d'un Illumina MiSeq et les *reads* en résultant ont été analysés avec DeepSNVMiner, FreeBayes, GATK, LoFreq et SAMTools et la capacité à détecter la mutation hétérozygote a été mesurée. DeepSNVMiner a réussi à détecter la mutation dans les niveaux de dilution jusqu'à 1/1000 contre à 1/100 pour LoFreq, 1/10 pour GATK, et uniquement dans l'échantillon non dilué pour FreeBayes et SAMTools. La mutation a été détectée de manière fiable à des concentrations de 1/1000 par DeepSNVMiner mais pas à des concentrations de 1/10000 (Figure 3.24), indiquant que la limite de détection inférieure se situe quelque part dans cette plage.

3.3.4 MAGERI [57]

MAGERI est un logiciel présenté en 2017 et qui implémente des méthodes d'extraction et de traitement des séquences UMI, une méthode d'assemblage qui regroupe les

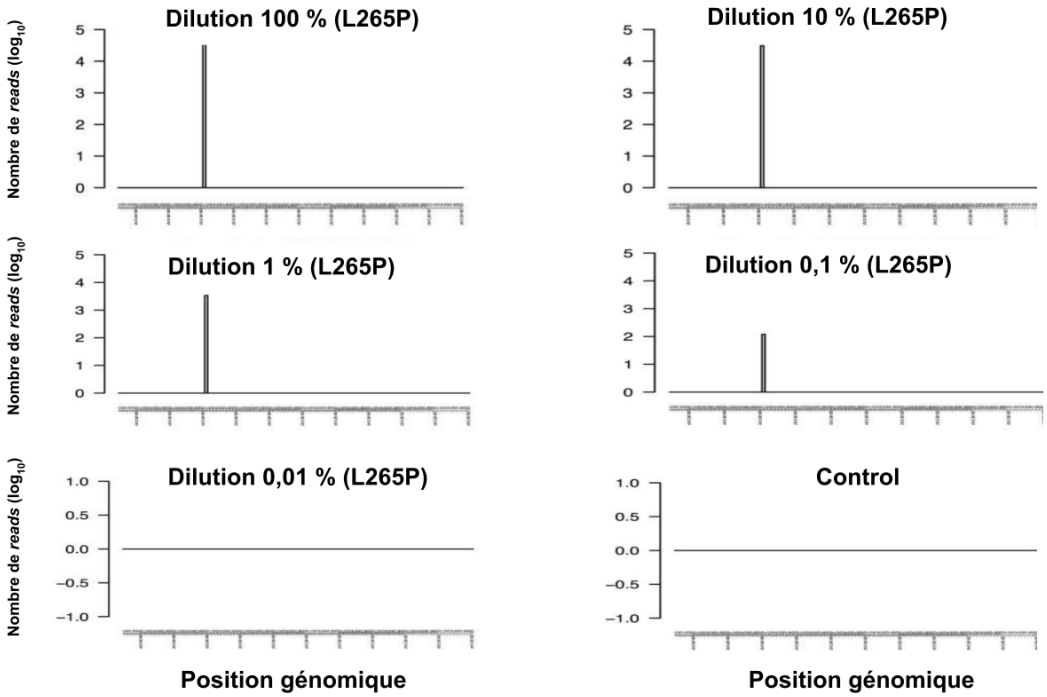


FIGURE 3.24 – Comparaison de détection de la mutation hétérozygote L265P MYD88 sur une série de dilutions croissantes entre DeepSNVMiner, FreeBayes, GATK, LoFreq et SAMtools. Figure adaptée de [58].

reads marqués avec le même UMI pour former des *reads* consensus, et des modules d'alignement de consensus et de *variant calling*. Le *pipeline* corrige les erreurs dans les séquences UMI et effectue un assemblage de consensus robuste capable de gérer des *reads* avec des taux d'erreur relativement élevés et les indels. Il profite également de la réduction des données grâce à la formation de consensus et de la connaissance *a priori* des positions des régions cibles pour exécuter un algorithme d'alignement très sensible. Comme la correction UMI supprime presque toutes les erreurs de séquençage, MAGERI implémente un modèle de score de qualité qui tient compte des erreurs de PCR introduites au stade de l'attachement de l'UMI et des erreurs de PCR du premier cycle qui peuvent se propager pour devenir des variants dominants dans la séquence consensus. Le *workflow* de MAGERI est illustré dans la Figure 3.25.

Pour tester la précision du *pipeline* MAGERI, les auteurs ont sélectionné un standard de référence contenant des mutations à des fréquences alléliques connues. Cet ensemble de données de référence est utilisé pour évaluer la précision du traitement des données étiquetées avec des UMI et des *variant callers*. L'étalon de référence a été utilisé tel quel ou mélangé avec de l'ADN d'un donneur sain dans un rapport de 1 : 9 pour obtenir un spectre de variants connus avec différentes fréquences regroupées en trois niveaux (0,1%, 1% et 5+%), tandis que l'ADN du donneur sain seul servait de témoin négatif. En total, 112 mutations validées sont présentes dans la référence. De ces 112 variants, 50 ont une fréquence autour de 0,1%, 50 ont une fréquence d'environ 1% et seulement 12 ont une fréquence $\geq$ 5%. MAGERI a été utilisé pour analyser les données et détecter ces variants. Les résultats du *variant calling* sont présentés dans la Figure 3.26. MAGERI réussit à détecter tous les variants de fréquence $\geq$ 5%, 46 des 50 variants (92%) ayant une fréquence de 1% et 43 des 50 variants (86%) à 0,1%.

Finalement, pour démontrer l'applicabilité du logiciel MAGERI à détecter des mutations ponctuelles bien précises dans différentes types d'échantillons, les développeurs du

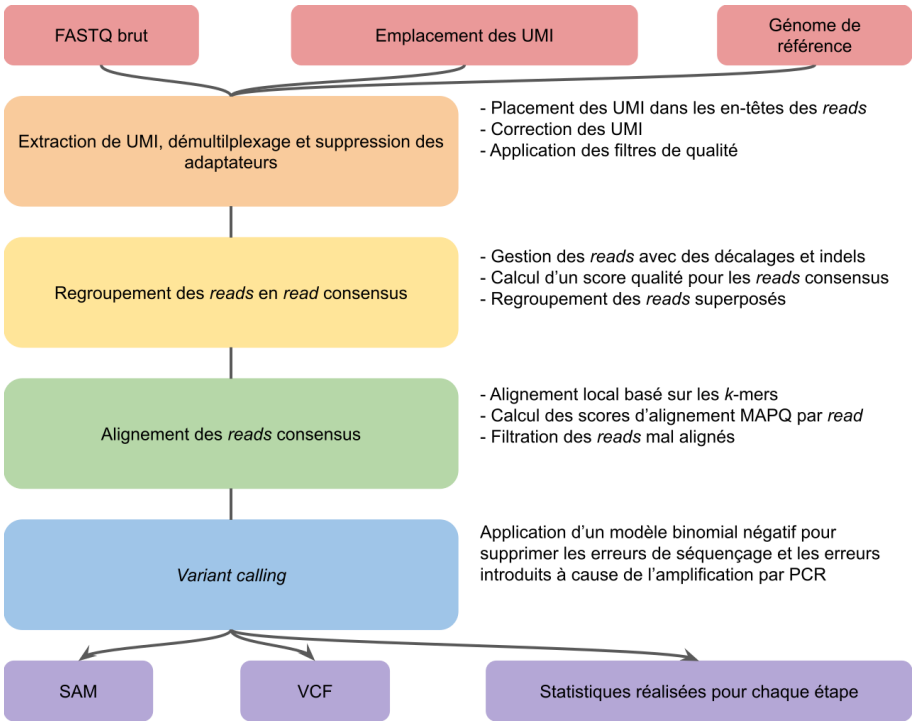


FIGURE 3.25 – *Workflow* de l'outil MAGERI. Figure adaptée de [57].

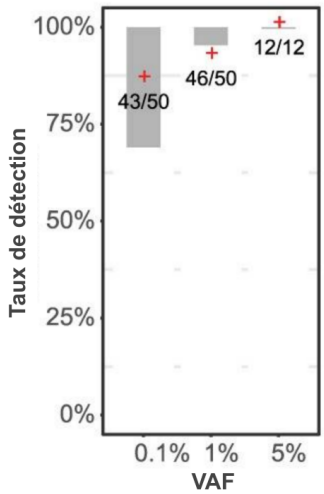


FIGURE 3.26 – Nombre de variants détectés pour chaque niveau de fréquence par MAGERI. Les zones ombrées montrent les intervalles de confiance à 95% pour la fraction attendue des variants récupérés. Figure adaptée de [57].

logiciel se sont attaqués au problème de la détection de l'ADN tumoral circulant (*ctDNA*) dans le sang périphérique des patients cancéreux. Des échantillons d'ADN de tumeur et de plasma sanguin de deux patients atteints de mélanome ont été séquencés en utilisant un protocole de préparation de librairie basé sur les UMI. Les résultats de la Figure 3.27 montrent le *variant calling* réalisé pour l'exon 15 du gène BRAF (les deux tumeurs étaient connues pour abriter la mutation du chromosome 7 dans le gène BRAF g.140453136A>T). La mutation g.140453136A>T a été détectée dans l'ADN plasmatique des deux patients à une fréquence de 0,4% et 3,3% respectivement. Notamment, le plasma du premier patient contenait aussi la mutation g.140453137C>T à une fréquence de 0,4%. Cette dernière est détectée conjointement (c'est-à-dire dans des groupes de *reads* ayant le même UMI) avec le variant g.140453136A>T. Le variant g.140453137C>T est également présent dans l'échantillon de tumeur correspondant, bien qu'à une fréquence beaucoup plus inférieure à celle du variant g.140453136A>T. La probabilité de détecter conjointement cette paire de mutations simplement par hasard est $P < 10^{-18}$, ainsi le premier patient démontre un cas intéressant d'une sous-population rare de cellules tumorales dominante dans son ADN tumoral circulant. Les résultats obtenus avec MAGERI montrent qu'il peut être utilisé dans un large éventail d'analyses en aval, telles que la détection des variants à très faible fréquence (jusqu'à 0,4%) ainsi que l'annotation des effets de variants surtout s'il est relié à des bases de données de variants telles que dbSNP ou COSMIC.

3.3.5 smCounter2

Chang Xu *et al.* ont présenté smCounter2 dans leur étude en 2019 [80], un *variant caller* basé sur les UMI et conçu spécifiquement pour la détection des SNV et des indels avec grande précision. L'outil smCounter2 adopte trois modèles statistiques différents pour détecter les erreurs dans les *reads* : la distribution bêta pour modéliser les taux d'erreur de séquençage, la distribution bêta-binomiale pour modéliser le nombre d'UMI portant une base alternative et un modèle de régression pour détecter les erreurs potentielles dans les régions homopolymériques. Une caractéristique importante de smCounter2 est que les paramètres du modèle sont ajustés dynamiquement pour chaque jeu de *reads* d'entrée. Il présente un seuil de détection de 0,5% et donc il est bien adapté aux expériences NGS pour la détection des variants rares. Les modèles statistiques utilisés par défaut sont spécifiques aux données QIAseq et donc, avant toute utilisation, de nouveaux modèles doivent être générés par l'utilisateur si les données utilisées proviennent d'un autre kit. Le *workflow* de smCounter2 est présenté dans la Figure 3.28. Le logiciel smCounter2 commence par des étapes de traitement des *reads* pour supprimer les adaptateurs, identifier la séquence UMI dans le *read* et l'ajouter à l'en-tête, et supprimer les *reads* trop courts. Les séquences restantes sont ensuite mappées sur le génome de référence avec BWA-MEM, suivi d'un filtrage des *reads* mal alignés. Un UMI avec un nombre de *reads* très petit est combiné avec une famille de *reads* beaucoup plus grande si leurs UMI sont à une distance d'édition de 1. Une fois le regroupement par UMI terminé, les *reads* sont produits sous format BAM pour commencer le *variant calling*. smCounter2 parcourt la région d'intérêt et traite chaque position indépendamment. À chaque position, les *reads* passent par plusieurs filtres de qualité et les *reads* de haute qualité restants sont regroupés pour former des *reads* consensus par UMI. Les variants potentiels détectés à cette étape sont ensuite soumis à des filtres supplémentaires comme les filtres pour le biais de brin et les homopolymères. Enfin, les variants sont annotés et produits dans un fichier VCF. Pour une meilleure flexibilité, les utilisateurs peuvent choisir d'exécuter uniquement le *variant calling* sur un fichier BAM déjà traité par le programme UMI-tools pour extraire les UMI et fgbio (https ://github.com/fulcrumgenomics/fgbio) pour construire les consensus.

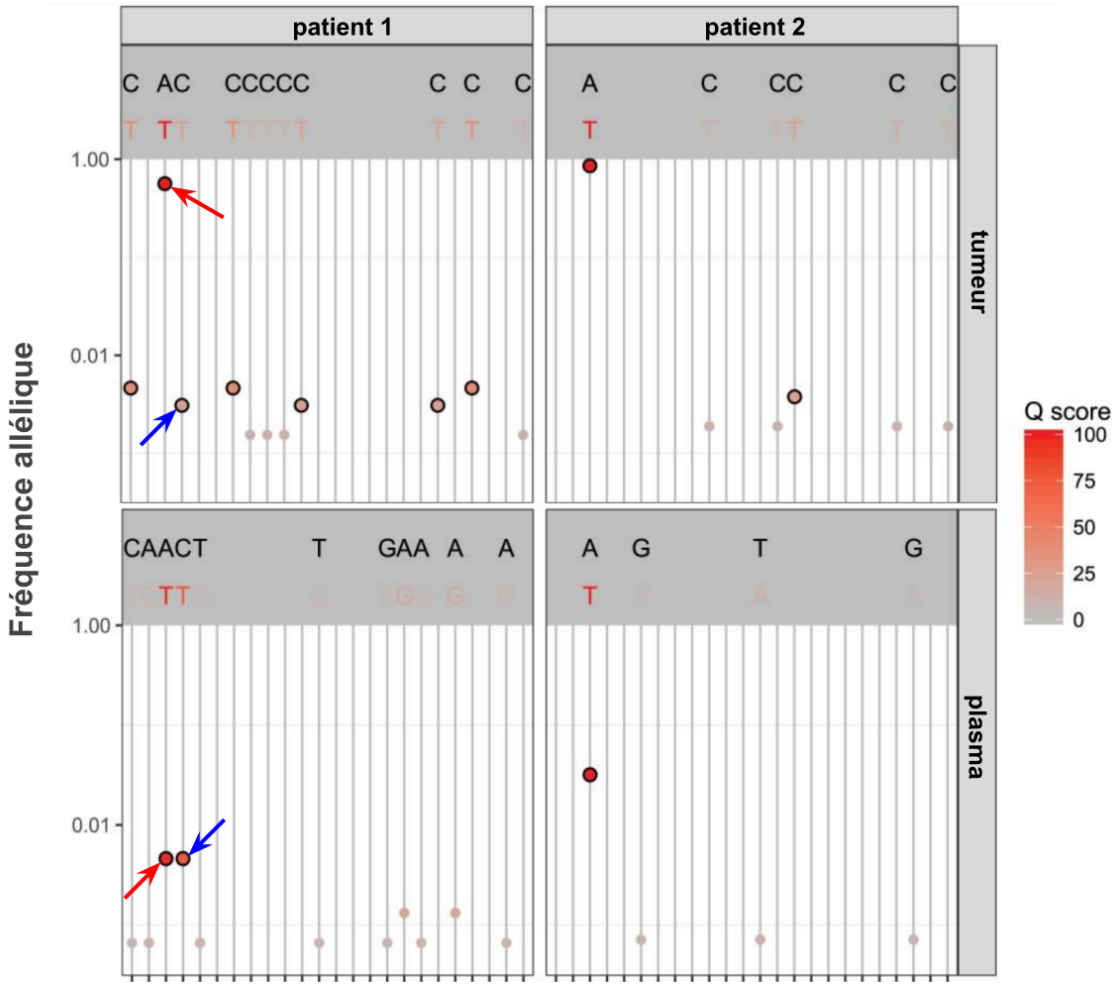


FIGURE 3.27 – Détection de variants du gène BRAF dans des échantillons de tumeurs et de plasma de deux patients cancéreux. Chaque point représente un variant et est coloré selon le score de qualité calculé par MAGERI, le panneau supérieur de chaque graphique montre les bases de référence (en haut) et alternatives (en bas). Les flèches rouges indiquent le variant g.140453136A>T alors que les flèches bleues indiquent le variant g.140453137C>T. Les variants dépassant le seuil Q 20 (P < 0,01) sont indiqués par des cercles en gras. Figure adaptée de [57].

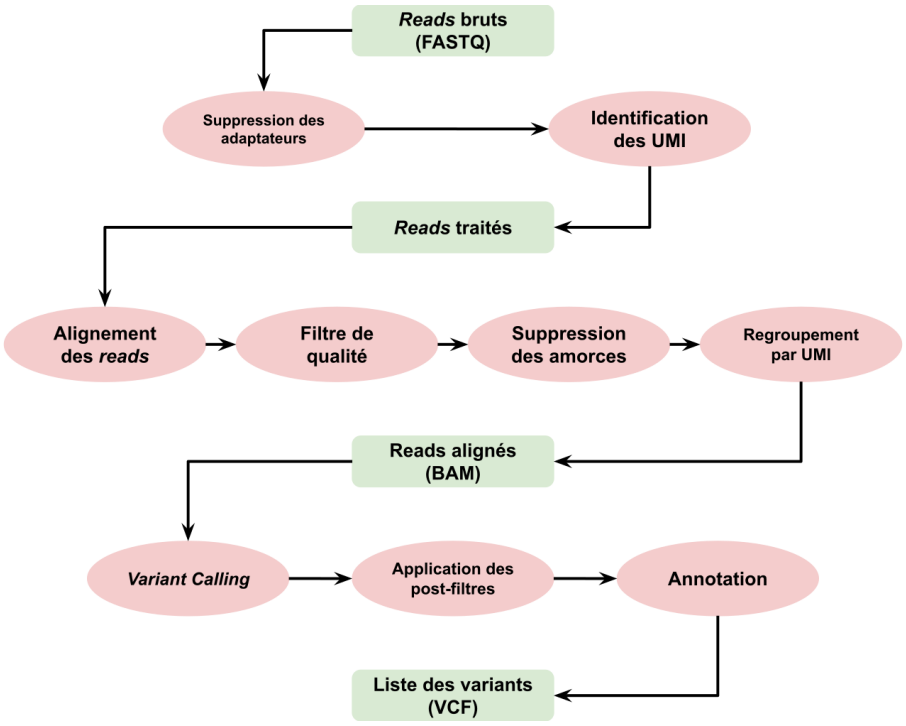


FIGURE 3.28 – *Workflow* de l'outil smCounter2. Figure adaptée de [80].

Les développeurs de smCounter2 l'ont comparé à quatre *variant callers* différents (fgbio + MuTect, fgbio + MuTect2, fgbio + VarDict et smCounter) sur un échantillon qui contenaient des variants à une fréquence de 0,5%. Les trois premiers algorithmes représentent l'approche en deux étapes consistant à utiliser l'outil fgbio pour construire les *reads* consensus et ensuite utiliser un *variant caller* conventionnel pour la détection des variants à faible fréquence comme MuTect, MuTect2 [53] et VarDict [84]. smCounter est l'ancienne version de smCounter2 capable de réaliser un *variant calling* basé sur les UMI. Les résultats de la Figure 3.29, stratifiés par type de variant (SNV et indel) et par région génomique (toutes, codantes et non codantes), ont été mesurés en calculant la sensibilité et le taux de faux positifs par mégabase (FP/Mbp) à plusieurs seuils : le Q-score pour smCounter2, l'indice de prédiction pour smCounter, rapport de vraisemblance pour MuTect et MuTect2 et la fréquence allélique minimale pour VarDict. smCounter2 a surpassé les autres méthodes dans toutes les catégories. Dans les régions codantes, smCounter2 a atteint une sensibilité de 92,4% à 12 FP/Mbp pour les SNV et de 84,4% de sensibilité à 7 FP/Mbp pour les indels. Dans les régions non codantes, smCounter2 a pu maintenir une précision comparable pour les SNV (83,3% sensible à 4 FP/Mbp), mais produit une sensibilité plus faible (56,8%) et un taux de faux positifs plus élevé (42 FP/Mbp) pour les indels. Ces résultats démontrent la grande efficacité de smCounter2 et le grand intérêt de son utilisation dans des études cliniques sur des échantillons de patients pour améliorer la détection des variants rares dans les tumeurs comme dans le plasma.

3.4 Synthèse

Dans ce chapitre, nous avons décrit l'état de l'art des principales publications se servant de l'étiquetage de l'ADN par des UMI pour mettre en évidence de nouveaux résultats et améliorer des résultats obtenus auparavant sans utilisation des UMI. Nous avons

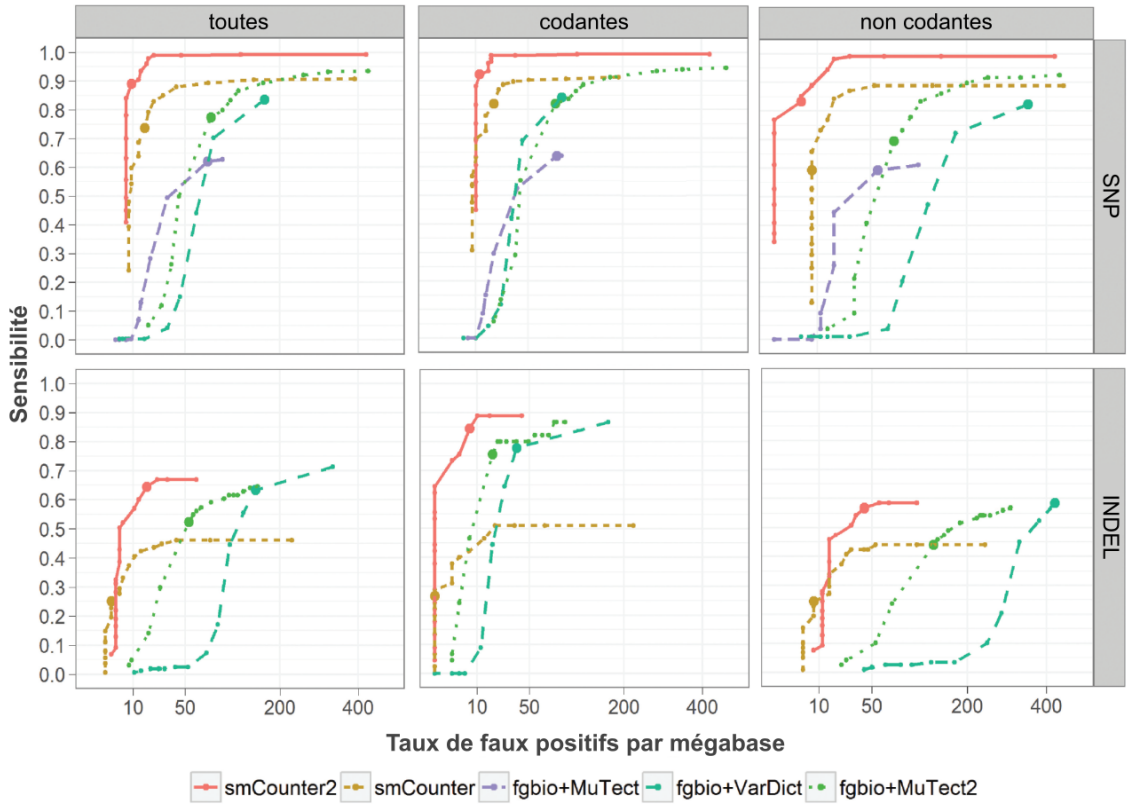


FIGURE 3.29 – Analyse comparative de smCounter2, smCounter, fgbio + MuTect, fgbio + VarDict et fgbio + MuTect2 sur des variants à 0,5%. Les performances sont mesurées en montrant la variation de la sensibilité en fonction du taux de faux positifs par mégabase, stratifiés par type de variant (SNV et indel) et par région génomique (toutes, codantes et non codantes). MuTect ne détecte pas les indels et n'est donc pas inclus dans la comparaison indel. Figure adaptée de [80].

présenté trois études où les UMI ont été utilisés dans du DNA-seq pour détecter une trisomie 21 par caryotypage digital, détecter des mutations *de novo* dans du *cfDNA* et pour démontrer que l'utilisation des UMI permet de réduire significativement le taux des doublons de PCR. D'autre part, deux études ont aussi été présentées dans des expériences de RNA-seq permettant de mettre en évidence un nouvel artefact de PCR jamais décrit auparavant ainsi que la suppression des doublons de PCR. Finalement, nous avons décrit aussi l'état de l'art des principaux outils se basant sur les UMI. Les outils UMI-tools, gencore, DeepSNVMiner, MAGERI et smCounter2 ont été présentés en détail. UMI-tools et gencore servent principalement à l'extraction et à la correction des erreurs dans les UMI et permettent donc une meilleure déduplication des *reads*. DeepSNVMiner, MAGERI et smCounter2 sont des *pipelines* complets et autonomes pour le traitement des données avec UMI et la détection des variants rares. Les trois outils ont des algorithmes distincts mais un peu similaires commençant par l'extraction des UMI, la suppression des adaptateurs et le filtrage des *reads* sur des critères de qualité. Les *reads* filtrés sont ensuite alignés, regroupés par UMI pour former des *clusters* et utilisés pour former des *reads* consensus dans lesquels les variants peuvent être appelés avec grande précision. Cependant, tous les trois *variant callers* présentés ont leurs limitations : DeepSNVMiner et MAGERI sont des *pipelines* complets et donc mettent un temps supplémentaire pour appliquer leurs algorithmes de traitement de séquence qui peut s'avérer dans certains cas, considérablement long. De plus, l'outil MAGERI consomme beaucoup de mémoire et avec smCounter2, nécessitent une modélisation du taux des erreurs spécifique à chaque kit d'analyse. Ces limitations nous ont motivés à développer un nouveau *variant caller* autonome, UMI-VarCal, qui implémente une nouvelle méthode de traitement des UMI dans les *reads*. D'autre part, l'absence d'un simulateur de *reads* avec UMI dédié nous a conduits à en développer un nous-même, UMI-Gen, et qui nous a servi à évaluer notre *variant caller* contre les autres outils. Ces deux nouveaux outils représentent la partie la plus importante du travail réalisé pendant cette thèse et seront présentés chacun, en détail, dans les chapitres suivants.

Chapitre 4

RT-MLPA et séquençage NGS

4.1 Introduction

Dans le Chapitre 3, nous avons présenté quelques travaux antérieurs dans lesquels les UMI ont été utilisés avec grand succès pour améliorer les résultats d'une expérience NGS pour séquencer de l'ADN ou de l'ARN. Dans le domaine de la transcriptomique, l'étude et le séquençage de l'ARN ont servi dans plusieurs applications telles que la mesure de l'expression génique et la détection des transcrits de fusion. La mesure de l'expression génique sert à quantifier le nombre de molécules d'ARN afin de distinguer entre les gènes surexprimés et les gènes sous-exprimés. Ceci est très utile pour comprendre les différentes voies métaboliques à l'origine des différents cancers ou des différents sous-types d'un même cancer. De plus, afin de permettre la détection précoce du cancer et donc d'une altération dans le génome, la détection des molécules chimères issues d'une recombinaison entre des parties de différents chromosomes peut être appliquée. Ces deux applications sont très utilisées au Centre Henri Becquerel au sein de l'unité Inserm 1245 et plus particulièrement dans l'équipe dirigée par le professeur Fabrice JARDIN et spécialisée dans l'étude des lymphomes à l'échelle moléculaire. Cette équipe utilise la RT-MLPA couplée à un séquenceur Illumina MiSeq pour séquencer les échantillons d'ARN des patients. Suite à ce séquençage, une analyse bioinformatique permet de quantifier l'ARN du patient, déduire le sous-type de son cancer et même détecter des transcrits de fusion permettant une prise en charge plus personnalisée et un traitement très spécifique et certainement plus efficace.

Dans ce qui suit, nous commencerons par une petite description des différents types et sous-types des lymphomes pour situer le contexte biologique de ce travail. Ensuite, nous allons présenter la méthode de quantification par RT-MLPA de première génération et le passage à une variante plus avancée couplée à un séquenceur NGS. Cette dernière nécessite une analyse bioinformatique plus poussée et un système de classification robuste et fiable pour assurer une classification précise des échantillons. Ces besoins nous ont conduits à développer l'outil et l'interface RT-MiS permettant de prendre en charge toute l'analyse bioinformatique et produisant les résultats dans un format clair et facile à interpréter.

4.2 Les lymphomes

Les lymphomes sont des tumeurs hétérogènes du système lymphatique qui se développent aux dépens des lymphocytes B ou T, cellules jouant un rôle essentiel dans les réactions de défense immunitaire. Selon leur nature, les lymphomes sont dits hodgkiniens ou non hodgkinens, et ont des degrés de gravité variables. Ce sont des cancers relativement fréquents puisqu'ils se placent en France au sixième rang en terme d'incidence (4,8 cas pour 100 000 personnes) et au premier rang des cancers chez les adolescents et jeunes

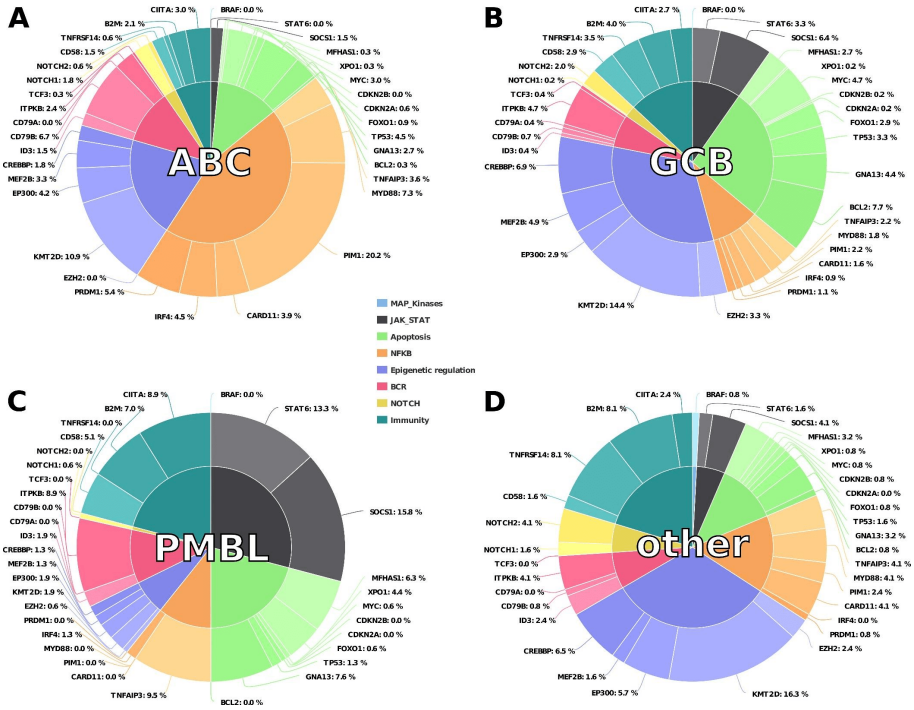


FIGURE 4.1 – Pourcentage des variants impactant les différents gènes en fonction des sous-types ABC, GCB, PMBL et indéterminés. Les gènes sont regroupés en grandes voies métaboliques [85].

adultes (15-25 ans). Le travail effectué pendant cette thèse a porté principalement sur les deux types de lymphome B et T. Dans ce qui suit, les différents types de lymphomes seront expliqués en détaillant le phénotype général de chacun. Selon la nomenclature officielle, le nom des gènes est écrit en italique alors que le nom de la protéine correspondante s'écrit en romain. Un phénotype présente une liste de marqueurs (et donc de protéines) surexprimés (indiqués par un "+") et sous-exprimés (indiqués par un "-") caractérisant chaque entité.

4.2.1 Le lymphome B diffus à grandes cellules

Le lymphome diffus à grandes cellules B, ou DLBCL (*Diffuse Large B-Cell Lymphoma*) est le lymphome le plus fréquent, représentant environ 40% de l'ensemble des cas. L'analyse des profils d'expression génique (GEP) a permis de démontrer l'hétérogénéité de ces tumeurs, et a permis d'identifier 3 sous-types moléculaires présentés dans la Figure 4.1.

Le premier sous-type, appelé ABC (*Activated B-Cell like*), serait issu de la transformation de lymphocytes B matures au stade plasmablastique. Les anomalies génétiques de ces tumeurs sont hétérogènes, mais environ 30 à 40% des cas présentent une translocation du gène *BCL6*. Le gène *CDKN2A* y est également délété de façon récurrente, et le gène *BCL2* amplifié. Le développement des techniques de séquençage de nouvelle génération a également permis de démontrer, notamment grâce à des études menées au sein du CHB [85], que ces tumeurs portent souvent des mutations de gènes impliqués dans la voie de signalisation NF-κB (dans environ 45% des cas), avec notamment des mutations des gènes *MYD88*, *CD79B* et *CARD11*.

Le deuxième sous-type, appelé GCB (*Germinal Center B-cell like*), serait issu des lymphocytes B plus immatures, issus des centres germinatifs des organes lymphoïdes secondaires. Il est également associé à des anomalies génétiques qui lui sont spécifiques,

comme la translocation t(14;18)(q32;q21) ou la perte du gène *PTEN*. Contrairement aux DLBCL de type ABC, ils présentent souvent des mutations de gènes impliqués dans les modifications épigénétiques de l'ADN (en particulier du gène *EZH2*), ainsi que dans les voies de l'apoptose/cycle cellulaire (26,3%).

Enfin, le troisième sous-type appelé PMBL (*Primary Mediastinal B-cell Lymphoma*), correspond à des tumeurs qui auraient pour origine des lymphocytes B thymiques. Ces cancers s'observent souvent chez des femmes jeunes, se développent au niveau du médiastin, et sont de pronostic plus favorable. Ils présentent fréquemment des amplifications des gènes *JAK2* et *PD1*, et des délétions de *SOCS1*. Les mutations qui leurs sont associées touchent souvent des gènes impliqués dans la voie de signalisation JAK-STAT, et en particulier le gène *STAT6*.

4.2.2 Le lymphome B à petites cellules

Le lymphome B à petites cellules se distinguent des DLBCL puisque la maladie évolue plus lentement en général et est moins agressive. Cependant, cette forme de lymphome peut évoluer en DLBCL. Comme pour le lymphome B à grandes cellules, ce type de lymphome est composé de plusieurs sous-groupes. Les quatre sous-groupes ont été mise en évidence grâce à des études d'analyse moléculaire et histopathologique démontrant l'hétérogénéité de ces tumeurs.

Le premier sous-type est le lymphome folliculaire, ou FH (*Follicular Lymphoma*) [86]. Aujourd'hui, le lymphome folliculaire représente entre 20 et 25% des cas de lymphome non hodgkinien diagnostiqués, et donc environ 3100 nouveaux cas par an en France. La maladie touche principalement les personnes ayant plus de 60 ans (médiane d'âge de survenue 60 - 65 ans). La majorité des patients présente des ganglions superficiels touchant la plupart des aires ganglionnaires. C'est une prolifération clonale de cellules B du centre germinatif (CD10+, CD20+ et CD5-), du follicule du ganglion lymphatique. Les tumeurs ont généralement un phénotype de marqueurs CD5-, CD9+, CD19+, CD20+, CD21+, CD22+, CD24+, CD79a+, BCL-2+ et BCL-6+. Une anomalie chromosomique est présente chez une très grande partie des patients : en biologie moléculaire, on note un réarrangement *BCL-2/IgH* tandis que sur le cayotype, une translocation chromosomique t(14;18) (q32;21) est présente.

Le deuxième sous-type s'appelle le lymphome lymphocytique ou SLL (*Small Lymphocytic Lymphoma*) [87]. La forme clinique de ce lymphome se distingue par le développement initial de la tumeur dans un organe lymphoïde secondaire. C'est une maladie qui évolue lentement avec une grande chance de guérison pour les formes localisées. La population tumorale de ces lymphocytes B présentent un phénotype CD5+, CD20+ et CD23+.

Le troisième sous-type est le lymphome à cellules du manteau ou MCL (*Mantle Cell Lymphoma*) [88]. Cette forme de lymphome est relativement rare (moins de 1000 cas par an en France) représentant 8 à 10% des lymphomes B. Elle est très rare chez les personnes ayant moins de 50 ans et touche généralement les hommes trois fois de plus que les femmes. La maladie est souvent diagnostiquée à des stades avancés avec des polyadénopathies et une atteinte splénique accompagnée d'une dissémination dans la moelle osseuse. Les lymphomes du manteau sont une prolifération clonale de cellules B très particulières de la couronne entourant les follicules normaux. Le marqueur de prolifération Ki67 est exprimé dans 20% des cellules. La translocation t(11;14)(q13;q32) touchant le gène *BCL-1* a été mis en évidence par les analyses cytogénétiques. Cette translocation place le proto-oncogène *CCND1* en 11q13 à côté d'une chaîne lourde d'immunoglobuline. En conséquence, une protéine impliquée dans le cycle cellulaire, la cycline D1, est surexprimée.

Enfin, le lymphome de la zone marginale ou MZL (*Marginal Zone Lymphoma*) est un lymphome non hodgkinien représentant 11% de l'ensemble des LNH [89]. C'est un groupe hétérogène de lymphomes caractérisés par une prolifération nodulaire puis diffuse de petites cellules caractérisées par un phénotype CD5-, CD10-, CD20+ et CD23-. Cette famille comprend les lymphomes des tissus associés aux muqueuses en association avec une infection à *H. Pylori*, *B. Burgdorferi* ou *C. Jejuni*, les formes spléniques associées à une infection par le virus de l'hépatite C et les formes ganglionnaires envahissant exclusivement la moelle osseuse et les ganglions.

4.2.3 Le lymphome T

Les lymphomes T périphériques représentent 6,3% des lymphomes non hodgkiniens, soit 1600 nouveaux cas/an en France [90]. La classification OMS 2016 répertorie 27 entités de néoplasies lymphoïdes T matures [91]. L'hétérogénéité et la rareté de ces tumeurs complexifient le diagnostic histopathologique, en l'absence de marqueur immunophénotypique spécifique et d'une reproductibilité variable entre pathologistes [92, 90].

L'AITL (*AngioImmunoblastic T-cell Lymphoma*) touche essentiellement les adultes âgés (âge médian de 60 ans), présentant des adénopathies périphériques diffuses, des signes généraux, une atteinte extra-ganglionnaire fréquente, ainsi que des anomalies biologiques hématologiques et immunologiques (Figure 4.2A). Le diagnostic histopathologique est caractérisé par un infiltrat cellulaire diffus, constitué de lymphocytes T tumoraux de taille moyenne aux cytoplasmes clairs, de phénotype TFH (*T Follicular Helper*) BCL6+, PD1+, CXCL13+, ICOS+, CD10+ et CXCR5+. Les cellules tumorales sont souvent peu nombreuses, dispersées au sein d'un microenvironnement inflammatoire polymorphe, constitué de petits lymphocytes B et T réactionnels, d'histiocytes, de polynucléaires éosinophiles, de plasmocytes et de grands lymphocytes B infectés par l'EBV (*Epstein-Barr Virus*). Cet infiltrat s'accompagne d'une hyperplasie des veinules postcapillaires et d'une expansion du réseau de cellules folliculaires dendritiques (qui expriment également *CXCL13*). Récemment, les techniques de séquençage haut débit ont identifié des mutations de gènes impliqués dans la régulation des mécanismes épigénétiques, notamment *DNMT3, TET2 et IDH2*, ainsi que des mutations récurrentes du gène *RHOA* [93, 94, 95, 96, 97].

L'ALCL (*Anaplastic Large Cell Lymphoma*) ALK+ est plus fréquemment observé chez le sujet jeune, avec une légère prédominance masculine, et se manifeste par des adénopathies diffuses, des signes généraux, et une atteinte extra-ganglionnaire fréquente (Figure 4.2B). Le pronostic est favorable comparé aux autres entités, avec une survie globale à 5 ans de 58-70%. L'ALCL ALK+ est caractérisé par une prolifération de cellules tumorales cohésives de grande taille, au noyau proéminent excentré, en forme de fer à cheval, appelées *hallmark*. En immunohistochimie, les cellules tumorales expriment fortement *CD30*, les marqueurs de cytotoxicité (Tia1, perforine et granzyme B) et *BCL6*. Le diagnostic est posé par la positivité de l'immunomarquage pour la protéine ALK, due à une translocation chromosomique impliquant le gène *ALK* situé sur le chromosome 2 [98].

L'ALCL ALK- a été individualisé dans la classification OMS en 2008. Cette entité concerne l'adulte d'âge moyen (40-65 ans). L'atteinte ganglionnaire est prédominante. Le pronostic est plus péjoratif que celui de l'ALCL ALK+, avec une survie globale à 5 ans comprise entre 34 et 49%. Le diagnostic d'ALCL ALK- repose sur une morphologie identique à l'ALCL ALK+ ainsi qu'une expression intense et homogène de *CD30* [98]. L'expression des marqueurs cytotoxiques n'est pas requise pour le diagnostic. Il s'agit d'un groupe très hétérogène sur le plan moléculaire, avec une valeur pronostique différente selon les anomalies moléculaires impliquées [99].

La leucémie/lymphome T de l'adulte (ATLL) est essentiellement observée dans les régions endémiques où il existe une forte prévalence de l'infection par le virus HTLV-1 au sein de la population (Figure 4.2C). L'infection par le rétrovirus HTLV-1 survient précocement au cours de la vie. L'incidence est estimée à 2 à 4% parmi les patients infectés après une longue période de latence, secondairement à une intégration clonale dans le génome. L'infection seule par HTLV-1 n'est pas suffisante au développement de l'ATLL et nécessite la survenue d'évènements secondaires. L'oncogenèse est initiée par l'expression du gène *TAX* et du domaine HBZ (*HTLV-1 Basic leucine Zipper factor*). Il existe quatre formes cliniques : leucémique ou aiguë, lymphomateuse, chronique, et indolente. Le spectre des lésions histopathologiques ganglionnaires de l'ATLL est extrêmement variable, pouvant mimer n'importe quel autre type de PTCL. La forme leucémique est caractérisée par la présence de cellules tumorales circulantes. Ces cellules tumorales ont un phénotype CD25+, FOXP3+ et CCR4+ reflétant l'origine T régulatrice (Treg), mais aussi GATA3+ induit par HBZ [100].

Les lymphomes NK/T (NKTCL) touchent les sujets d'âge moyen (40 ans), le plus souvent de sexe masculin, originaires d'Asie ou d'Amérique du Sud. L'atteinte est essentiellement extra-ganglionnaire de type nasal, concernant le plus souvent le tractus aérodigestif supérieur (Figure 4.2E). Les cellules tumorales sont dérivées de cellules NK ou de lymphocytes T infectés par le virus EBV, ayant un phénotype cytotoxique activé (Tia1+, perforine+ et granzyme B+) en immunohistochimie, et exprimant *CD56* de façon inconstante. La survie est de 50% à 5 ans [101].

Le lymphome T hépatosplénique (HSTL) est un lymphome extra-ganglionnaire rare et agressif, affectant les sujets jeunes (35 ans) dans un contexte d'immunodépression ou de stimulation antigénique chronique (Figure 4.2F). Ce lymphome est caractérisé par une infiltration tumorale des sinus du foie, de la rate et de la moelle osseuse par des lymphocytes T de phénotype CD3+, CD5-, CD56+, Tia1+ et granzyme B- [101].

Et finalement, les PTCL-NOS (*Peripheral T-Cell Lymphoma, Not Otherwise Specified*) constituent une entité hétérogène, regroupant l'ensemble des cas qui ne correspondent à aucun des sous-groupes définis par la classification OMS (environ 26,9%). Il s'agit d'un diagnostic d'exclusion, en l'absence de marqueur histologique ou moléculaire spécifique. Ces dernières années, plusieurs études d'expression génique ont permis d'identifier différentes catégories de PTCL-NOS :

— 14 à 30% des PTCL NOS expriment des marqueurs TFH, et présentent une signature d'expression génique de type TFH. De plus, des mutations des gènes *TET2, DNMT3A et RHOA*, fortement associées à l'AITL, sont également retrouvées dans une proportion variable de PTCL-NOS, justifiant le regroupement de ces cas avec les AITL dans la classification OMS 2016.

— Une proportion variable des PTCL-NOS expriment CD30 en immunohistochimie. Le diagnostic entre PTCL-NOS CD30+ et ALCL ALK- n'est pas toujours évident, du fait de critères histologiques subjectifs, notamment le caractère anaplasique de la cellule tumorale, ainsi que l'interprétation du marquage immunohistochimique anti-CD30. Ces cas représentent une partie du groupe CD30TH2 (Voir Figure 4.2D) observé en RT-MLPA.

— Certains PTCL-NOS expriment les marqueurs de cytotoxicité (Tia1, granzyme B et perforine) sur plus de 30% des cellules tumorales, et sont associés à un pronostic péjoratif. Cette proportion représente le groupe PTCL cytotoxique.

— Deux sous-groupes ontogéniques de LTP-NS ont été proposés sur la base de profils d'expression génique et/ou de l'expression des marqueurs Th1 (*TBX21*) et Th2 (*GATA3*) [102, 103, 104].

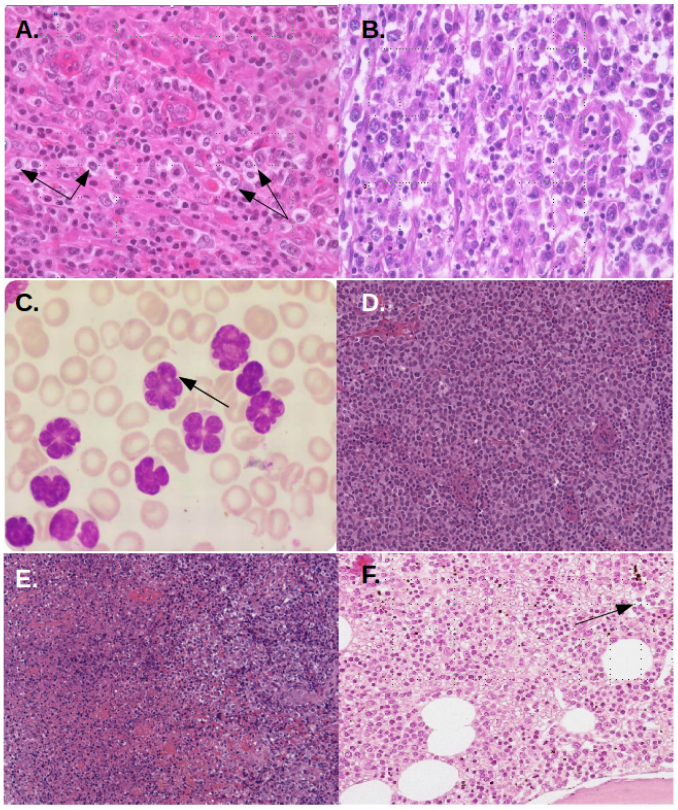


FIGURE 4.2 – Les différents sous-types de lymphomes T. A. AITL (les flèches montrent les cellules tumorales). B. ALCL (seules les cellules tumorales sont représentées). C. ATLL (les flèches montrent les cellules tumorales). D. CD30TH2 (seules les cellules tumorales sont représentées). E. NKTCL (les cellules tumorales sont dans la partie droite de l'image). F. HSTL (la flèche montre un exemple de cellule tumorale).

4.3 Analyse par RT-MLPA classique

4.3.1 Principe de la RT-MLPA

Les tests diagnostiques qui ont été développés au laboratoire pour la classification des lymphomes sont basés sur une méthode de type RT-MLPA (*Reverse Transcriptase-Multiplex Ligation-dependant Probe Amplification*). La RT-MLPA est une variation de la réaction de polymérisation en chaîne quantitative (qPCR), qui permet d'atteindre un haut degré de multiplexage. Elle a été décrite initialement par Eldering *et al.* en 2003 [105] et elle est basée sur l'utilisation de paires d'oligonucléotides spécifiques pour chaque ADNc ciblé, et qui reconnaissent des séquences adjacentes (Figure 4.3). L'ADNc est l'ADN complémentaire obtenu lors d'une rétrotranscription d'un fragment d'ARN par la transcriptase inverse. Lorsque ces deux oligonucléotides sont hybridés à leur cible, ils peuvent être ligués pour former une sonde complète. Tous les oligonucléotides de gauche portent en 5' une extension d'une vingtaine de paires de bases identiques. De la même façon, tous les oligonucléotides de droite portent en 3' une extension du même type, mais dont la séquence est différente. L'avantage de diviser la sonde en deux parties est que seuls les oligonucléotides ligués, et non les oligonucléotides libres, vont pouvoir être amplifiés par PCR en utilisant ces queues additionnelles comme amorces. Si les sondes n'étaient pas divisées de cette façon, les séquences des extrémités permettraient l'amplification des sondes indépendamment de leur hybridation à l'ADNc, et le produit d'amplification ne dépendrait pas du nombre de sites cibles présents dans l'échantillon. Pour tous les tests développés au sein du CHB, chaque sonde complète a une longueur unique, de sorte que les différents amplicons puissent être séparés par électrophorèse. L'une des deux amorces de PCR étant marquée par un fluorochrome, chaque amplicon génère un pic fluorescent qui peut être détecté par un séquenceur capillaire. En comparant les pics obtenus pour un échantillon donné avec ceux obtenus pour des échantillons de référence, on peut déterminer la quantité relative de chaque amplicon. L'intensité de fluorescence est ainsi proportionnelle à la quantité d'ADNc ciblé, et donc au niveau d'expression de la cible.

4.3.2 Analyse bioinformatique

Un logiciel a été développé, en R, pour gérer les fichiers FSA (extension .fsa) produits par les séquenceurs capillaires d'Applied Biosystems dans le cadre des expériences RT-MLPA. Pourvu d'un fichier de configuration approprié, il normalise le signal, infère les dimensions maximales et calcule les probabilités d'appartenir à une des classes précisées dans le fichier de configuration. Il a été largement testé sur un analyseur génétique 3130, mais il devrait être compatible avec n'importe quel séquenceur produisant des fichiers FSA.

4.3.2.1 Le fichier FSA

Le fichier FSA est un fichier binaire de type *application/octet-stream* (extension .fsa) contenant toutes les informations (date et temps de lancement, les hauteurs des pics captés, le nom de l'utilisateur, le nom de la machine ainsi que le temps d'injection) concernant un run déterminé.

4.3.2.2 Le fichier de configuration

Le fichier de configuration est un fichier texte normal (extension .conf) comprenant au moins huit sections représentant chacune le nom d'une fonction du logiciel et contient

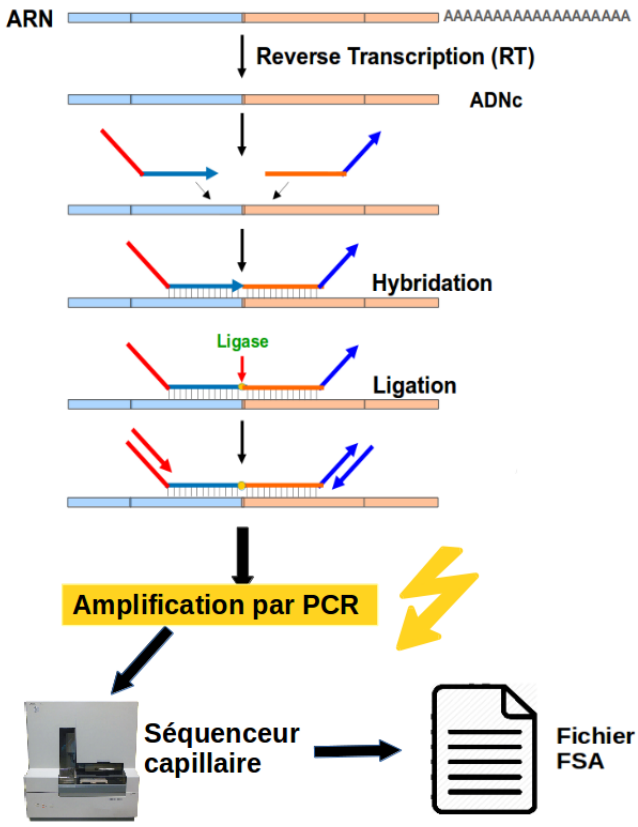


FIGURE 4.3 – Schéma représentant le principe de la RT-MLPA.

des paramètres que l'utilisateur peut changer selon son fichier FSA et ses préférences. Le fichier de configuration contient aussi les paramètres d'un modèle bayésien sur lesquels se base le logiciel pour classer un échantillon déterminé. Les paramètres de cette section sont le résultat d'un entraînement d'un modèle bayésien sur les valeurs exactes d'expression de certains gènes, permettant une classification précise des échantillons.

4.3.2.3 Les fichiers résultats

Après avoir choisi un (ou plusieurs) fichier(s) FSA avec le fichier de configuration approprié, l'analyse peut être lancée. L'analyse prend environ 1 seconde/fichier et produit comme résultat un fichier PDF contenant le profil d'expression du fichier analysé. En effet, le graphe obtenu montre l'intensité de la fluorescence capturée pour chaque ADNc ayant une taille comprise entre 80 et 120 pb. De plus, à droite, on obtient un calcul de score bayésien avec la probabilité d'appartenir à chacune des classes. En effet, le modèle de classification bayésien de type LPS (*Linear predictor Score*) est seulement appliqué dans le cas de classification des lymphomes B entre deux/trois classes différentes. Cependant, pour les lymphomes T, vu que plusieurs sous-types existent, le modèle bayésien ne peut pas être utilisé et le logiciel a recours à un modèle de classification par SVM (*Support Vector Machine*). Dans les deux cas, le modèle se base sur l'expression d'une vingtaine de gènes (spécifiques à chaque type de lymphome) pour effectuer la prédiction. Un exemple des résultats obtenus pour chaque type d'analyse est présenté dans la Figure 4.4.

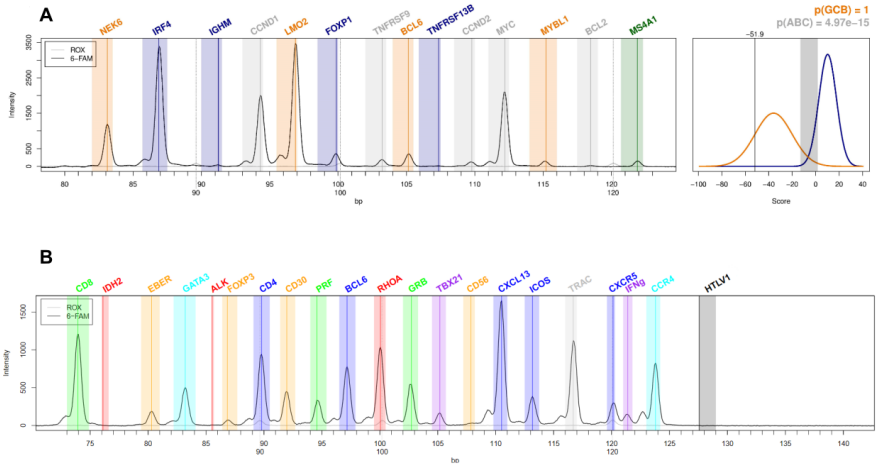


FIGURE 4.4 – Profils d'expression génique de deux échantillons de lymphome. (A) Échantillon de lymphome B : la classification à droite montre qu'il appartient au sous-type GCB. (B) Échantillon de lymphome T : la classification est de type SVM et elle n'est pas montrée dans cette figure.

4.4 RT-MLPA couplée à un séquenceur NGS

4.4.1 Principe

La RT-MLPA classique présentait deux limitations principales : le nombre de gènes pouvant être mesurés en même temps était limité à une vingtaine et un phénomène se saturation de signal qui se produisait lorsqu'un des gènes est considérablement surexprimé. Pour cela, l'équipe de recherche au CHB a cherché à résoudre ces deux problèmes en remplaçant le séquenceur capillaire par un séquenceur NGS de type Illumina MiSeq. Le principe de la nouvelle méthode est illustré par la Figure 4.5. Les premières étapes sont identiques à la RT-MLPA classique. Une des différences concerne le nombre de sondes pouvant être utilisées simultanément, plusieurs centaines au lieu d'une vingtaine. La principale évolution consiste en l'utilisation d'oligonucléotides modifiés à l'étape d'amplification par PCR. Au lieu d'amorces classiques, des oligonucléotides rallongés portant des queues additionnelles permettant l'analyse sur le séquenceur Illumina MiSeq ont été ajoutées. Pour permettre le séquençage de plusieurs échantillons lors d'une même analyse de séquençage, appelée *run*, et ainsi diminuer les coûts, ces oligonucléotides portent également des séquences index qui permettent d'attribuer les séquences détectées aux différents échantillons analysés. Contrairement au séquenceur capillaire, qui renvoie des intensités de fluorescence proportionnelles aux niveaux d'expression des gènes, le séquenceur Illumina analyse plusieurs dizaines de milliers de courtes séquences individuelles pour chaque patient, chacune correspondant à une molécule d'ADNc différente détectée dans l'échantillon de départ. L'analyse de ces séquences et leur comptage à l'aide d'outils bioinformatiques dédiés permet d'évaluer directement le niveau d'expression de chacun des marqueurs. En tenant compte de toutes ces informations, l'expérience a été construite de façon à obtenir des *reads* avec une configuration constante et optimale pour l'analyse bioinformatique. Nous avons ajouté des UMI de longueur 7 pb au début des *reads*, permettant une analyse bioinformatique et une interprétation biologique plus poussées et surtout plus précises. Ainsi, les *reads* générés par cette méthode seront composés de cinq parties principales illustrées dans la Figure 4.5 :

— la séquence aléatoire de l'UMI de longueur 7 pb ;

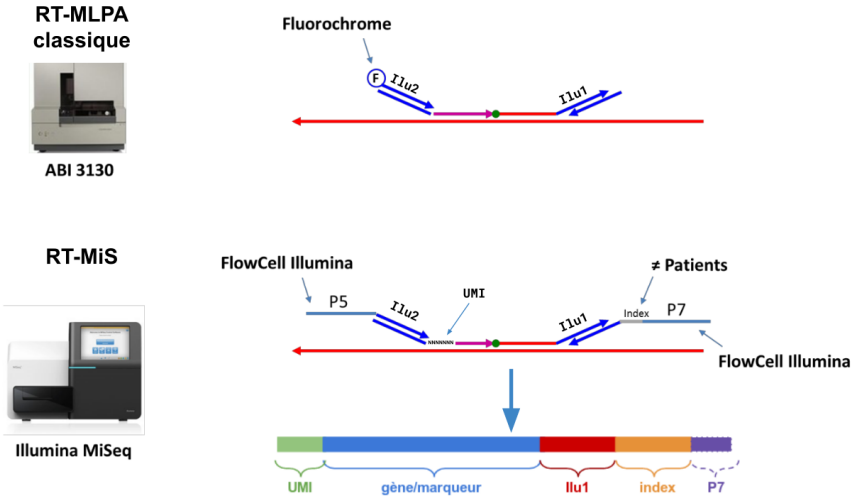


FIGURE 4.5 – Schéma représentant la différence entre une analyse RT-MLPA classique et une analyse RT-MLPA couplée à un séquenceur NGS.

— la séquence d'un des marqueurs ciblés ;
— la séquence de Ilu1 présente à l'extrémité 3' de toutes les sondes de RT-MLPA (séquence : TCCAACCCTTAGGGAACCC) ;
— la séquence de l'index/*barcode* ;
— quelques bases résiduelles correspondant au primer Illumina MiSeq P7.

Selon l'expérience réalisée, un *run* peut produire entre 5 et 20 millions de *reads* dans un fichier FASTQ. Le fichier FASTQ doit ensuite subir plusieurs traitements bioinformatiques afin de retrouver efficacement les séquences de chaque patient, extraire et corriger les UMI, supprimer les doublons de PCR et produire les résultats dans un format facilement interprétable.

4.4.2 Analyse bioinformatique

4.4.2.1 Le fichier d'index

Le fichier d'index est obligatoire pour lancer l'analyse. Ce fichier est sous format CSV composé de deux colonnes uniquement : le nom de l'index et sa séquence. Ce fichier est nécessaire afin de retrouver l'identité du patient auquel appartient la ou les séquences en cours d'analyse. Un exemple d'un fichier d'index est présenté dans la Figure 4.6

4.4.2.2 Le fichier des marqueurs

Comme le fichier d'index, le fichier des marqueurs est sous format CSV également mais il est composé de trois colonnes : le nom du marqueur, la séquence de l'amorce gauche et la séquence de l'amorce droite. La séquence de l'amorce à gauche commence toujours par une série de la lettre N indiquant l'emplacement normal de l'UMI et sa longueur. Ce fichier est également essentiel à l'analyse qui va chercher à retrouver les séquences présentes dans ce fichier dans les *reads* du fichier FASTQ. Un exemple d'un fichier des marqueurs est présenté dans la Figure 4.7.

indexName	sequence
D701	ACCCCGAGTAAT
D702	ACCCTCTCCGGA
D703	ACCCAATGAGCG
D704	ACCCGTAGCAGT
D705	ACCCGACGTAGC
D706	ACCCATGATGCG
D707	ACCCCATGTAGA
D708	ACCCAATGGACG
D709	ACCCGAGACTGA
D711	ACCCCCAAGTAG

FIGURE 4.6 – Un exemple du fichier d'index.

markerName	left	right
AIDe2-3	NNNNNNNTCACTGGACTTTGGTTATCTTCGCAATAAG	AACGGCTGCCACGTGGAATTGC
AIDe4-5	NNNNNNNAGACAGCTTCGGCGCATCCTTTTG	CCCCTGTATGAGGTTGATGACTTACGAGACG
ALK	NNNNNNNCCTCCGAGAGACCCGCCCTCGCCCG	AGCCAGCCCTCCTCCCTGGCCATGC
ANXA1	NNNNNNNCTGCCTTGCATAAGGCCATAATGGTTAAAG	GTGTGGATGAAGCAACCATCATTGACATTC
APRIL	NNNNNNNGTTCCCATTAACGCCACCTCCAAGG	ATGACTCCGATGTGACAGAGGTGATGTG
ASB13	NNNNNNNGCACGAGGCCTGCATGAGCG	GGAGTTCCGAATGTGTGAGGCTTCTTATTG
B2M	NNNNNNNCTTTGTCACAGCCCAAGATAGTTAAGTGGG	ATCGAGACATGTAAGCAGCATCATGGAG
BAFF	NNNNNNNAGCTGTCACCGCGGGACTGAAA	ATCTTTGAACCACCAGCTCCAGGAGAAG
BANK	NNNNNNNGAAAAAGTGGCCTGGAAATGATTCAGCAG	GAGAAATTACGACAACTACGAGACTGCATT
BCL2e1b-2b	NNNNNNNAGAGGATCATGCTGTACTTAAAAAATACAA	CATCACAGAGGAAGTAGACTGATATTAACA
BCL2e1-2	NNNNNNNCCTGGATCCAGGATAACGGAGGCTGG	GATGCCTTTGTGGAACTGTACGGCC
BCL6e1-2	NNNNNNNAAGAAGTTTCTAGGAAAGGCCGGACACCAG	GTTTTGAGCAAAATTTTGGACTGTGAAGCA
BCL6e3-4	NNNNNNNCATAAAACGGTCCTCATGGCCTGCAG	TGGCCTGTTCTATAGCATCTTTACAGACCAGTTG
BCMA	NNNNNNNCTCTAACATGTCAGCGTTATTGTAATGCAA	GTGTGACCAATTCAGTGAAAGGAACG
BRAFV600E	NNNNNNNAAAAATAGGTGATTTTGGTCTAGCTACAGA	GAAATCTCGATGGAGTGGGTCCC
CARD11	NNNNNNNCCACTCGGAGATTCTCCACCATTGTGG	TGGAGGAAGGCCACGAGGGCC
CCDC50	NNNNNNNGACGACGCATTCAGGAGAAGAAGGATGAG	GACATAGCTCGCCTTTTGCAAGAAAAGGAG
CCND1	NNNNNNNACCTTCGTTGCCCTCTGTGCCACAG	ATGTGAAGTTCATTTCCAATCCGCCCT
CCND2	NNNNNNNGGTGGCCACCTGGATGCTGGAG	GTCTGTGAGGAACAGAAGTGCGAAGAAGAG
CCR4	NNNNNNNCCTCAGAGCCGCTTTCAGAAAAGCAAG	CTGCTTCTGGTTGGGCCCAGACCT

FIGURE 4.7 – Un exemple du fichier des marqueurs.

4.4.2.3 Le fichier FASTQ

Le fichier FASTQ a déjà été décrit dans le Chapitre 2. Il est le produit normal d'un séquençage NGS, dans ce cas effectué par un séquenceur su type Illumina MiSeq. Les *reads* qui y sont présents respectent la configuration présentée dans la Figure 4.5.

4.4.2.4 Mesure de l'expression génique dans les lymphomes

La première application de cette nouvelle méthode d'analyse est la mesure de l'expression génique. En effet, grâce à cette technologie, nous avons pu passer de l'analyse de 21 gènes pour classifier les lymphomes B et T en RT-MLPA classique à 137 gènes actuellement. Ces gènes sont le fruit d'un immense travail biologique et statistique qui a permis de déterminer les plus discriminants entre les différents types et sous-types de lymphomes. L'ensemble de gènes appartenant à chaque type ou sous-type de lymphomes est présenté dans la Figure 4.8. Un modèle statistique de type *Random Forest* (RF) a été développé au sein de l'équipe et a été entraîné sur des centaines d'échantillons séquencés dans notre laboratoire.

Les *Random Forests* constituent une méthode d'apprentissage d'ensemble pour la classification qui signifie forêts aléatoires et qui est basé sur l'assemblage d'arbres de décision (*Decision Trees*). Un arbre de décision est un outil d'aide à la décision représentant un ensemble de choix sous la forme graphique d'un arbre. Les différentes décisions possibles (ou feuilles) sont situées aux extrémités des branches, et sont atteintes en fonction de décisions prises à chaque étape. Son plus grand avantage est qu'il est lisible et rapide à exécuter. En revanche, son problème principal est que sa performance est fortement dépendante de l'ensemble de données utilisé pour l'entraînement. Par conséquent, l'ajout de quelques nouveaux échantillons peut modifier radicalement les résultats et le modèle. Pour résoudre ce problème, une *Random Forest* se base sur le principe suivant (Figure 4.9) : au lieu d'avoir un seul modèle compliqué, il utilise plusieurs estimateurs simples et indépendants. Chaque estimateur est représenté par un arbre de décision unique grâce à un double tirage aléatoire (sur les observations et sur les variables). Ensuite, l'ensemble de ces estimateurs est réuni pour obtenir la vision globale du problème. C'est l'assemblage de tous ces estimateurs qui rend la prédiction très précise. À la fin, tous ces arbres de décisions indépendants sont assemblés et la prédiction faite par la *Random Forest* pour un nouvel échantillon est alors obtenue par vote majoritaire de tous les arbres. Le ratio entre le nombre d'arbres ayant rendu la classe prédite finale et le nombre total des arbres construits peut servir comme un score de confiance de la prédiction. Le modèle de classification des lymphomes est illustré dans la Figure 4.10. Il permet de classer n'importe quel lymphome en trois étapes :

- la première étape vise à distinguer entre lymphome B et T.
- La deuxième étape dépend du premier résultat :
  - si le premier test classe l'échantillon en tant que lymphome T, le deuxième test sera le dernier et classera l'échantillon en un des huit sous-types de lymphomes T (AITL, ALCL ALK+, ALCL ALK-, ATLL, CD30TH2, NKTCL, HSTL et PTCL NOS).
  - Sinon, si le premier test classe l'échantillon en tant que lymphome B, le deuxième test classera l'échantillon en lymphome à grandes ou à petites cellules.
- Un troisième test n'est nécessaire que si l'échantillon est prédit comme lymphome B : il sera classé en ABC, GCB ou PMBL si le deuxième test lui accorde la classe de lymphome à grandes cellules et d'autre part, il sera classé en MCL, FL, SSL ou MZL si le deuxième test lui donne la classe de lymphome à petites cellules.

Gènes ABC	Gènes GCB	Gènes PMBL	Expresseurs doubles	Autres marqueurs
TACI	*CD10*	*IL4I1*	*BCL2* #1 (exon1-2)	EBER1
FOXP1	*LMO2*	*CD23*	*BCL2* #2 (exon2-3)	HTLV1
LIMD1	*ASB13*	*CD30*	*MYC* #1 (exon1-2)	*KI67*
IRF4	*NEK6*	*MAL*	*MYC* #2 (exon2-3)	*CD68*
PIM2	*MYBL1*	*CD95*		*CD163*
CCDC50	*MAML3*	*CD71*	**Cellules T**	*CCND1*
CREB3L2	*ITPKB*	*FGFR1*	*TCRα*	*CCND2*
CYB5R2	*SERPINA9*	*JAK2*	*TCR β*	*ZAP70*
SH3BP5	*S1PR2*	*TRAF1*	*TCRγ*	*ANXA1*
RAB7L1	*BCL6#1 (exon1-2)*	*STAT6*	*TCR δ*	*CRBN*
	BCL6#2 (exon3-4)	*PD-L1*	*CD3*	*STAT6*
		PD-L2	*CD5*	*APRIL*
			CD4	*BAFF*
Cellules B	**Gènes Ig**	**Fusion de gènes**	*CD8*	*BCMA*
CD19	Iα-Cα	*BCL6-C α*	*TBET*	*CCR4*
MS4A1 (CD20)	Iα-Cε	*BCL6-C ε*	*INF γ*	*CCR7*
CD22	Iα-Cγ	*BCL6-C γ*	*GRB*	*CD56*
CD27	Iα-Cμ	*BCL6-C μ*	*PRF*	*CD70*
CD38	Iε-Cα	*Iγ-BCL6*	*CD45RO*	*DUSP22*
CD138	Iε-Cε	*Iε-BCL6*	*CXCR5*	*MEF2B*
CD86	Iε-Cγ	*Iα-BCL6*	*CXCL13*	*PRDM1*
CD80	Iε-Cμ	*Iμ-BCL6*	*GATA3*	*XBP1*
CTLA4	Iγ-Cα	*JH-BCL6*	*CD28*	*CARD11*
B2M	Iγ-Cε		*ICOS*	*TCL1A*
	Iγ-Cγ		*FOXP3*	*BANK*
CSR/SHM	Iγ-Cμ	**Mutations**	*PD1*	
AID#1 (exon2-3)	IGHD	*XPO1* E571K	*LAG3*	
AID#2 (exon4-5)	IGHM	*MYD88* L265P	*ALK*	
CD40	Iμ-Cα	*BRAF* V600E		
CD40L#1 (exon2-3)	Iμ-Cε	*IDH2* R172K		
CD40L#2 (exon4-5)	Iμ-Cγ	*RHOA* G17V		
	Iμ-Cμ	*MYD88 (exon3-4)*		
	JH-Cα	*XPOWT*		
	JH-Cε			
	JH-Cγ			
	JH-Cμ			

FIGURE 4.8 – Les gènes utilisés par le *Random Forest* pour la classification des lymphomes B et T. Figure adaptée de [106].

Finalement, une analyse de chimères est effectuée pour s'assurer que l'expérience s'est bien déroulée. Une chimère est le produit d'une combinaison de l'amorce gauche d'un marqueur A avec l'amorce droite d'un marqueur B. Les chimères sont considérées comme un contaminant, car une chimère peut être interprétée comme une nouvelle séquence alors qu'il s'agit en fait d'un artefact. L'analyse des chimères vise à retrouver toutes les chimères dans les *reads* et calculer leur pourcentage. Un pourcentage inférieur à 0,1% reflète normalement une expérience bien réussie.

4.4.2.5 Détection des transcrits de fusion

La seconde application de la RT-MLPA couplée à un séquenceur NGS est son utilisation pour la détection des transcrits de fusion dans différents types de cancers. En effet, les transcrits de fusion, ou ARN chimériques, résultent de la juxtaposition de deux gènes, précédemment localisés séparément l'un de l'autre, en raison d'événements chromosomiques ou non chromosomiques. Ils peuvent être la conséquence de réarrangements chromosomiques structurels ou être le produit d'un épissage alternatif ou de relectures transcriptionnelles [107]. Les transcrits de fusion peuvent conduire à l'activation de proto-oncogènes ou à l'inactivation des gènes suppresseurs de tumeurs et sont considérés comme l'un des principaux mécanismes responsables de la carcinogenèse. Les transcrits de fusion sont considérés comme de puissants biomarqueurs diagnostiques présentant un grand intérêt clinique et sont donc de plus en plus explorés comme potentielles cibles

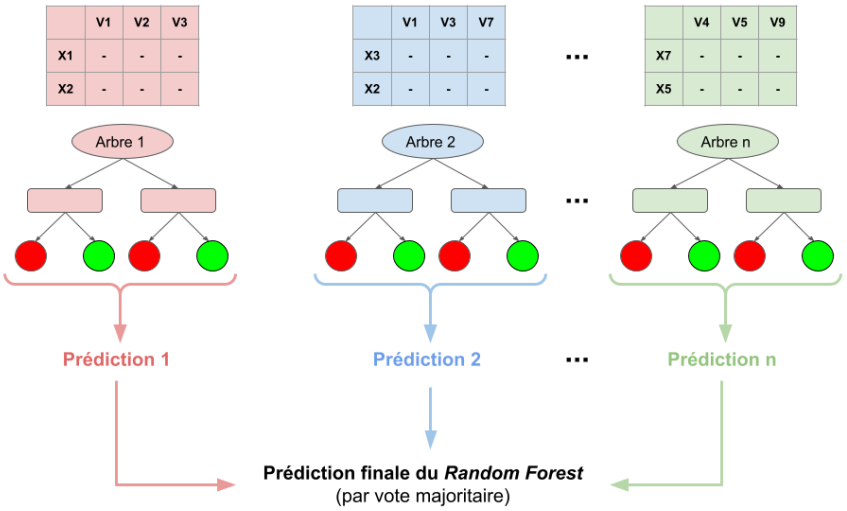


FIGURE 4.9 – Le principe d'un classifieur de type *Random Forest*.

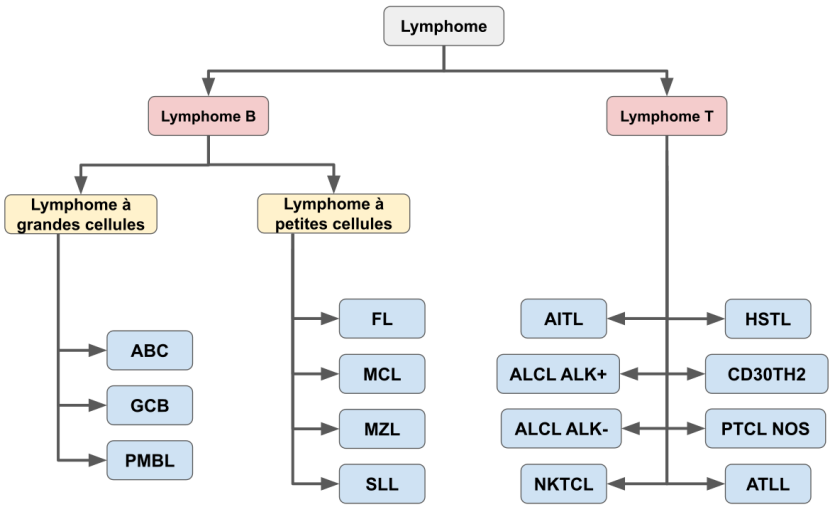


FIGURE 4.10 – Le modèle statistique de type *Random Forest* pour la classification des lymphomes B et T.

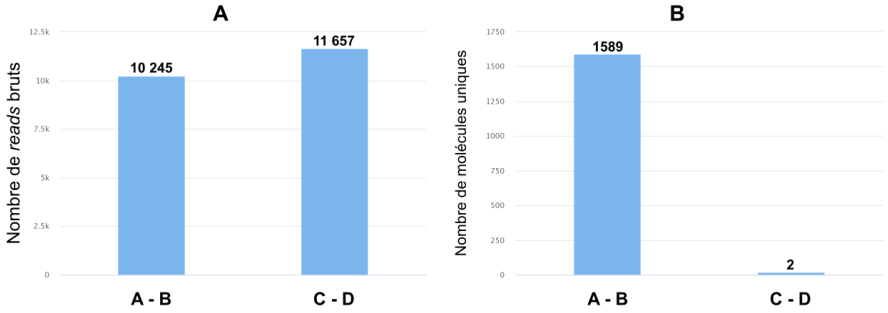


FIGURE 4.11 – Exemple d'analyse de détection de deux transcrits de fusion A-B et C-D. (A) et (B) représentent respectivement le nombres de *reads* bruts et le nombre de molécules uniques (UMI) pour chaque transcrit.

thérapeutiques, notamment dans les sarcomes, les gangliomes et le carcinome pulmonaire. Dans ce contexte, les marqueurs recherchés sont, en effet, des combinaisons entre une amorce gauche d'un marqueur et une amorce droite d'un autre marqueur. Ainsi, le fichier des marqueurs contient généralement toutes les combinaisons possibles entre chaque amorce de chaque marqueur. Dans le cas des sarcomes par exemple, le fichier contient plus de 20 000 combinaisons possibles. Dans ce type d'analyse, le but principal est de distinguer entre les vrais transcrits de fusion présents initialement dans l'échantillon et les faux positifs générés par l'expérience elle-même. C'est ici où est démontrée la grande utilité des UMI puisque justement, en supprimant les doublons de PCR, ils permettent de calculer avec grande précision le nombre de molécules initiales au lieu de leur nombre brut. Par exemple, dans la Figure 4.11, en comparant le nombre de *reads* bruts obtenus pour les deux transcrits de fusion A-B et C-D, on remarque que les deux ont des comptes comparables (10 245 pour A-B contre 11 657 pour C-D). En regardant ces comptages seulement, on aurait tendance à dire que le patient est positif pour les deux transcrits de fusion. Cependant, en regardant le nombre de molécules uniques (ou nombre d'UMI) pour chaque transcrit, on constate que l'échantillon initial contenait 1589 molécules distinctes pour A-B alors que seulement 2 pour C-D. De cette manière, nous pourrons conclure que l'échantillon est positif au transcrit A-B seulement et que le transcrit C-D est, en effet, un artefact. Cette analyse doit être faite pour toutes le combinaisons possibles entre les marqueurs.

4.5 Développement de RT-MiS

Ainsi, nous avons développé l'outil RT-MiS capable de gérer l'analyse bioinformatique pour les deux types d'applications : la mesure de l'expression génique et la détection des transcrits de fusion. Dans les deux cas, les premières étapes sont identiques alors que l'étape finale est spécifique pour chaque type d'application. L'outil est implémenté dans une interface d'analyse *web* permettant de gérer automatiquement toute l'analyse bioinformatique de la récupération du FASTQ produit par le séquenceur Illumina jusqu'à la production des résultats sous forme de graphique facilement interprétable. L'outil et l'interface on été déposés à l'Agence pour la Protection des Programmes le 14/12/2018 sous le nom de RT-MiS et sous le numéro : IDDNFR001510018000SC201800030000. Il est régulièrement mis à jour pour l'adapter aux nouveaux domaines d'application mis en place au CHB (dernière mise à jour en juin 2021).

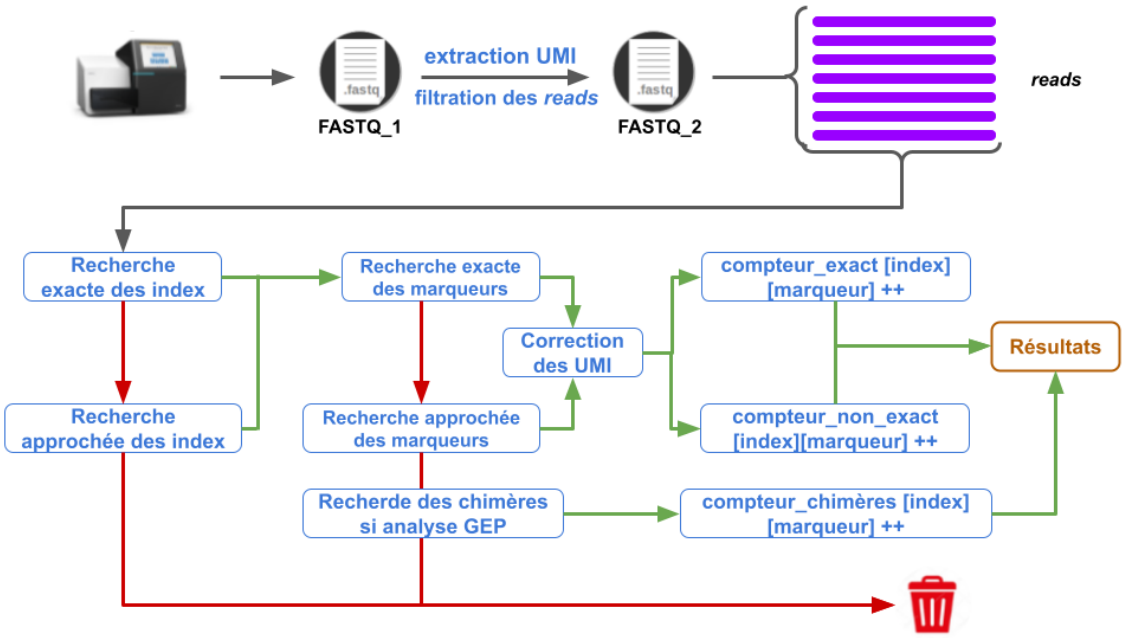


FIGURE 4.12 – Le *workflow* de l'outil RT-MiS. Les flèches en rouge représentent une opération ayant échouée tandis que les flèches en vert représentent une opération ayant réussie.

4.5.1 L'outil RT-MiS

Le *workflow* de RT-MiS est illustré dans la Figure 4.12 et consiste en six étapes principales : le traitement du fichier FASTQ pour l'extraction et la correction des UMI, le traitement du fichier d'index, le traitement du fichier des marqueurs, la recherche exacte et approchée des marqueurs dans les *reads* et, finalement, le calcul et la production des résultats. L'enchaînement et le déroulement de ces étapes permettent d'assurer une analyse très efficace que ce soit en terme de temps d'exécution ou de consommation mémoire.

4.5.1.1 Le traitement du fichier FASTQ

Le traitement du fichier FASTQ se fait en trois étapes : l'extraction des UMI du début des *reads*, la filtration des *reads* et la création d'une structure de données permettant de stocker l'ensemble des *reads* et les informations associées à chacun d'eux. La structure de données utilisée doit permettre d'accéder à un *read* rapidement à partir d'une clé. Le fichier FASTQ, décrit dans le Chapitre 2, contient la totalité des *reads* dans un format décrivant chaque *read* sur 4 lignes. La configuration de chaque *read* est connue et a été décrite dans la Section 4.4.1 et dans la Figure 4.5. L'extraction des UMI est gérée par un module écrit en C/C++ dont la fonction est de parcourir l'ensemble des *reads* du fichier, extraire les n premières bases de chaque séquence et l'ajouter à la fin de l'identifiant du *read*, précédée par le caractère "_". Ce module, présenté dans la Figure 4.13, réalise l'extraction exactement comme l'outil UMI-tools, décrit en détail dans le Chapitre 3 ; cependant, nous avons préféré de réécrire le module en C++ afin d'être plus rapide (UMI-tools est écrit en Python) et surtout pour que RT-MiS soit autonome, ne dépendant d'aucun module extérieur. Le paramètre n dépendra de chaque expérience, il doit être précisé par l'utilisateur et représente la taille de l'UMI.

Ensuite, la deuxième étape consiste à filtrer les *reads* selon leur score de qualité. En effet, nous avons développé un deuxième module en C++ pour transformer le FASTQ initial (4 lignes par *read*) en un FASTQ à 2 lignes par *read*. Ceci permet de réduire la taille

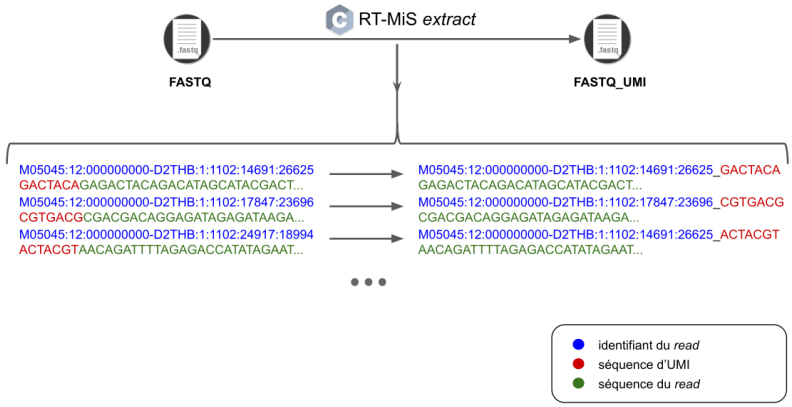


FIGURE 4.13 – L'extraction des UMI par RT-MiS.

du fichier et d'analyser la séquence de qualité plus rapidement. Un seuil de score qualité est choisi par l'utilisateur et est fixé à 10 par défaut. Pour chaque *read*, le module convertit le code ASCII représentant la qualité de chaque base en score de qualité (entre 1 et 41) et le compare au seuil : si le score est supérieur au seuil, la lettre est gardée en majuscule mais si le score lui est inférieur, la lettre sera réécrite en minuscule dans le nouveau FASTQ. Un exemple est présenté dans la Figure 4.14. Les *reads* ayant plus de lettres en minuscule qu'en majuscule seront filtrés.

Une fois la filtration terminée, RT-MiS analyse la séquence de chaque *read* selon le schéma de la Figure 4.15. Dans la séquence du *read*, la séquence du marqueur est variable (selon chaque marqueur) et elle est séparée de la séquence de l'index par l'espaceur Ilu1. Ilu1 a une taille constante de a et la taille de la séquence de l'index est aussi constante, égale à b. Ainsi, en localisant la séquence de Ilu1, nous extrayons toute la séquence qui la précède et la stockons en tant que séquence du marqueur. De l'autre côté, la séquence de taille b suivant Ilu1 représente la séquence de l'index. Une recherche exacte et approchée (tolérance d'une distance d'édition de 1) sont effectuées sur la séquence du *read* vu que Ilu1 peut contenir des erreurs de séquençage. Pour stocker les séquences obtenues et les associer à un *read*, une table de hachage est créée. La table de hachage est une structure de données ne comportant pas d'ordre (contrairement aux tableaux/listes) et qui permet une association clé–valeur. Son but - et avantage - principal est de permettre de retrouver une clé donnée très rapidement, en la cherchant à un emplacement de la table correspondant au résultat d'une fonction de hachage calculée en temps constant. Cela constitue un gain de temps très important pour les grosses tables surtout lors d'une recherche. Ainsi, l'utilisation de cette structure permet un accès rapide aux informations de chaque *read* : elle contient comme clé l'identifiant du *read* et une liste composée de l'UMI associé, la séquence de l'index et la séquence du marqueur en tant que valeur. La structure de cette table est présentée dans la Figure 4.16.

4.5.1.2 Le traitement du fichier d'index

Les deux colonnes du fichier d'index sont analysées ligne par ligne et leur contenu est stocké dans une table de hachage ayant comme clé le nom de l'index et sa séquence en tant que valeur.

4.5.1.3 Le traitement du fichier des marqueurs

Le traitement du fichier d'index est relativement simple et ne produit qu'une seule table de hachage en conséquence. D'autre part, le traitement du fichier des marqueurs

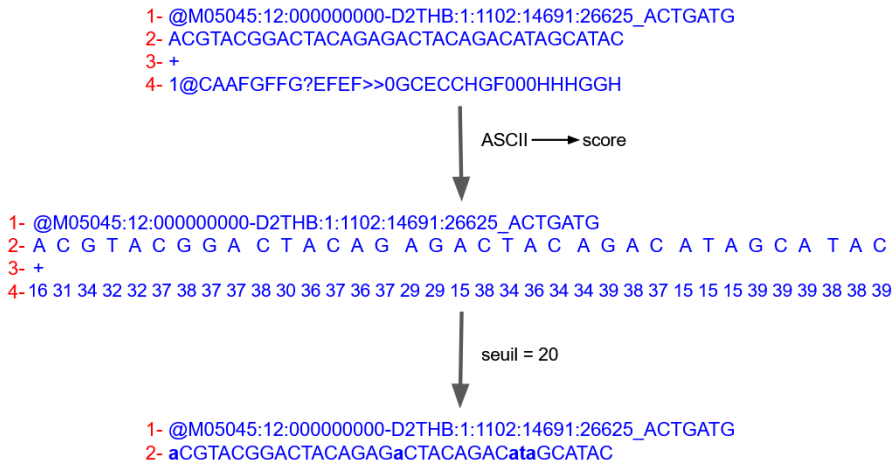


FIGURE 4.14 – La méthode de filtration utilisée par l'outil RT-MiS pour traiter les *reads* dans le FASTQ. Les lettres modifiées sont représentées en gras.

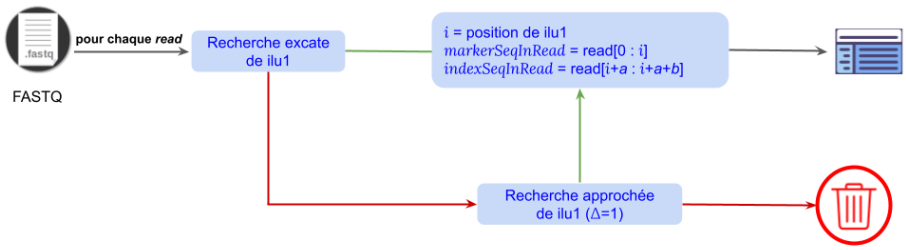


FIGURE 4.15 – L'extraction de la séquence de l'index et du marqueur de la séquence entière du *read*. Les flèches en rouge représentent une opération ayant échouée tandis que les flèches en vert représentent une opération ayant réussie.

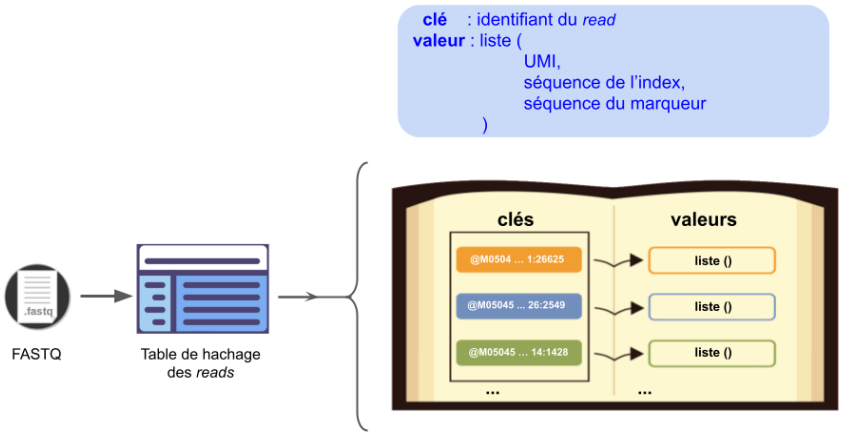


FIGURE 4.16 – La table de hachage contenant les *reads* résultant du traitement du fichier FASTQ.

nécessite plus d'étapes afin d'assurer une recherche approchée efficace. Dix tables de marqueurs sont produites dans cette étape :

1. clé : séquence du marqueur et valeur : nom du marqueur ;
2. clé : nom du marqueur et valeur : séquence du marqueur ;
3. clé : longueur du marqueur et valeur : nom du marqueur ;
4. clé : les bases 1 à 3 de la séquence du marqueur et valeur : nom du marqueur ;
5. clé : les bases 4 à 6 de la séquence du marqueur et valeur : nom du marqueur ;
6. clé : les bases 7 à 9 de la séquence du marqueur et valeur : nom du marqueur ;
7. clé : les trois bases au milieu de la séquence du marqueur et valeur : nom du marqueur ;
8. clé : les bases -9 à -7 de la séquence du marqueur et valeur : nom du marqueur ;
9. clé : les bases -6 à -4 de la séquence du marqueur et valeur : nom du marqueur ;
10. clé : les bases -3 à -1 de la séquence du marqueur et valeur : nom du marqueur.

La première structure servira dans la recherche exacte alors que les neuf autres tables seront essentielles pour réaliser la recherche approchée.

4.5.1.4 La recherche des index

La recherche des index est une étape essentielle permettant d'attribuer chaque séquence au patient correspondant. Vu que la séquence de l'index peut contenir des erreurs de séquençage, une recherche exacte et approchée sont obligatoires et se font en deux temps. D'abord, la recherche exacte est effectuée en essayant de retrouver la séquence de l'index dans les clés de la table de hachage des index de la Section 4.5.1.3. Deux cas existent : si la recherche exacte est réussie, le nom de l'index est récupéré et la recherche des marqueurs commencera ; sinon, le logiciel tentera une recherche approchée. Pour cela, il va comparer la séquence en question avec toutes les séquences présentes dans la structure contenant les index tout en calculant la distance avec chacun d'eux. La distance maximale acceptée est un paramètre choisi par l'utilisateur et qui est, par défaut, définie à 1. Un autre paramètre sert à choisir entre la distance de Hamming (valeur par défaut) ou la distance d'édition. Si la recherche approchée aboutit à retrouver une séquence semblable à celle de l'index analysé, le nom de l'index sera récupéré et la recherche des marqueurs est déclenchée. Dans ce cas, un booléen est créé pour indiquer que la recherche des marqueurs est lancée suite à une recherche approchée aboutie. Cependant, si la recherche approchée échoue, le *read* est alors ignoré. L'algorithme de la recherche des index est illustré dans la Figure 4.17.

4.5.1.5 La recherche des marqueurs

Une fois la séquence attribuée à un nom d'index, la recherche du marqueur est déclenchée. Cette étape est nécessaire puisqu'elle permet de compter le nombre de séquences exprimant un marqueur déterminé. Comme la séquence de l'index, la séquence du marqueur est susceptible de contenir des erreurs de séquençage, voire plus puisque sa taille est plus grande. De ce fait, une recherche exacte et approchée sont également obligatoires dans ce cas. De la même façon que pour les index, la recherche exacte est effectuée en essayant de retrouver la séquence du marqueur dans les clés de la première table de hachage des marqueurs de la Section 4.5.1.4. D'autre part, la recherche approchée utilisée pour les index ne peut pas être utilisée ici pour deux raisons principales : nous avons

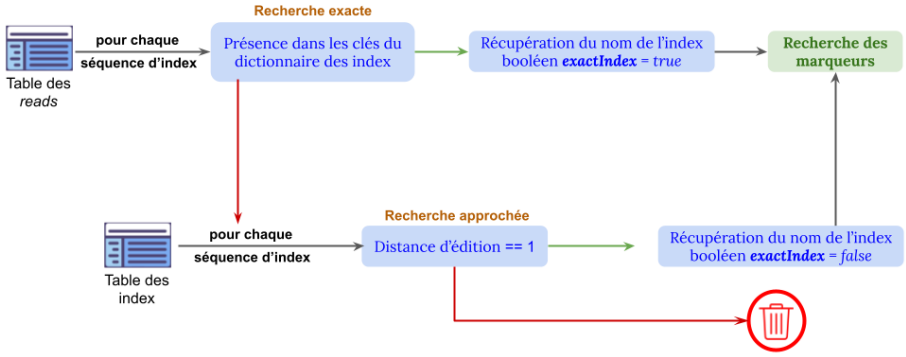


FIGURE 4.17 – La recherche exacte et approchée des index dans les *reads*. Les flèches en rouge représentent une opération ayant échouée tandis que les flèches en vert représentent une opération ayant réussie.

beaucoup plus de marqueurs que d'index (entre 137 et 20 000 marqueurs selon l'application contre une quarantaine d'index seulement) et que la taille moyenne des marqueurs est cinq fois supérieure à celle des index (59 contre 12). De ce fait, la recherche approchée serait très lente et le temps augmenterait exponentiellement avec le nombre des *reads* et le nombre de marqueurs à rechercher. La méthode que nous avons développée consiste à utiliser les huit structures construites en 4.5.1.4 (3 à 10) à partir du fichier des marqueurs pour établir moins de calcul de distances. À partir de chaque table, nous récupérons le nom des marqueurs ayant les mêmes caractéristiques que la séquence à identifier. Par exemple, pour une séquence commençant par ATG, de longueur 55 pb et se terminant par GCA, nous récupérons de la table 3 tous les noms de marqueurs ayant une longueur de 55 pb, de la table 4 tous les noms de marqueurs commençant par ATG et de la dernière table, tous les noms de marqueurs se terminant par GCA. Ceci est fait pour toutes les structures de 3 à 10 et les noms obtenus sont ajoutés dans une liste de candidats. Ensuite, nous trions la liste par ordre décroissant de fréquence d'observation dans la liste pour choisir enfin celui qui sort en tête. Après avoir récupéré le nom du marqueur, nous utilisons la table de hachage 2 pour obtenir sa séquence et donc calculer la distance entre la séquence du marqueur retenu et celle contenue dans la séquence du *read*. Si cette distance est inférieure au seuil choisi par l'utilisateur (même seuil que pour la recherche des index), le marqueur est sélectionné. Sinon, selon le type d'application, le sort de la séquence du *read* est différent. S'il s'agit d'une analyse d'expression génique, le *read* est ajouté à une nouvelle table de hachage contenant les chimères potentielles pour être analysé plus tard. Dans le cas d'une analyse de transcrits de fusion, le *read* est ignoré. L'algorithme de la recherche des marqueurs est illustré dans la Figure 4.18. Si le marqueur est trouvé suite à une recherche approchée, un compteur spécifique au marqueur et à l'index sera incrémenté dans une table de hachage spécifique aux comptages dûs à une recherche approchée. De même, si la recherche exacte du marqueur est réussie mais que le booléen défini lors de la recherche des index a la valeur *false*, le compteur de la table des recherches approchées est incrémenté. Les compteurs exacts se trouvent dans une autre table et ne sont incrémentés que lorsque l'identification de l'index et du marqueur sont, toutes les deux, le résultat d'une recherche exacte. Les deux structures de données sont illustrées dans la Figure 4.19.

4.5.1.6 La correction des UMI

Les tables de hachage produites par la recherche des index et des marqueurs contiennent les comptages de chaque UMI pour chaque marqueur et pour chaque index. Les

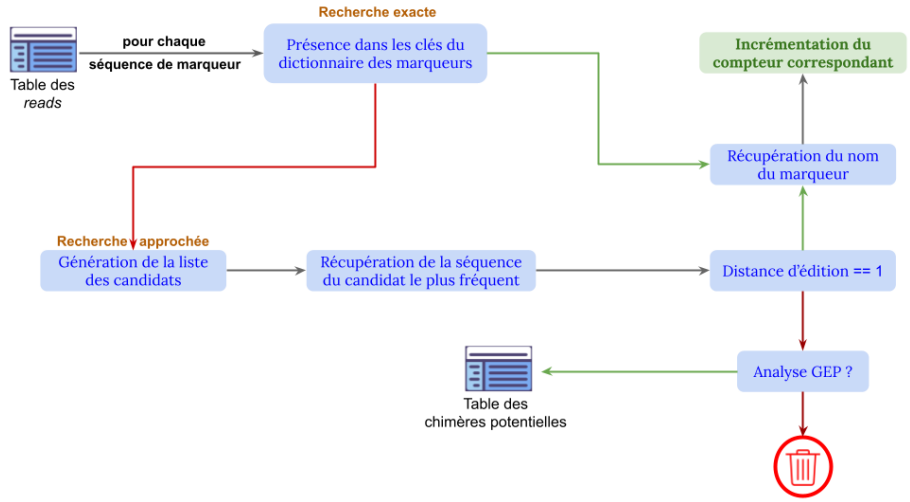


FIGURE 4.18 – La recherche exacte et approchée des marqueurs dans les *reads*. Les flèches en rouge représentent une opération ayant échouée tandis que les flèches en vert représentent une opération ayant réussie.

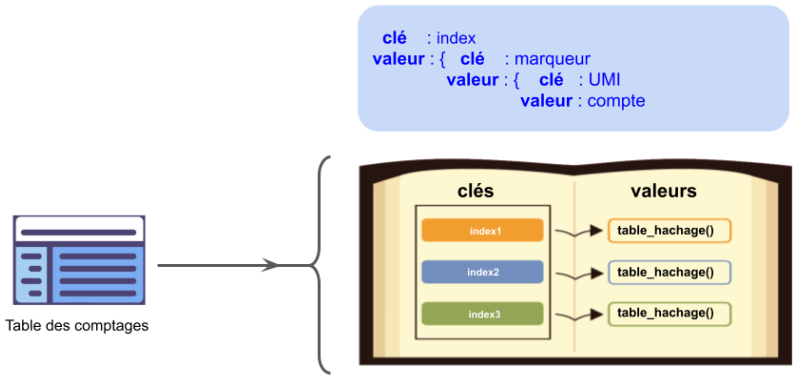


FIGURE 4.19 – La structure des tables de hachage des comptages résultant des recherches des index et des marqueurs.

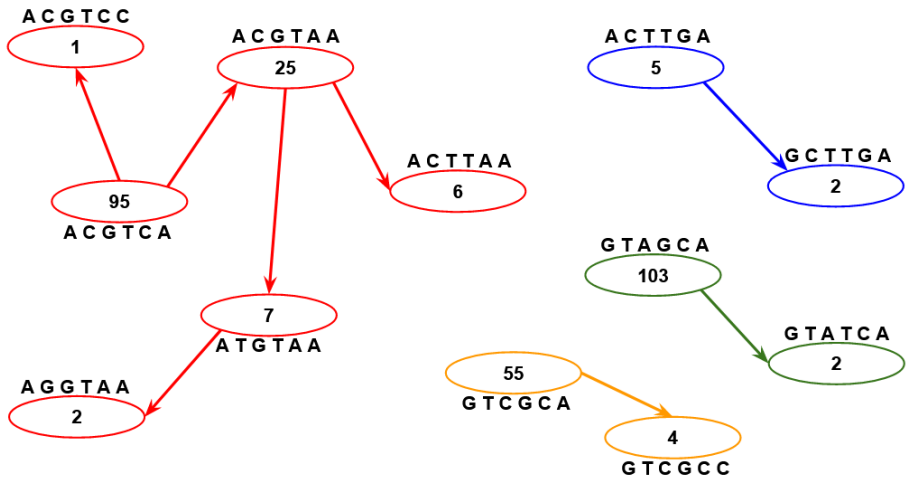


FIGURE 4.20 – La méthode *directional* utilisée par l'outil RT-MiS pour la correction des UMI.

deux tables ont des structures identiques, la seule différence étant la valeur des compteurs dans chacune d'elles, l'une provenant d'une recherche exacte et l'autre de la recherche approchée. Pour cela, nous allons décrire la correction des UMI sur l'une des structures seulement, sachant que la méthode est exactement ré-appliquée sur l'autre. Comme décrite dans la Figure 4.18, la table est en effet composé de trois tables de hachage imbriquées. La correction des UMI est réalisée pour chaque index et pour chaque marqueur selon la méthode *directional* développée par Smith *et al.* dans l'outil UMI-tools. Pour un index et un marqueur déterminés, un graphe est créé entre tous les UMI. Ce graphe est orienté et il est formé par des nœuds, contenant la séquence de l'UMI et sa fréquence, et des arêtes pour relier les nœuds entre eux. Deux nœuds sont liés si et seulement si deux conditions sont vérifiées :

1. la fréquence d'un des nœuds $\geq 2 \times$ la fréquence de l'autre nœud-1.
2. La distance d'édition entre les séquences des UMI est exactement égale à 1.

Ainsi, en appliquant ces deux règles pour relier les UMI, des *clusters* bien distincts seront formés. Pour résoudre le graphe, il suffit de considérer chaque *cluster* comme un seul UMI. La séquence de l'UMI final sera celle du nœud avec la plus grande fréquence et la fréquence finale sera la somme de toutes les fréquences du *cluster*. Un exemple est donné dans la Figure 4.20. Dans cet exemple, grâce à la correction des UMI, on est passé de 12 UMI à 4 UMI distincts seulement. Les UMI gardés sont : ACGTCA (136), ACTTGA (7), GTAGCA (105) et GTCGCA (59). La correction des UMI est appliquée pour les deux types d'analyses.

4.5.1.7 La production des résultats

Une fois les opérations de correction et de comptage dans les deux tables de hachage terminées, le logiciel produira les résultats sous forme de fichiers CSV. Le premier fichier obtenu contient des statistiques par rapport à la recherche exacte et approchée effectuées. Des mesures telles que le nombre total de séquences analysées, le pourcentage des séquences sans aucun marqueur trouvé ou encore le pourcentage des *reads* filtrés y sont présentes. Deux autres fichiers sont produits contenant les matrices d'expression génique pour chaque marqueur et pour chaque index. Un fichier contient les comptages provenant de la recherche exacte tandis que l'autre contient ceux provenant de la recherche

	LMO2	JH.Cgamma	S1PR2	PD1	SH3BP5	CRBN	JAK2	ASB13	CRBN	CD23
D701	403	0	562	92	2	13555	569	1006	13126	110
D702	34	0	495	227	148	601	238	0	724	144
D703	112	0	1170	679	231	1799	373	356	5276	18
D705	266	0	224	135	11	4257	406	183	3224	64
D707	39	0	9	59	36	315	68	12	375	35
D708	95	0	131	58	29	295	70	5	243	82
D709	38	0	85	0	450	407	429	39	962	66
D711	1673	0	55	61	24	9708	355	58	6492	82

FIGURE 4.21 – Un exemple de la matrice d'expression génique produite par l'outil RT-MiS.

	marqueur	ASB13	CRBN	CD23	CXCL13	CCR7XPOWT	JH.Cgamma	JAK2	MS4A1	CD95
2	**CD38**	1	1	5	2	34	1	3	1	2
3	**SH3BP5**	2	2	2	3	1	2	2	9	3
4	**CCND1**	2	1	2	1	3	1	1	8	1
5	**CCDC50**	5	3	2	3	1	2	1	3	1
6	**JH-Cmu**	7	5	1	2	4	2	1	4	5
7	**MEF2B**	3	7	6	1	7	2	2	3	4
8	**LAG3**	5	1	2	2	5	1	6	3	6
9	**KI67**	7	8	1	3	2	5	11	2	6
10	**PD1**	23	1	1	3	8	4	9	1	3
11	**LMO2**	9	1	4	4	5	3	4	1	2
12	**PIM2**	2	2	5	1	11	2	8	1	2
13	**SERPINA9**	8	3	1	5	3	2	1	1	1
14	**S1PR2**	1	3	1	2	3	5	1	1	1

FIGURE 4.22 – Un exemple de la matrice de chimères produite par l'outil RT-MiS.

exacte et approchée. Un exemple de la matrice d'expression génique produite par RT-MiS est illustré dans la Figure 4.21. Ensuite, un fichier par index est produit contenant tous les *reads* dans lesquels la séquence de l'index a pu être trouvée. Ce fichier contient cinq colonnes : identifiant du *read*, séquence du *read*, UMI, la séquence théorique du marqueur et le nom du marqueur (si un marqueur a pu être identifié, sinon, NA). Finalement, un fichier (appelé *trash.csv*) est produit contenant toutes les séquences dans lesquelles la séquence de Ilu1 ou d'un index n'ont pas pu être retrouvées. Dans le cas d'une analyse de type GEP, un fichier supplémentaire est généré contenant la matrice de comptage de toutes les chimères retrouvées. Un exemple de cette matrice est présenté dans la Figure 4.22. Tous ces fichiers sont stockés dans un répertoire unique et spécifique à l'analyse lancée et seront utilisés par l'interface RT-MiS dédiée pour générer les résultats sous formes de graphiques interactifs.

4.5.1.8 Implémentation

RT-MiS est composé de trois outils principaux : un outil pour l'extraction des UMI, un deuxième pour la conversion des FASTQ (format 4 lignes en format 2 lignes) et le troisième pour effectuer la recherche exacte et approchée des index et des marqueurs. Les outils d'extraction et de conversion ont été développés en C++ puisqu'ils ne nécessitent pas de structures compliquées et que le but est de parcourir le fichier le plus rapidement possible. D'autre part, nous avons trouvé que les structures en tables de hachage nécessaires pour associer des clés à des valeurs (pouvant être des tables de hachage elles-mêmes) est facilement implémentable en Python grâce aux dictionnaires, des structures de base de ce langage de programmation permettant un accès très rapide aux valeurs associées aux clés stockées. Ainsi, l'outil de recherche a été développé en Python : il est composé de plusieurs fonctions, chacune responsable d'une opération et dont l'appel est géré par un script principal.

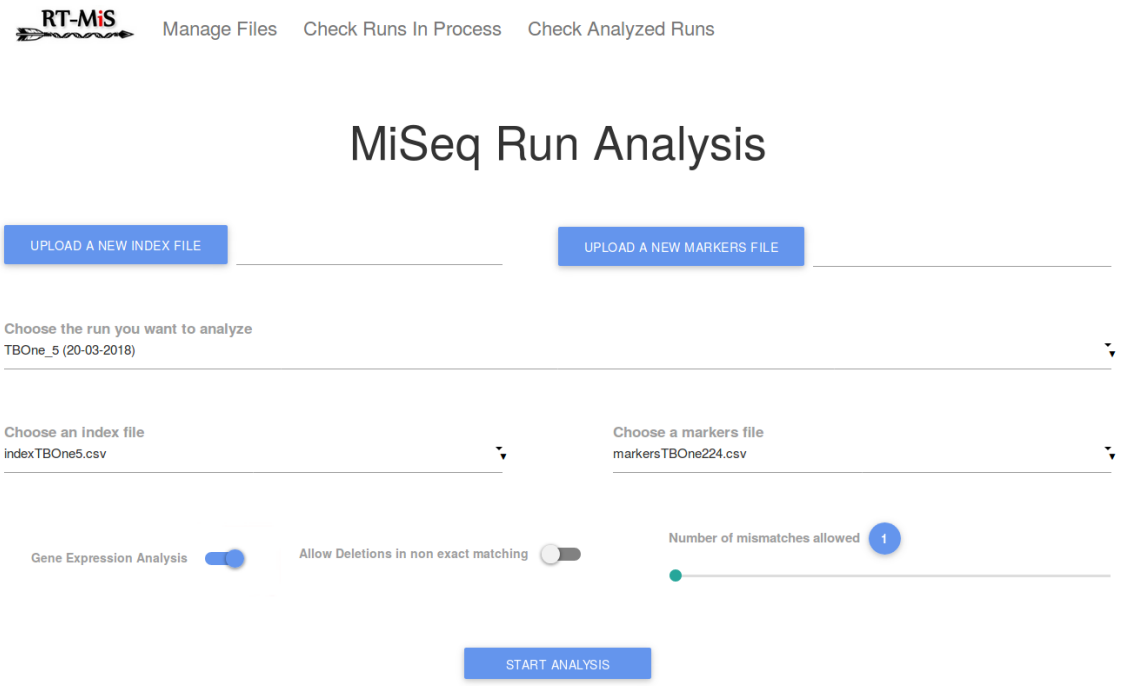


FIGURE 4.23 – La page d'accueil de l'interface RT-MiS.

4.5.2 L'interface d'analyse dédiée RT-MiS

RT-MiS est un logiciel écrit en Python et exécutable facilement à partir d'une interface en ligne de commande. Cependant, il a été développé pour permettre aux chercheurs du CHB, principalement des biologistes, de lancer leurs analyses avec simplicité. Ainsi, le développement d'une interface *web* implémentant l'algorithme de recherche et d'analyse nous a paru indispensable. L'interface RT-MiS est la combinaison de plusieurs scripts PHP, CSS, JavaScript, Ajax et R permettant une gestion complète de toutes les étapes des deux types d'analyses à n'importe quel instant. Elle est actuellement utilisée au CHB et constitue un élément essentiel du travail de recherche réalisé par l'équipe.

4.5.2.1 La gestion des fichiers d'index

La page d'accueil présentée dans la Figure 4.23 contient tous les éléments nécessaires pour lancer une analyse. Tout d'abord, il faut choisir un fichier d'index à partir d'un menu déroulant. Pour ajouter un nouveau fichier d'index, il existe un bouton *UPLOAD NEW INDEX FILE* permettant de charger un fichier CSV contenant le nom et la séquence de chaque index. Une fois la vérification du fichier terminée, il sera ajouté à la liste des fichiers dans le menu déroulant. L'interface offre aussi la possibilité de créer un nouveau fichier d'index à partir des *barcodes*. Pour cela, il faut choisir les *barcodes* à inclure dans le nouveau fichier depuis un menu déroulant et lui donner un nom. Si besoin, l'utilisateur peut aussi ajouter de nouveaux *barcodes*. Le nouveau fichier d'index apparaîtra dans le menu déroulant de la page d'accueil. Ce système permet une configuration facilitée pour les utilisateurs qui doivent parfois gérer plusieurs dizaines de *barcodes* par analyse et un contrôle de la validité des *barcodes* et de leur séquence respective.

4.5.2.2 La gestion des fichiers des marqueurs

Indispensable aussi pour lancer une analyse, le choix d'un fichier des marqueurs se fait à partir d'un menu déroulant comme pour les fichiers d'index. Pour ajouter un nouveau fichier des marqueurs, il existe un bouton *UPLOAD NEW MARKERS FILE* dédié comme le montre la Figure 4.23. Ceci permet de charger un fichier CSV contenant le nom, la séquence gauche et la séquence droite de chaque marqueur. Une fois la vérification du fichier terminée, ce dernier pourra être sélectionné à partir du menu déroulant de la page d'accueil. Par contre, l'interface n'offre pas la possibilité de créer un nouveau fichier des marqueurs. Ceci est dû au fait que les fichiers de marqueurs utilisés sont souvent les mêmes. Par conséquent, l'utilisateur n'aura que rarement à en créer un nouveau. Ici encore, l'objectif est d'éviter les erreurs de manipulation des utilisateurs qui n'auront à paramétrer qu'une seule fois leur fichier des marqueurs pour une librairie donnée.

4.5.2.3 La gestion des fichiers FASTQ

De la même façon, le choix des fichiers à analyser est fait à partir d'un menu déroulant sur la page d'accueil. Cette liste représente l'ensemble de tous les fichiers éligibles à une analyse avec RT-MiS. Au CHB, un répertoire sur le NAS (*Network Attached Storage*) contient toutes les données brutes produites par le séquenceur Illumina MiSeq. Dans la mesure où ce support de sauvegarde intègre toutes les données omiques du CHB (DNA-Seq, transcriptomique, ...), une étape de filtration est réalisée pour ne permettre de lancer l'outil que sur des fichiers compatibles et pré-traités. Donc, une étape manuelle est requise pour l'ajout d'un fichier dans la liste des fichiers de la page d'accueil. Tout d'abord, il faut choisir l'onglet *Manage Files* et puis cliquer sur *Add a new run*. Sur cette nouvelle page, l'utilisateur pourra lancer les modules d'extraction des UMI et de filtration des *reads* de l'outil RT-MiS pour préparer le fichier FASTQ à l'analyse. Une fois les deux opérations terminées, le fichier apparaîtra dans le menu déroulant de la page d'accueil et l'utilisateur pourra le sélectionner pour lancer une analyse.

4.5.2.4 La gestion des analyses

Pour lancer une analyse, l'utilisateur doit obligatoirement choisir un fichier d'index, un fichier des marqueurs et le fichier qu'il souhaite analyser, chacun du menu déroulant correspondant. Une fois ces fichiers choisis, trois arguments restent à préciser (Figure 4.23). Le premier correspond au type de l'analyse que l'on souhaite réaliser. Si le bouton est sélectionné, une analyse du type mesure d'expression génique est réalisée. Sinon, une recherche de transcrits de fusion est réalisée sur le fichier choisi. De plus, l'interface offre la possibilité de personnaliser la recherche approchée. L'option s'appelle *Allow deletions in non exact matching* et si elle est sélectionnée, l'algorithme utilisera la distance d'édition au lieu de la distance de Hamming pour la recherche approchée. Par défaut, cette option n'est pas choisie et l'algorithme ne considère que les substitutions, et donc la distance de Hamming, lors de la recherche approchée. Le dernier argument concerne aussi la recherche approchée. Un autre niveau de personnalisation est offert par l'interface. Elle permet à l'utilisateur de choisir le nombre maximal d'erreurs tolérées lors de la recherche approchée. Par défaut, ce seuil est fixé à 1. Après avoir choisi le fichier à analyser, le fichier des marqueurs, le fichier d'index et avoir réglé les paramètres de recherche, l'interface lancera l'analyse en appelant l'outil RT-MiS pour effectuer la recherche. Une barre de progression apparaît permettant à l'utilisateur de suivre en temps réel le déroulement de l'analyse comme le montre la Figure 4.24. En effet, l'analyse est lancée en arrière plan ce qui permet à l'utilisateur de fermer la fenêtre pour y accéder dans un temps ultérieur

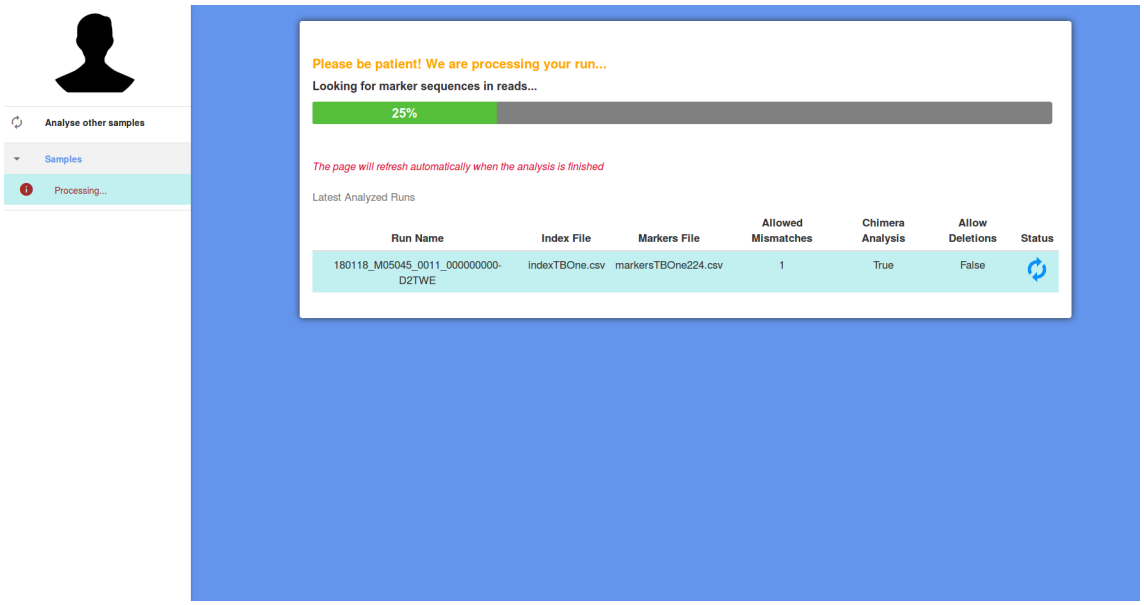


FIGURE 4.24 – L'interface RT-MiS permet le suivi de la progression d'une analyse en temps réel.

si besoin, ou encore, de lancer plusieurs analyses simultanément. Une fois l'analyse terminée, l'interface lance une étape de vérification dont le but est de s'assurer que tout s'est bien passé. Si la vérification réussit, l'utilisateur sera redirigé vers la page des résultats. D'autre part, si la vérification échoue, l'interface redirigera l'utilisateur vers une page d'erreur pour le notifier que son analyse n'a pas réussi. Généralement, une analyse n'échoue que si les fichiers des marqueurs et/ou des index ne sont pas compatibles avec le fichier FASTQ analysé.

Le fait que les analyses soient lancées en arrière plan permet à l'utilisateur d'en lancer plusieurs simultanément. Tenant compte du fait qu'une analyse n'est pas instantanée (la durée d'exécution dépend du nombre de marqueurs et de la taille du fichier FASTQ, elle est comprise entre 1 et 5 minutes), la mise en place d'un système de gestion des analyses fut obligatoire. Ce système, présenté dans la Figure 4.25, a cinq buts principaux :

- il permet à l'utilisateur de surveiller en temps réel la progression d'une analyse lancée.
- Il permet à l'utilisateur d'accéder aux résultats des analyses déjà réalisées et terminées.
- Lors du lancement d'une analyse, le système vérifie qu'elle n'est pas identique à une analyse qui est déjà en cours. Sinon, il redirigera l'utilisateur vers la page de progression de l'analyse lancée en premier et ne lancera pas la seconde.
- Lors du lancement d'une analyse, il vérifie qu'elle n'est pas identique à une analyse qui est déjà terminée et dont les résultats sont déjà obtenus. Sinon, il redirigera l'utilisateur vers la page des résultats de la première et ne lancera pas la seconde.
- Lors du lancement d'une analyse, ce système vérifie si elle ressemble à une analyse déjà terminée et dont les résultats sont obtenus.

Pour tester la ressemblance entre les analyses, ce système se base sur le nom du fichier analysé, le nom du fichier d'index, le nom du fichier des marqueurs, le type de l'analyse et les paramètres de la recherche approchée. S'il trouve que l'analyse lancée ressemble à une analyse déjà réalisée, il proposera à l'utilisateur le choix d'accéder directement aux résultats d'une des analyses similaires ou de lancer son analyse avec les paramètres qu'il avait choisi au début.

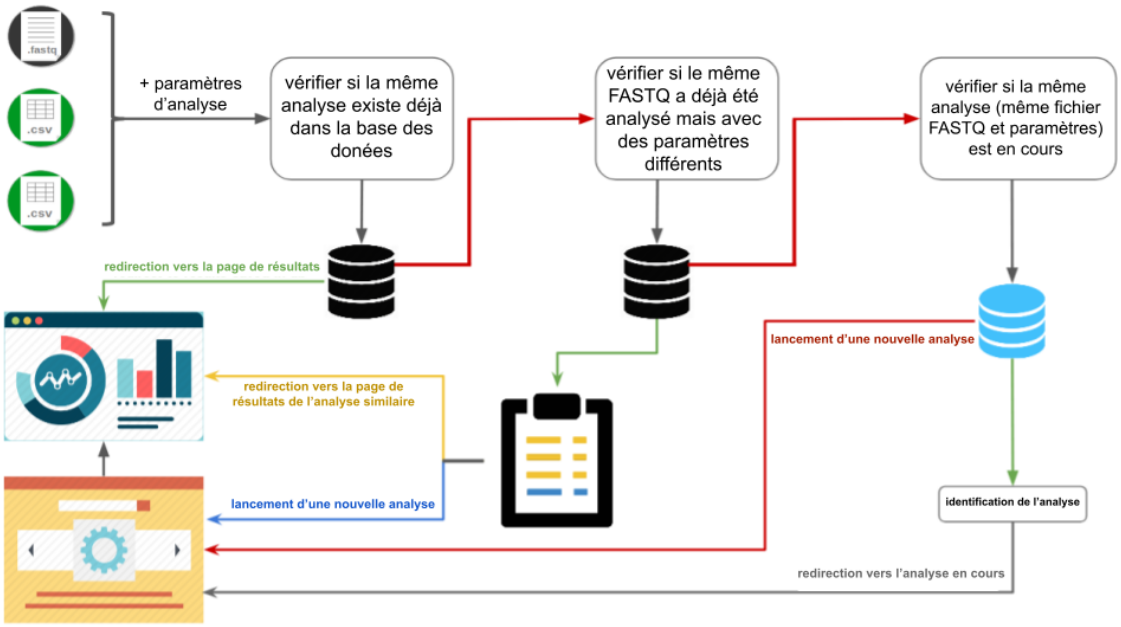


FIGURE 4.25 – Le système de gestion des analyses implémenté par l'interface RT-MiS. Les flèches en rouge représentent une opération ayant échouée tandis que les flèches en vert représentent une opération ayant réussie.

4.5.2.3 L'affichage des résultats

La page des résultats est composée de quatre parties principales :

1. analyse et statistiques globales ;
2. analyse et statistiques spécifiques à chaque index ;
3. une barre de navigation à gauche pour naviguer entre les échantillons ;
4. analyse des chimères dans le cas d'une analyse GEP.

En accédant aux résultats d'une analyse d'un fichier, la page d'accueil montre des informations générales sur l'analyse telles que le nom du fichier analysé, sa date de lancement et le nombre total de *reads* contenus dans le fichier (Figure 4.26). De plus, on trouve les noms des fichiers des index et de marqueurs utilisés pour effectuer cette analyse. Au-dessous, deux graphiques renseignent l'utilisateur sur le pourcentage d'informativité du fichier analysé. L'informativité représente le nombre total des *reads* dans lesquels une séquence d'index et une séquence d'un marqueur ont été trouvées. L'informativité sur cette page représente l'informativité moyenne du fichier, donc la moyenne d'informativité pour tous les index. Deux graphiques sont affichés sur cette page : le premier représentant l'informativité moyenne exacte (en ne considérant que les séquences retrouvées par une recherche exacte) et le second représentant l'informativité moyenne approchée (en considérant les séquences retrouvées suite à une recherche exacte ou approchée). Finalement, à partir de cette page, on peut télécharger tous les fichiers CSV générés par l'analyse.

Ensuite, en choisissant un index parmi la liste des index dans la barre de navigation, les données d'expression et l'informativité de chaque index sont affichées (Figure 4.27). De plus, la classe de l'échantillon est donnée seulement dans le cas d'une analyse de type GEP (Figure 4.28). L'informativité est représentée sous forme d'un histogramme horizontal affichant le pourcentage de *reads* ayant à la fois la séquence de l'index en question et une séquence d'un des marqueurs recherchés. Au-dessous, on a le nombre total de *reads*

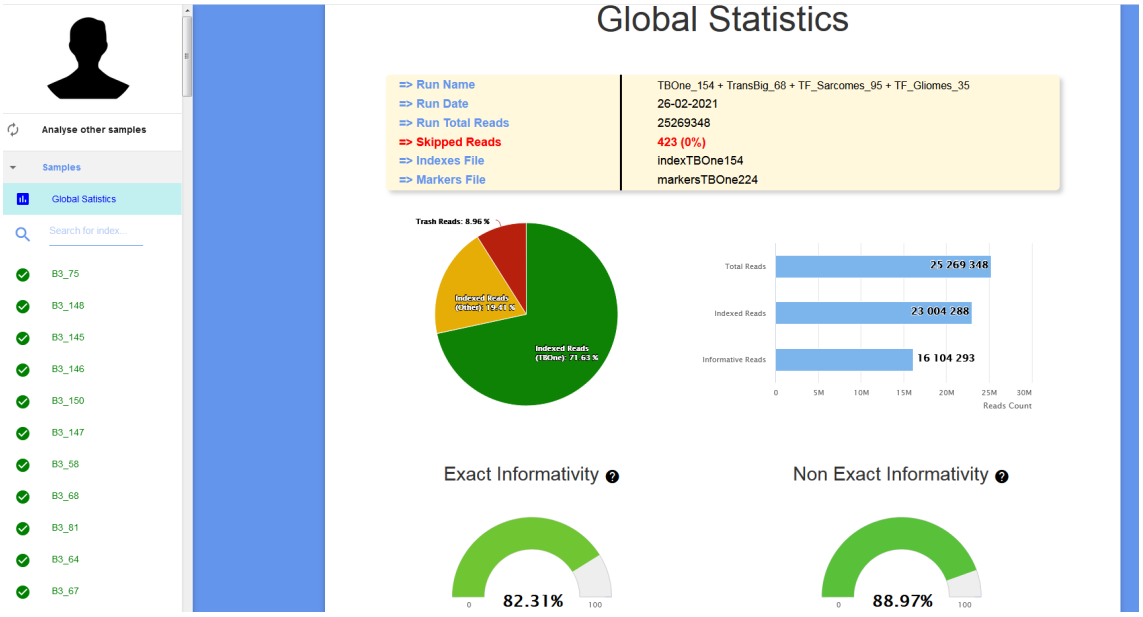


FIGURE 4.26 – La page des résultats montrant les statistiques globales d'une analyse lancée sur l'interface RT-MiS.

trouvés portant cet index ainsi qu'une estimation du facteur d'amplification spécifique à cet index. De plus, on y trouve les profils d'expression de chaque marqueur/gène présent dans le fichier des marqueurs. Le niveau d'expression correspond au nombre de *reads* total portant la séquence de l'index en question avec la séquence du gène exprimé. Les niveaux d'expression sont triés par ordre décroissant facilitant la détection des gènes surexprimés chez un patient et rendant l'interprétation globale du profil plus facile. Des onglets spécifiques à chaque index permettent d'accéder aux résultats de recherche exacte seule ou aux résultats de recherche exacte et approchée. Depuis cette page, on peut télécharger les tableaux de comptage de chaque index ainsi que les séquences de tous les *reads* portant cet index.

Enfin, dans le cas d'une analyse GEP, l'interface affiche l'analyse des chimères, présentée dans la Figure 4.29. De la même façon, un histogramme se trouve tout en haut pour afficher le pourcentage de chimères trouvées dans le fichier analysé. Normalement, ce pourcentage doit être faible (inférieur à 0,1%) pour s'assurer de la bonne qualité du séquençage. Le pourcentage est calculé en divisant le nombre total des *reads* portant des chimères sur le nombre total des *reads* du fichier FASTQ. Au-dessous du pourcentage, une *heatmap* est affichée et montre le nombre de *reads* retrouvés pour chaque chimère. La *heatmap* reprend le contenu exact du fichier généré par l'algorithme de l'analyse (contenant un tableau bidimensionnel avec l'expression de chaque chimère) et l'affiche d'une manière beaucoup plus facile à interpréter par les chercheurs. Le fichier de la matrice des chimères est également téléchargeable depuis cette page.

4.6 Synthèse

Dans ce chapitre, nous avons présenté la RT-MLPA classique ainsi que la nouvelle méthode consistant à utiliser la RT-MLPA couplée à un séquenceur NGS. Cette méthode peut être utilisée dans plusieurs applications et est actuellement utilisée dans le domaine de la transcriptomique par l'unité Inserm 1245 au Centre Henri Becquerel pour effectuer des mesures d'expression génique afin de classifier les échantillons et aussi pour

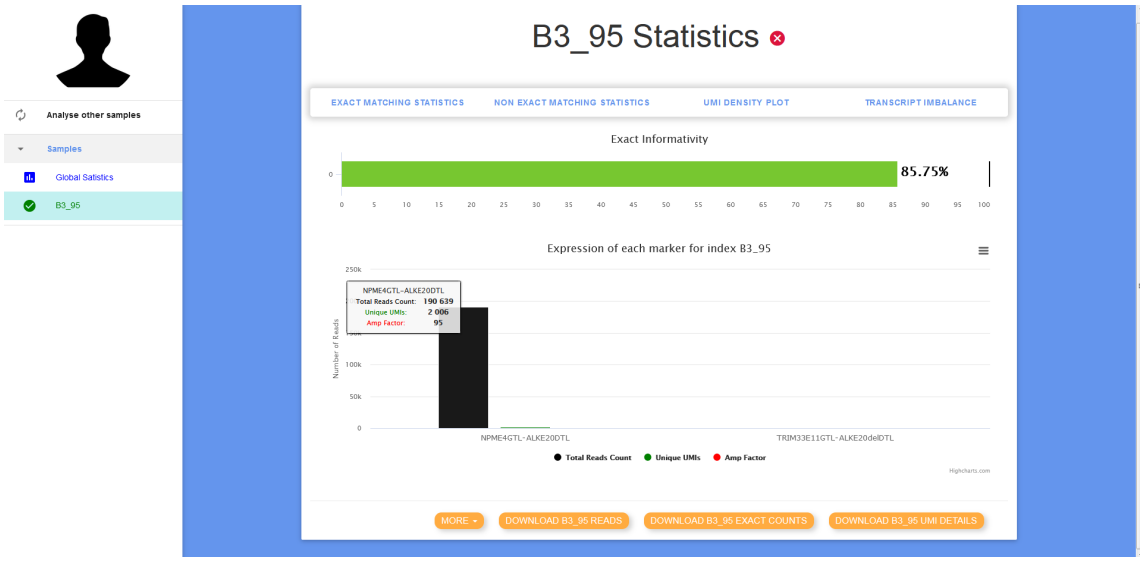


FIGURE 4.27 – La page des résultats montrant les niveaux d'expression d'une analyse de recherche des transcrits de fusion lancée sur l'interface RT-MiS.

FIGURE 4.28 – La page des résultats montrant la classification des échantillons suite à une analyse de type GEP lancée sur l'interface RT-MiS.

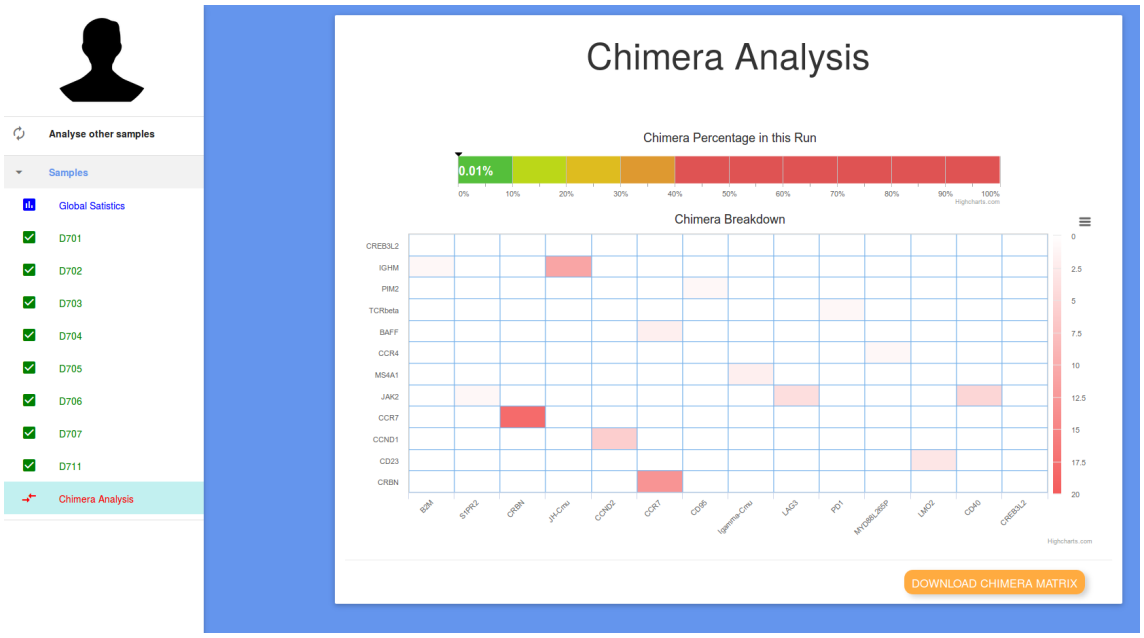


FIGURE 4.29 – La page des résultats montrant l'analyse des chimères suite à une analyse de type GEP lancée sur l'interface RT-MiS.

rechercher des transcrits de fusion dans les ARN extraits, un élément très utile dans le diagnostic de différents cancers. Nous avons présenté l'outil RT-MiS, un logiciel permettant d'extraire les UMI et filtrer les *reads* d'un fichier FASTQ. Ensuite, en lui fournissant un fichier d'index et un fichier des marqueurs, il est capable d'effectuer des recherches exactes et approchées dans chacun des *reads* pour l'attribuer au bon index et identifier la séquence du marqueur qu'il exprime. Les algorithmes de recherche utilisés sont très efficaces en terme de temps et de consommation en mémoire : le temps et la consommation en mémoire augmentent linéairement avec la taille du FASTQ, la profondeur du séquençage et le nombre de marqueurs. RT-MiS intègre aussi une méthode de correction des UMI permettant de supprimer les doublons de PCR et retrouver le nombre de molécules uniques avec une grande efficacité. La correction des UMI est très utile, voire indispensable dans des applications telles que la détection des transcrits de fusion puisque la présence des doublons de PCR est susceptible de fausser les résultats de l'analyse. Dans le cas d'une analyse pour la mesure d'expression génique, une analyse des chimères est effectuée reflétant le degré de réussite de l'expérience. De plus, un modèle de classification de type *Random Forest* est appelé pour classer les échantillons analysés selon les niveaux d'expression de chaque gène. Cet outil a été implémenté dans une interface *web* permettant de gérer automatiquement la plupart des étapes de l'analyse, de la récupération du fichier FASTQ brut jusqu'à la production des résultats sous formes de graphiques interactifs, plus pratiques à utiliser et plus faciles à interpréter. L'interface d'analyse RT-MiS est aujourd'hui l'un des principaux outils d'analyse au CHB ; elle constitue une énorme base de données de tous les patients séquencés, analysés et classés grâce à cet outil et permettant, après chaque analyse, d'entraîner à nouveau le modèle de classification pour l'améliorer.

Chapitre 5

Détection des variants somatiques

5.1 Introduction

Dans le Chapitre 4, nous avons présenté comment les UMI peuvent être utiles dans le domaine de la transcriptomique. Cependant, ils peuvent avoir une très grande utilité aussi dans le domaine de la génomique et surtout, pour la détection des variants somatiques. Le *variant calling* est une étape de l'analyse bioinformatique secondaire consistant à parcourir l'ensemble des fragments d'ADN séquencés pour comparer les bases de chaque position à un génome de référence. En théorie, cette opération ne semble pas très compliquée vu qu'il existe un très grand nombre d'algorithmes efficaces pour la comparaison des séquences. Mais en pratique, le faible pourcentage des cellules tumorales dans les échantillons, les erreurs de PCR ainsi que les erreurs de séquençage insérées aléatoirement dans les séquences des *reads* contribuent tous ensemble à rendre la tâche plus fastidieuse, surtout pour les variants de faible fréquence. Pour cela, les *variant callers* implémentent différents modèles statistiques et des filtres sur certains critères pour distinguer entre les vrais variants et les faux positifs. De plus, l'utilisation des UMI a permis de reconnaître les doublons de PCR pour en tenir compte lors de l'analyse des variants permettant alors une suppression de la plupart des faux positifs. Dans ce chapitre, nous présenterons UMI-VarCal, un *variant caller* spécialement conçu pour la détection des variants somatiques (substitutions, insertions et délétions) dans les échantillons d'ADN séquencés tout en assurant une analyse très efficace en terme de temps et de consommation en mémoire. Dans ce qui suit, la problématique de la détection des variants de faible fréquence est expliquée. L'approche classique de *variant calling* sera brièvement présentée ainsi que l'approche utilisée par UMI-VarCal pour assurer une grande précision des résultats obtenus.

5.2 Problème des variants de très faible fréquence

Le nombre de mutations cliniquement pertinentes a conduit au développement de tests à haut débit pour détecter les mutations somatiques directement à partir d'échantillons de cancer en combinant la capture ciblée avec des plateformes de séquençage de nouvelle génération (NGS). Ces approches peuvent fournir un profil mutationnel complet sur un grand nombre de gènes simultanément, avec une très grande efficacité et un coût réduit par rapport aux méthodes de séquençage conventionnelles. À ce jour, cette approche a été mise en œuvre avec succès dans plusieurs laboratoires cliniques pour la détection de mutations somatiques dans le cancer. Malgré la promesse de ces approches, le *variant calling* utilisé pour détecter les mutations somatiques dans le cancer posent des défis techniques uniques par rapport à ceux utilisés pour détecter les variants constitutionnels. On parle de mutation constitutionnelle lorsqu'une mutation est présente ou survient avant la fécondation (soit nouvellement apparue, soit transmise de génération

en génération), ou survient lors des premières divisions du zygote (donc nouvellement apparue). Les échantillons de cancer provenant de petits échantillons de biopsie peuvent contenir peu de cellules tumorales, et les échantillons petits et grands peuvent contenir une grande quantité de cellules normales et de cellules inflammatoires. Cela peut entraîner une dilution de l'ADN tumoral avec celui des cellules non tumorales, et donc les mutations somatiques présentes dans chaque cellule tumorale seront finalement observées à de basses fréquences dans l'ADN de l'échantillon. De plus, des études récentes sur les cancers solides et hématologiques ont démontré une remarquable hétérogénéité génétique ; bien que les tumeurs soient généralement dérivées d'un seul clone fondateur défini par des variants somatiques présents dans chaque cellule tumorale, il existe souvent plusieurs sous-populations de cellules tumorales avec des variants somatiques supplémentaires [108, 109, 110, 111, 112]. D'autre part, les aneuploïdies (cellule qui ne possède pas le nombre normal de chromosomes) acquises somatiquement et la variation du nombre de copies sont courantes dans de nombreuses tumeurs solides et les gains ou la perte de matériel génétique peuvent modifier la fraction allélique observée des variants de séquence dans ces régions. Les méthodes de détection basées sur le NGS peuvent capturer des mutations à basse fréquence car elles fournissent une lecture numérique des variants de séquence et une redondance de séquençage élevée allant de centaines à des milliers de molécules d'ADN individuelles. Cependant, la détection *a priori* des mutations à basse fréquence (c'est-à-dire la détection de variants à des positions non *hotspot* (les positions *hotspot* sont des positions généralement très touchées par des mutations) dans lesquelles la probabilité précédente d'un variant est faible repose sur des méthodes capables de différencier les vrais variants du bruit telles que les erreurs de séquençage, les erreurs de PCR et les artefacts d'alignement. Sans ces algorithmes, un grand nombre de variants faussement appelés par le *variant caller* ne représenterait que le taux d'erreur inhérent des séquenceurs NGS, qui s'approche de 1%. Actuellement, de nombreux programmes d'analyse NGS populaires sont conçus pour l'analyse constitutionnelle du génome dans laquelle des variants devraient se produire dans 50% (hétérozygote) ou 100% (homozygote) des *reads*. Ces probabilités sont souvent intégrées dans les algorithmes de détection, et les variants avec des fréquences alléliques (VAF) tombant trop en dehors de la plage attendue pour les variants homozygotes et hétérozygotes peuvent être considérés comme de mauvaise qualité et ne pas être appelés en raison de la forte probabilité qu'ils soient faussement positifs plutôt que des variants héréditaires. Pour répondre à ce problème, plusieurs logiciels ont été spécifiquement développés pour la détection de variants à basse fréquence et qui ont réussi à détecter des mutations à moins de 0,1%. Ces méthodes nécessitent généralement une préparation de bibliothèque spécialisée, des échantillons de contrôle enrichis, des modèles statistiques compliqués ou d'autres modifications des protocoles de laboratoire NGS standards pour détecter les variants avec moins de 2% de VAF [113]. Cependant, pour la grande majorité de ces outils, la détection des variants de faible fréquence est souvent accompagnée d'un grand nombre de faux positifs, reflétant l'incapacité de ces algorithmes à distinguer efficacement entre vrai et faux positif.

5.3 Approche classique du *variant calling*

Plus d'une vingtaine de *variant callers* existe actuellement pour la détection des variants somatiques et/ou constitutionnels sans utilisation des UMI. Chacun de ces outils utilise un algorithme, une loi statistique et des filtres spécifiques afin de détecter le plus de variants tout en réduisant le taux de faux positifs. Deux métriques sont généralement utilisées pour comparer les *variant callers* : la sensibilité et la spécificité. Ces deux valeurs

représentent une relation entre quatre critères différents : le taux de vrais positifs (VP) (les vrais variants qui ont été détectés), le taux de faux positifs (FP) (des erreurs de séquençage et/ou de PCR détectées en tant que variants), le taux de vrais négatifs (VN) (les positions non appelées par le *variant caller* et où aucun variant n'est présent) et le taux de faux négatifs (FN) (les vrais variants ratés par le *variant caller)*. La sensibilité représente la capacité d'un outil de détecter le plus grand nombre de variants et donc de réduire au maximum le taux de faux négatifs. Elle est calculée par l'équation ci-dessous :

$$sensibilité = \frac{VP}{VP + FN} \tag{5.1}$$

La meilleure façon d'augmenter la sensibilité d'un outil est de le rendre le plus permissif possible. En conséquence, il ne ratera aucun vrai variant mais détectera beaucoup de faux positifs en même temps. Pour cela, la sensibilité n'est pas suffisante toute seule pour déduire le meilleur outil et elle est toujours accompagnée d'un calcul de la spécificité. Cette dernière représente la capacité d'un *variant caller* de filtrer le mieux possible les faux positifs. Ainsi, le meilleur outil serait celui qui maximise la sensibilité et la spécificité.

$$spécificité = \frac{VN}{VN + FP} \tag{5.2}$$

Différents modèles et lois statistiques sont utilisés par les *variant callers* pour atteindre une meilleure précision. Par exemple, BAYSIC [114] et MutationSeq [54] utilisent un modèle d'apprentissage automatique alors que des outils tels que LoFreq [52], EBCall [115] et deepSNV [116] appliquent une analyse de fréquence allélique pour rendre leurs résultats. D'autre part, des logiciels tels que SiNVICT [56] et Pisces [117] se servent d'un modèle de Poisson pour appeler les variants. De plus, de nombreux outils ont recours à une estimation du bruit de fond pour supprimer les faux positifs, comme ce que font outLyzer [55], SomVarIUS [118] et SPLINTER [119]. Enfin, un outil, MuSE [120], se distingue de tous les autres puisqu'il applique un modèle de Markov caché, ou HMM (*Hidden Markov Model*) pour calculer la probabilité de chaque position d'abriter un vrai variant. Un autre critère de différenciation entre les *variant callers* est leur capacité à détecter les variants sans avoir besoin d'un échantillon contrôle apparié (*matched normal sample*). Cette obligation peut parfois être très limitante vu que dans la grande majorité des cas, un échantillon contrôle apparié est impossible à obtenir. Tous ces outils que nous avons cités appliquent la même approche classique pour la détection des variants : analyser les *reads* position par position, les comparer à un génome de référence pour détecter les bases alternatives (bases observées au lieu de la base de référence) et appliquer un modèle statistique déterminé pour différencier entre une vraie mutation et un faux positif. Certains outils appliquent aussi des filtres complémentaires pour supprimer encore plus de faux positifs. Deux filtres sont les plus utilisés : le filtre de biais de brin et le filtre sur les homopolymères. Un variant avec un biais de brin représente un variant dont les bases alternatives sont majoritairement observées sur un des deux brins. Ceci peut faire penser que ce variant est, en effet, une erreur de séquençage due à une difficulté du séquenceur de bien lire la bonne base sur l'un des brins. Ainsi, ce variant serait traité comme un artefact et, par conséquent, filtré. Le filtre sur les régions homopolymériques est utile pour supprimer les variants détectés dans des régions contenant des homopolymères (une même base répétée plusieurs fois de suite). En effet, plusieurs séquenceurs (surtout l'Ion Torrent) font des erreurs lors de la lecture d'une longue région homopolymérique. Ce phénomène est causé par une saturation du signal et mène alors à une sous-estimation ou surestimation du nombre de bases répétées, créant ainsi une délétion ou une insertion. Dans ce qui suit, nous allons présenter deux *variant allers* classiques, SiNVICT et outLyzer, spécialement

conçus pour la détection des variants à faible fréquence.

5.3.1 SiNVICT

SiNVICT est un *variant caller* développé par C. Kockan *et al.* en 2017. SiNVICT a été conçu pour la détection des variants à faible fréquence et n'a pas besoin d'un échantillon normal apparié pour effectuer son analyse. D'autre part, il possède des caractéristiques faisant de lui un *variant caller* bien distinct et l'un des plus performants lorsque la fréquence allélique tombe au dessous de 1%. Tout d'abord, SiNVICT accepte deux types de fichiers comme entrée : FASTQ ou BAM/SAM. Si un FASTQ est fourni, une étape de prétraitement des *reads* est effectuée pour supprimer les adaptateurs, recalibrer les scores de qualité des bases et aligner les *reads* sur le génome de référence. Cette étape assure la production d'un fichier BAM/SAM de haute qualité. Si l'utilisateur fournit directement un BAM, le prétraitement est ignoré et l'identification des variants est automatiquement lancée. Pour identifier les vrais variants des erreurs de séquençage, SiNVICT applique un modèle statistique en réalisant un test de Poisson à chaque position. En effet, le test de Poisson permet de calculer une *p-value* et par la suite une *q-value* (*p-value* corrigée) pour filtrer les variants : seuls ceux ayant une *q-value* $< \alpha$ sont considérés. α représente le risque de première espèce : le risque d'obtenir un faux positif. En général, cette valeur est fixée à 5% mais elle peut être plus faible parfois pour rendre le test plus prudent et donc augmenter la spécificité de l'outil. Si un variant potentiel réussit le test de Poisson, il subit quatre filtres complémentaires :

1. la profondeur de la position du variant doit être supérieure à un seuil.
2. Le variant doit passer le filtre de biais de brin.
3. Le variant ne doit pas se situer dans une région homopolymérique.
4. Si plusieurs échantillons sont analysés, SiNVICT est capable de calculer le bruit de fond moyen sur une région donnée. Dans ce cas, il faut que la VAF du variant soit supérieure au bruit de fond calculé.

Tous ces filtres sont modifiables par l'utilisateur et les variants qui passent ces critères sont reportés dans le fichier VCF final des résultats. De plus, SiNVICT est capable d'effectuer la classification entre variant somatique et constitutionnel. Le *workflow* de l'outil SiNVICT est présenté dans la Figure 5.1. Il peut également fonctionner sur la même tumeur à plusieurs stades et effectuer une analyse de séries chronologiques, ce qui est particulièrement utile pour comprendre l'évolution de la tumeur.

Pour mettre en évidence sa performance, SiNVICT a été testé contre trois autres outils (MuTect [53], Freebayes [82] et VarScan2 [61]) sur des données simulées où 18 SNV aléatoires ont été ajoutés dans une copie d'un échantillon normal de façon à obtenir sept niveaux de mélange tumeur-normal différents (50, 20, 10, 5, 2,5, 1 et 0,5% de taux de tumeur). Pour les comparer, deux métriques ont été utilisées : la sensibilité et la valeur prédictive positive ou PPV (*Positive Predictive Value*). La PPV représente le rapport entre le nombre de vrais variants appelés par l'outil et le nombre total d'appels. C'est une autre façon de représenter la spécificité d'un outil et elle est calculée par la formule ci-dessous :

$$PPV = \frac{VP}{VP + FP} \tag{5.3}$$

SiNVICT était très sensible sur cet ensemble de données simulées : il était capable de détecter les 18 mutations (en tant que SNV à haute confiance) à des niveaux de contenu tumoral de 50, 20, 10, 5 et 2,5%. À un taux de contenu tumoral de 1%, SiNVICT a pu détecter 13 des 18 mutations et à un taux de 0,5%, il a détecté 12 des 18 mutations. Dans

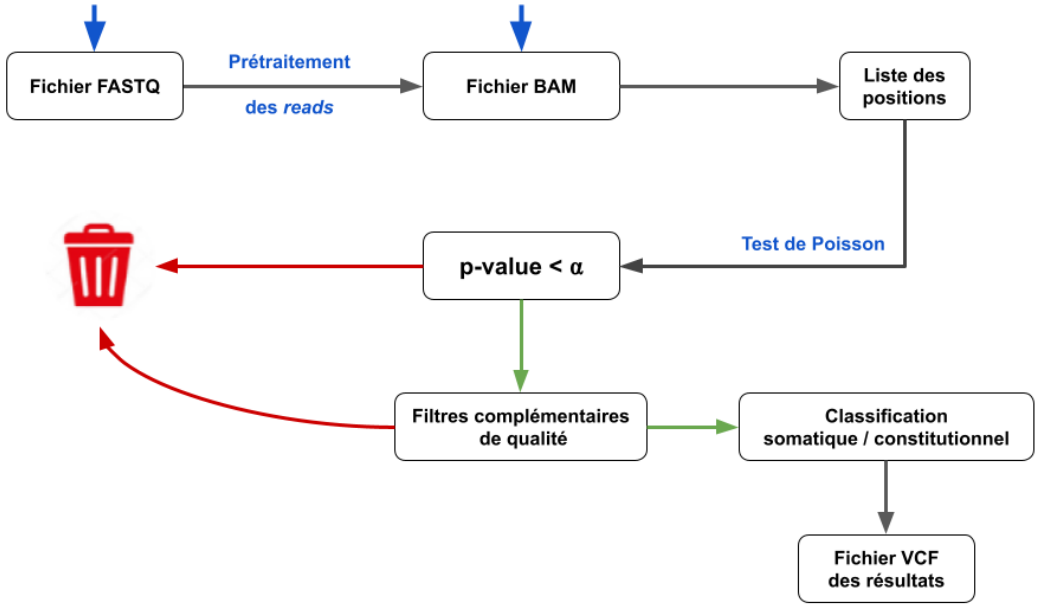


FIGURE 5.1 – Le *workflow* de l'outil SiNVICT. Les flèches en rouge représentent une opération ayant échouée tandis que les flèches en vert représentent une opération ayant réussie. Les flèches en bleu présentent les différentes entrées de l'outil.

tous les cas, SiNVICT avait une meilleure PPV (et donc spécificité) que MuTect, Freebayes et VarScan2. En ce qui concerne la sensibilité, SiNVICT n'est dépassé que dans un contenu tumoral de 1% par VarScan2 qui arrive à détecter un SNV de plus que lui. Les résultats de cette comparaison sont illustrés dans la Figure 5.2.

5.3.2 OutLyzer

OutLyzer est un *variant caller* développé par E. Muller *et al.* en 2016 et qui, comme SiNVICT, a été conçu pour la détection des variants à faible fréquence dans les échantillons de mauvaise qualité par exemple. Il est capable d'effectuer son analyse sans avoir besoin d'un échantillon normal apparié ce qui est très pratique. Pour identifier les vrais variants des erreurs de séquençage, outLyzer applique un modèle d'identification des valeurs aberrantes : le test Tau modifié de Thompson. En effet, en utilisant ce test, l'outil est capable d'estimer le bruit de fond dans une région déterminée et alors d'appeler les variants dont la fréquence dépasse un certain seuil. Si un variant potentiel réussit le test, il subit quatre filtres complémentaires :

1. le nombre de *reads* mutés doit être supérieur au double du bruit de fond.
2. Le score de qualité Phred moyen du variant doit être > 20.
3. Le score de qualité Phred moyen doit avoir un écart type < 7.
4. Le ratio entre les observations du variant sur chaque brin doit être entre 0,3 et 0,7.

Tous ces filtres sont personnalisables par l'utilisateur et les variants qui passent ces critères sont reportés dans le fichier VCF final des résultats. Le *workflow* de l'outil outLyzer est présenté dans la Figure 5.3. Contrairement à SiNVICT, outLyzer ne fait pas la différence entre variant somatique ou constitutionnel.

Pour illustrer les performances des trois *variant callers* testés contre outLyzer (HaplotypeCaller [78], LoFreq et VarScan [121]), un échantillon commercial (HorizonDX) dans

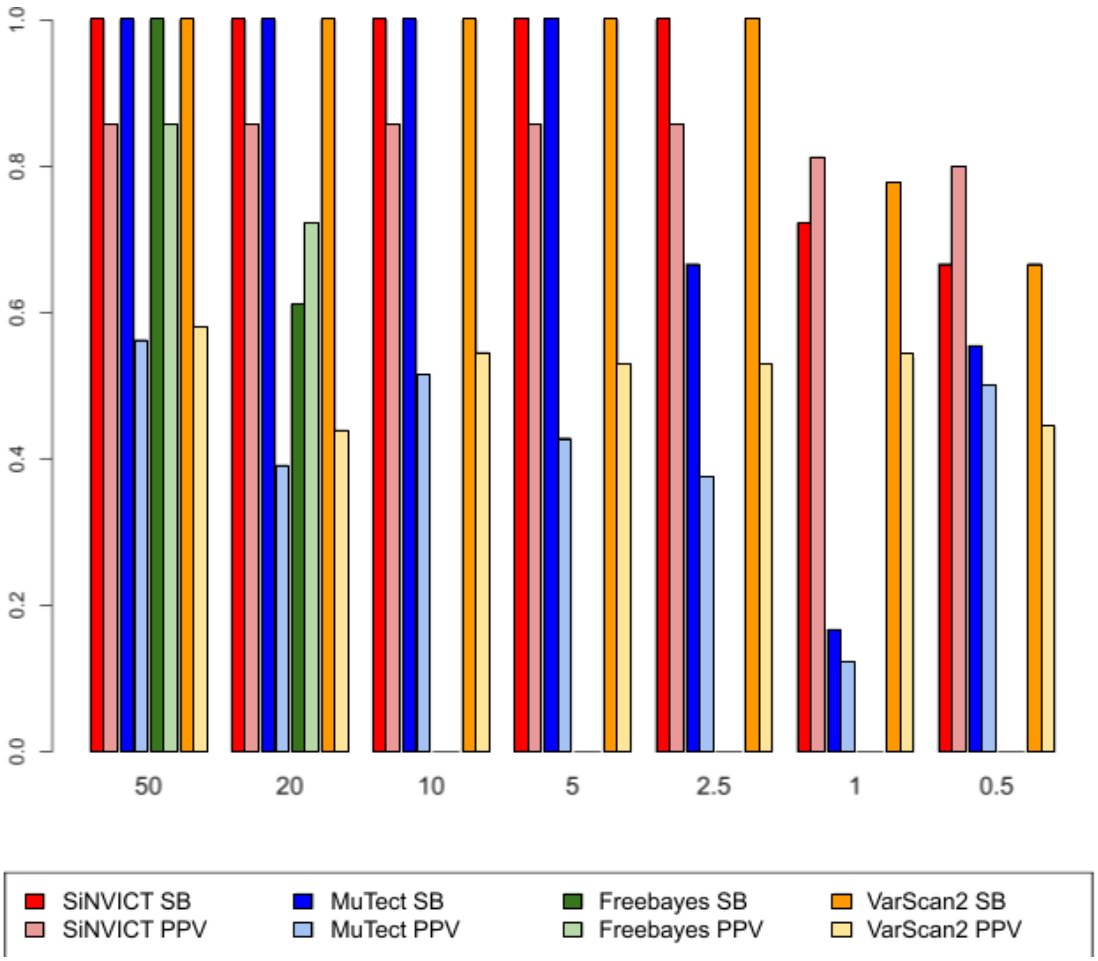


FIGURE 5.2 – Comparaison de la performance entre SiNVICT, MuTect, Freebayes et VarScan2 en terme de sensibilité et de PPV. Figure adaptée de [56]

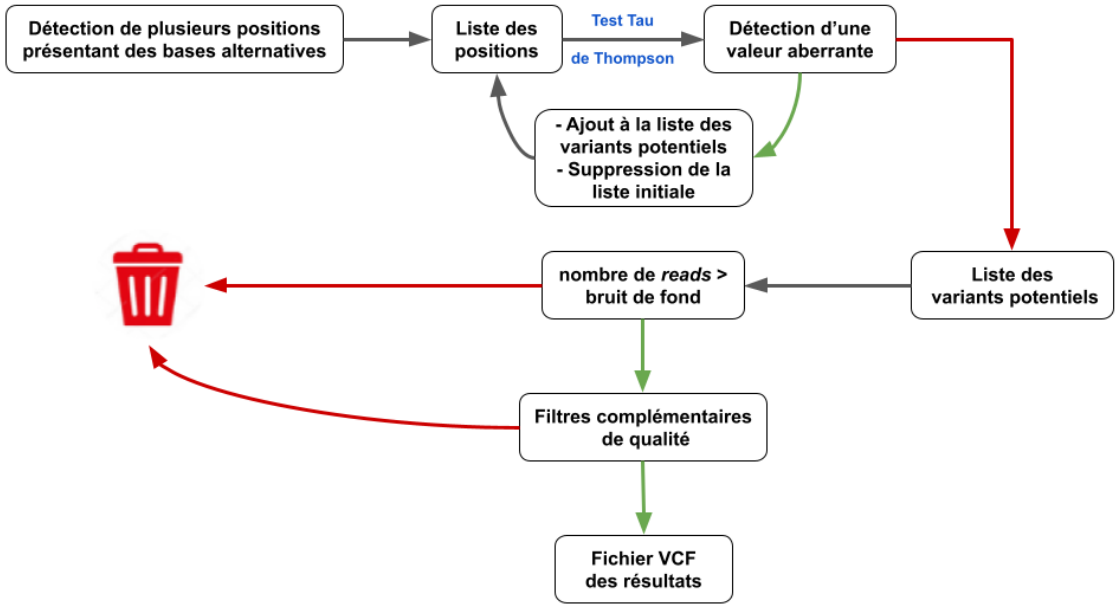


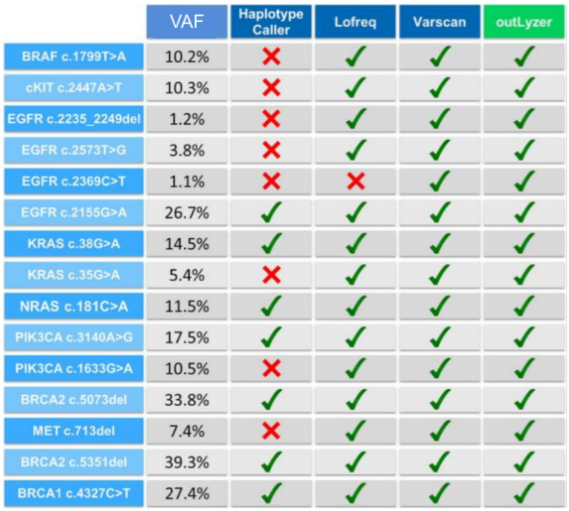
FIGURE 5.3 – Le *workflow* de l'outil outLyzer. Les flèches en rouge représentent une opération ayant échouée tandis que les flèches en vert représentent une opération ayant réussie.

lequel un certain nombre de mutations avec différentes VAF connues a été séquencé et analysé indépendamment des autres échantillons. LoFreq réussit à détecter toutes les mutations sauf une montrant une très bonne sensibilité et un seuil de détection aux alentours de 1%. Par contre, HaplotypeCaller conçu pour détecter les variants constitutionnels n'arrivent pas à appeler les variants avec une VAF < 11%. Seuls outLyzer et VarScan ont pu détecter toutes les mutations attendues ce qui suggère que leur seuil de détection est inférieur à 1%. Les résultats sont présentés dans la Figure 5.4.

5.4 Nouvelle approche par analyse des UMI

Les développements récents dans le diagnostic précoce du cancer ont augmenté la demande de détection de l'ADN tumoral circulant (*ctDNA*). Dans ces applications, les outils doivent être capables de détecter des variants à des fréquences de 0,1%, voire moins, étant donné la quantité infime d'ADN dans le sang. Même si tous les différents *variant callers* cités ci-dessus effectuent un très bon travail pour détecter la plupart des variants pertinents dans les échantillons, leur spécificité commence à chuter considérablement lorsqu'il s'agit d'appeler des variants dont la fréquence est inférieure à 1%. Ceci est principalement dû au fait que beaucoup d'erreurs de séquençage ont des fréquences inférieures à cette limite et que les modèles statistiques ne sont pas efficaces à les distinguer des vrais variants. Ainsi, ces outils sont obligés de réduire leur restriction quant aux seuils des filtres de qualité utilisés. En conséquence, un grand nombre de variants est détecté à des faibles fréquences mais il n'y a aucun vrai moyen pour filtrer les faux positifs.

L'utilisation des UMI représente une méthode très efficace pour résoudre ce problème. Les séquences des UMI s'attachent sur chaque fragment initial de l'ADN séquencé et constitue une sorte d'étiquette unique à chacun d'eux. L'étape de l'introduction des UMI est effectuée avant l'amplification ce qui permet de regrouper les *reads* en groupes d'UMI : un groupe d'UMI représente un ensemble de *reads* étiquetés par la même séquence d'UMI, et donc provenant d'un même fragment d'ADN initial. Dans ce cas, si le



	VAF	Haplotype Caller	Lofreq	Varscan	outLyzer
BRAF c.1799T>A	10.2%	✗	✓	✓	✓
cKIT c.2447A>T	10.3%	✗	✓	✓	✓
EGFR c.2235_2249del	1.2%	✗	✓	✓	✓
EGFR c.2573T>G	3.8%	✗	✓	✓	✓
EGFR c.2369C>T	1.1%	✗	✗	✓	✓
EGFR c.2155G>A	26.7%	✓	✓	✓	✓
KRAS c.38G>A	14.5%	✓	✓	✓	✓
KRAS c.35G>A	5.4%	✗	✓	✓	✓
NRAS c.181C>A	11.5%	✓	✓	✓	✓
PIK3CA c.3140A>G	17.5%	✓	✓	✓	✓
PIK3CA c.1633G>A	10.5%	✗	✓	✓	✓
BRCA2 c.5073del	33.8%	✓	✓	✓	✓
MET c.713del	7.4%	✗	✓	✓	✓
BRCA2 c.5351del	39.3%	✓	✓	✓	✓
BRCA1 c.4327C>T	27.4%	✓	✓	✓	✓

FIGURE 5.4 – Comparaison des *variant callers* sur un ensemble de mutations somatiques avec des VAF connues. Figure adaptée de [55].

fragment d'ADN initial portait une mutation déterminée, cette dernière devrait se retrouver sur la plupart, voire la totalité des séquences étiquetées par ce même UMI. D'autre part, si au sein d'un groupe d'UMI, un variant est retrouvé sur quelques *reads* seulement, cela voudra dire que la variation est apparue après l'introduction de l'UMI et donc n'était pas présente sur le fragment initial : ce sera un artefact causé par une erreur lors de l'amplification par PCR ou lors du séquençage.

Ce principe a conduit au développement de trois autres *variant callers* utilisant les UMI dans leur analyse : DeepSNVMiner [58], MAGERI [57] et smCounter2 [80]. Les *workflows* de ces outils ont été détaillés dans le Chapitre 3. DeepSNVMiner génère d'abord une liste initiale de variants à l'aide de SAMtools *calmd*, puis compare les UMI des *reads* portant la mutation pour filtrer les faux positifs. MAGERI construit une lecture de consensus pour chaque groupe d'UMI et adopte une approche de modélisation bêta-binomiale pour estimer le taux d'erreur de l'ADN polymérase en utilisant des données externes. De plus, MAGERI suppose une distribution bêta universelle sur tous les sites plutôt que des taux d'erreur spécifiques au site. smCounter2 génère implicitement l'appel de base consensus position par position et calcule la probabilité postérieure du variant en considérant conjointement les erreurs de PCR et de séquençage. DeepSNVMiner et MAGERI sont tous deux des *pipelines* de bout en bout dotés de fonctions intégrées d'extraction, d'alignement et de *variant calling*. Quant à lui, smCounter2 est un *variant caller* autonome qui prend les données d'alignement BAM/SAM comme entrée. Nous rappelons que MAGERI et smCounter2 ont des limites de détection d'environ 0,5% alors que DeepSNVMiner peut détecter des variants à 0,1%.

5.5 Développement d'UMI-VarCal

5.5.1 Introduction

Bien que les outils classiques sont très performants pour la détection des variants somatiques, leur limitation principale est leur incapacité à bien distinguer entre les artefacts et les variants somatiques lorsque ces derniers sont présents à de très faible fréquence. En général, ces outils implémentent un seuil de fréquence allélique minimale au dessous duquel les variants détectés sont considérés comme artefacts et sont dont filtrés.

Dans le cas de recherche des variants à très faible fréquence, ce seuil doit être diminué menant à une augmentation significative du taux de faux positifs. Ainsi, pour répondre à ce problème, les *variant callers* basés sur les UMI utilisent ces derniers à la place du seuil pour filtrer les faux positifs. Les trois outils basés sur les UMI décrits dans le paragraphe précédent sont basés sur ce principe mais présentent quelques limitations : leur temps d'analyse est significativement plus élevée que les outils classiques, smCounter2 et MAGERI ont une limite de détection d'environ 0,5%, la consommation mémoire de MAGERI est très élevée et smCounter2 est conçu sur et pour des kits de séquençage spécifiques. Pour répondre à ces limitations, nous avons développé UMI-VarCal, un *variant caller* basé sur les UMI et conçu pour les protocoles de séquençage NGS ciblés en *paired-end*. UMI-VarCal se distingue des autres outils puisqu'il applique un algorithme de *pileup* spécialement conçu pour traiter efficacement les séquences UMI présentes dans les *reads*. Ceci lui permet d'être plus rapide que les outils utilisant les UMI mais aussi les *variant callers* classiques. Pour tester sa performance, nous l'avons comparé à deux des meilleurs logiciels classiques pour la détection des variants à faible fréquence et qui n'ont besoin que de l'échantillon tumoral pour effectuer leur analyse, SiNVICT et outLyzer. Nous démontrons également qu'il est aussi, voire plus spécifique et plus sensible que les *variant callers* basés sur les UMI en le comparant à DeepSNVMiner.

5.5.2 Implémentation

5.5.2.1 Fichiers d'entrée

UMI-VarCal est composé de deux outils : un outil d'extraction d'UMI et un outil de *variant calling*. L'outil d'extraction n'a besoin que d'un seul fichier en entrée : le fichier BAM/SAM ayant les UMI toujours dans les séquences. D'autre part, pour lancer l'outil de *variant calling* d'UMI-VarCal, trois fichiers sont nécessaires : le fichier d'alignement BAM/SAM, le fichier BED contenant les coordonnées des régions génomiques ciblées et un fichier FASTA du génome de référence. Pour faciliter l'utilisation, UMI-VarCal peut accepter des fichiers BAM ainsi que des fichiers SAM en entrée. Dans les deux cas, le fichier en entrée pour le *variant calling* doit avoir subit l'extraction des UMI. Cette extraction peut être faite soit par l'outil d'extraction des UMI d'UMI-VarCal, soit par un autre outil tel que UMI-tools. Les deux types de fichiers d'entrée sont présentés dans la Figure 5.5. UMI-VarCal peut accepter un quatrième fichier (facultatif) au format PILEUP. En effet, lors de l'exécution d'UMI-VarCal sur un échantillon, un fichier PILEUP est, par défaut, produit. Dans le cas où, par exemple, l'utilisateur souhaite réaliser un autre *variant calling* avec des paramètres différents, il a le choix de donner ce fichier en entrée au logiciel lors de l'exécution. Ceci permet de sauter l'étape de construction du *pileup* et de charger l'ancien à la place. Étant l'étape qui prend le plus de temps à s'exécuter, la capacité de charger directement le fichier PILEUP à la place fera gagner un temps significatif à l'utilisateur.

5.5.2.2 L'outil d'extraction des UMI

L'outil d'extraction implémenté dans UMI-VarCal est développé en C++ mais appelé par le script Python principal. Il prend en entrée un fichier d'alignement BAM ou SAM pour y extraire les UMI et les concaténer à la fin de l'identifiant du *read* comme UMI-tools. Pour l'utiliser, il suffit de lancer l'outil, lui donner le chemin du fichier d'alignement et préciser la longueur de la séquence UMI à extraire. Il ne peut extraire que les UMI présents au début de la séquence du *read*. L'extraction des UMI est présentée dans la Figure 5.5.

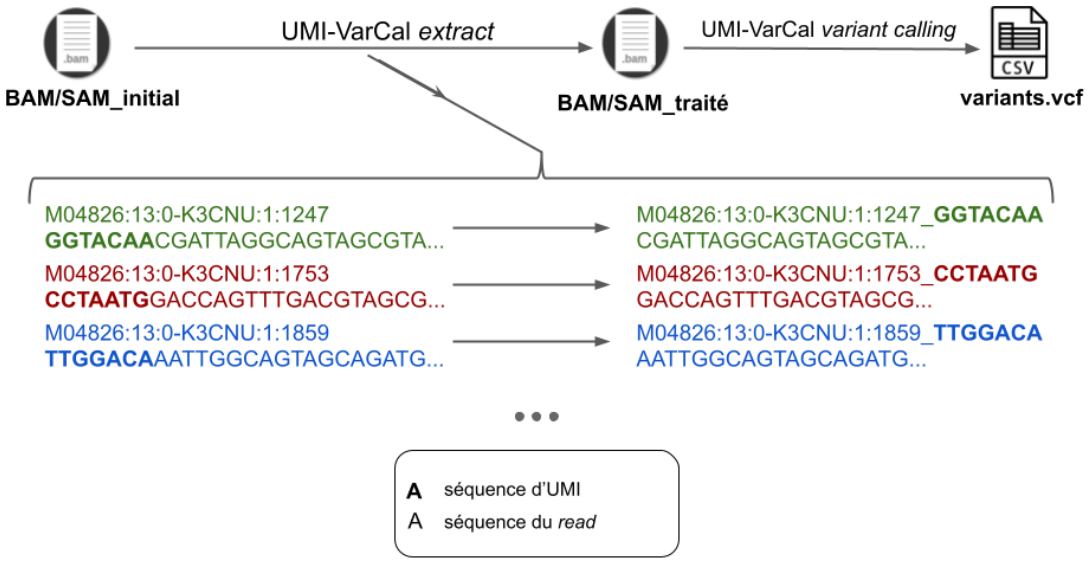


FIGURE 5.5 – La méthode d'extraction des UMI et les deux types de fichiers d'entrée acceptés par UMI-VarCal. Figure adaptée de [122]

5.5.2.3 Construction du *pileup*

La première étape du *workflow* du *variant calling* consiste à construire le *pileup*. Un *pileup* comprend le nombre total d'événements d'égalité, de substitution, d'insertion et de délétion à chaque position couverte par le fichier BED. En effet, après avoir filtré tous les *reads* ayant des scores de qualité faibles, UMI-VarCal parcourt l'ensemble des *reads* et compte le nombre de fois où chaque événement a été observé. Étant donné que chaque *read* est associé à un UMI, il est capable d'associer chaque événement observé sur ce *read* à l'UMI correspondant. Après avoir analysé tous les *reads*, cette étape génère une liste complète du nombre d'observations de A, C, G, T, d'insertion et de délétion ainsi que leurs UMI associés, à chaque position et pour chaque chromosome. Un exemple est donné dans la Table 5.1. C'est également à ce moment que notre algorithme estime le bruit de fond pour chaque position. Une fois la construction du *pileup* terminée, UMI-VarCal génère automatiquement un fichier PILEUP avec toutes les informations nécessaires. Ce fichier peut être utilisé si l'utilisateur souhaite lancer une nouvelle analyse du même échantillon mais avec des paramètres d'appel de variants différents, car la modification de ceux-ci n'affecte pas le *pileup* en soi mais uniquement les variants appelés. Par conséquent, UMI-VarCal pourra charger directement les informations du *pileup* à partir du fichier PILEUP sans avoir besoin de le reconstruire et permettant ainsi de compléter l'analyse plus rapidement.

5.5.2.4 Estimation du bruit de fond

Afin de faire la distinction entre les variants réels et les artefacts techniques (erreurs de PCR et de séquençage), le bruit de fond doit être estimé. Nous savons déjà que le taux d'erreur n'est pas uniforme et peut varier à des positions différentes, et donc nous pouvons supposer que chaque position a un taux d'erreur spécifique. Pour estimer les taux d'erreur par position, certains outils se servent d'un échantillon normal apparié en plus de l'échantillon tumoral pour effectuer l'analyse, tandis que d'autres utilisent de nombreux échantillons de contrôle pour modéliser le bruit de fond et le fournir au *variant*

Chromosome	Position	A	C	G	T	ins	del	liste d'UMI
chr1	27022890	6	4	1132	25	0	1	[ACGAGTA, ...]
chr1	27023171	16	2	988	99	10	2	[AGGTGTC, ...]
chr1	27100074	8	0	1251	0	5	4	[GAGAGTC, ...]
chr16	3779490	0	2	25	720	1	2	[TGTGCTC, ...]
chr16	3820929	1	952	39	2	0	0	[AACCTCT, ...]
chr6	138198219	3	42	992	0	0	1	[GAATATG, ...]
chr16	81953081	0	2	17	864	56	1	[TTCCAGA, ...]
chr16	11001770	13	4	799	99	2	5	[CCCATAG, ...]
chr19	19256791	1	1025	43	22	5	9	[AATTTCG, ...]
chr19	19261545	2	1067	8	8	15	7	[CGTATGC, ...]

TABLE 5.1 – Un exemple du *pileup* construit par UMI-VarCal. Dans cet exemple, dix positions seulement sont montrées alors qu'en réalité, les comptages sont réalisés pour chaque position du fichier BED.

caller sous forme intégrée. Sachant que la première approche est certainement la plus précise pour estimer le taux d'erreur, les échantillons normaux appariés sont très difficiles à obtenir, en particulier pour les échantillons de *cfDNA*. D'autre part, l'estimation du modèle sur un nombre d'échantillons de contrôle est une bonne approche qui est capable de filtrer de nombreux artefacts techniques mais présente la limitation d'être spécifique au protocole de séquençage et au kit utilisés. Par conséquent, le même modèle ne peut pas être utilisé sur différents protocoles et un modèle doit être établi pour chaque protocole et kit utilisés. C'est pourquoi UMI-VarCal utilise une stratégie différente basée sur les scores de qualité de base pour estimer les probabilités d'erreur relatives à chaque position.

Puisque chaque base séquencée est associée à un score de qualité, nous pouvons l'utiliser pour déterminer la probabilité d'erreur de base à chaque position en calculant le score de qualité moyen. En supposant que X_i représente le nombre total de *reads* découvrant la position i tel que $X_i = \{x_i^1, x_i^2, ..., x_i^n\}$, et que Q_i représente les scores de qualité des bases à la position i pour chaque *read* tel que $Q_i = \{q_i^1, q_i^2, ..., q_i^n\}$, nous pouvons facilement calculer le score de qualité moyen de la position i par la formule :

$$\overline{q_i} = \frac{\sum_{j=1}^{n} q_i^j}{n} \tag{5.4}$$

En effet, un score de qualité représente la probabilité que la base séquencée soit mal appelée. Ainsi, le score de qualité $\overline{q_i}$ calculé par la formule ci-dessus reflète la probabilité d'erreur de la base à la position i. À partir de ce score moyen, nous pouvons facilement calculer la probabilité d'erreur de base ϵ_i à chaque position par la formule :

$$\epsilon_i = 10^{\frac{-\overline{q_i}}{10}} \tag{5.5}$$

5.5.2.5 Recherche des variants potentiels

Le *pileup* généré contient les décomptes de A, C, G, T, insertions et délétions pour toutes les positions couvertes par le fichier BED. UMI-VarCal parcourt toutes ces positions et applique à chacune d'elles un test de Poisson pour déterminer si l'allèle alternatif peut être distingué du bruit de fond. Le test de Poisson représente la première étape de recherche des variants. Son application est très rapide et permet d'éliminer toutes les positions pour lesquelles un variant n'est pas présent. Cette première étape sert ainsi d'éviter d'appliquer une analyse des UMI complètement inutile sur des positions où un variant

potentiel n'est pas présent. L'utilisation d'une distribution de Poisson sur des données de séquençage présente l'inconvénient d'augmenter le taux d'erreur de première espèce α (taux de faux positifs) mais l'avantage de diminuer le taux d'erreur de deuxième espèce β (taux de faux négatifs). Ainsi, l'utilisation d'un modèle de Poisson dans un *variant caller* aura comme conséquence d'augmenter la sensibilité de l'outil mais de réduire sa spécificité. C'est exactement pour cette raison que nous avons utilisé ce modèle puisque notre objectif essentiel était de ne rater aucun vrai variant et donc réduire le nombre de faux négatifs. En ce qui concerne la spécificité de l'outil, même si le nombre de faux positifs sera potentiellement élevé suite à ce test, l'application d'une analyse des UMI par la suite est plus que suffisante pour filtrer les faux positifs, et par conséquent, augmenter la spécificité.

Nous avons supposé que la présence d'un variant est un événement rare. Il pourrait être traité comme un problème de test d'hypothèse statistique, où l'hypothèse nulle ($H0$) est que l'allèle alternatif (substitution, suppression et insertion) ne peut pas être distingué du bruit de fond et l'hypothèse alternative ($H1$) est que l'allèle alternatif est significativement supérieur au bruit de fond et pourrait, en effet, représenter une vraie mutation. À une position i, nous définissons d_i comme la profondeur à cette position, ϵ_i sa probabilité d'erreur de base et k_i étant le nombre total d'observations de l'allèle alternatif. Sous $(H0)$, k_i suit une distribution de Poisson (λ_i), tel que λ_i est le nombre d'erreurs attendues à la position i. On peut simplement calculer λ_i par l'équation :

$$\lambda_i = d_i.\epsilon_i \tag{5.6}$$

Par la suite, à une position i, nous pouvons calculer la *p-value* représentant la probabilité d'observer plus de λ_i erreurs par la formule :

$$p(k_i;\lambda_i) = 1 - \sum_{j=0}^{k_i} \frac{e^{-\lambda_i}.\lambda_i^j}{j!} \tag{5.7}$$

Lorsque nous effectuons plusieurs tests d'hypothèse, nous avons une probabilité accrue de faux positifs. Autrement dit, si nous effectuons le même test statistique plusieurs fois, les chances d'appeler un résultat nul comme significatif deviennent plus élevées. Le taux de faux positifs, ou FDR (*False Discovery Rate*) fait référence au nombre de faux positifs attendus lorsque nous effectuons un test d'hypothèse. Donc, si nous fixons la probabilité d'erreur de type 1 (alpha) à 0,05, nous pouvons nous assurer qu'au pire, le pourcentage de faux positifs dans tous les tests que nous avons effectués sera à 5%. Par exemple, si nous testons 10 000 positions et contrôlons l'alpha à 0,05 (5%), en moyenne 500 faux variants (10 000 $\times$ 0,05) seront appelés. Cette méthode pose un problème lorsque nous menons plusieurs tests car elle devient trop permissive et nous ne voulons pas avoir un si grand nombre de faux positifs. En règle générale, plusieurs procédures de tests d'hypothèse multiples contrôlent le FDR en essayant d'identifier les caractéristiques les plus significatives des tests et en essayant de filtrer autant de faux positifs que possible en même temps. UMI-VarCal applique la procédure de Benjamini-Hochberg [123] afin de diminuer le FDR, réduisant ainsi considérablement le nombre total de faux positifs. Après avoir appliqué la correction FDR aux *p-values*, nous obtenons les *q-values* correspondantes. Si la *q-value* est $\geq \alpha$, l'hypothèse nulle sera acceptée et donc la position sera filtrée car cela signifie que l'allèle alternatif observé à cette position n'est pas significativement différent du bruit de fond observé à cette position, et représente très probablement un artefact. Même avec la correction FDR, la modélisation de Poisson appliquée à cette situation maintient une sensibilité relativement élevée nous laissant avec un nombre non négligeable de faux positifs. Ceci est principalement dû au fait que le test ne prend pas

en compte le biais de brin ni le contexte environnant du variant. Par conséquent, afin de réduire le nombre de faux appels, nous appliquons trois procédures de post-traitement présentées ci-dessous.

5.5.2.6 Analyse des UMI

Lorsque le test de Poisson est effectué à chaque position, trois scénarios différents sont possibles :

1. aucun allèle alternatif n'est retrouvé et donc la position est filtrée.
2. Un allèle alternatif est retrouvé : la *q-value* du test de Poisson est $\geq \alpha$ ce qui signifie que la variation est due à une erreur (PCR/séquençage) et donc la position est également filtrée.
3. Un allèle alternatif est retrouvé : la *q-value* du test de Poisson est $< \alpha$ ce qui signifie que la variation est probablement due à une vraie mutation et donc la position est conservée.

Dans ce dernier cas uniquement, une analyse UMI est appliquée. Cette étape consiste principalement à séparer la liste de tous les UMI trouvés à une position en trois listes distinctes :

1. ref_umi : une liste de tous les UMI associés à des *reads* portant l'allèle de référence ;
2. alt_umi : une liste de tous les UMI associés à des *reads* portant l'allèle alternatif ;
3. noise_umi : une liste de tous les UMI associés à des *reads* ne portant ni l'allèle de référence ni l'allèle alternatif.

Théoriquement, si une variation est causée par la présence d'un vrai variant, elle devait se retrouver sur le fragment d'ADN initial. Ainsi, lorsque nous marquons le fragment d'ADN avec un UMI, nous marquons également le variant avec cet UMI. Après amplification, le fragment d'ADN est amplifié et produira des milliers de *reads*, tous portant le même UMI ainsi que le même allèle alternatif. Cela signifie qu'à une position spécifique, si l'allèle alternatif représente un vrai variant, tous les *reads* associés au même UMI doivent présenter le même allèle alternatif au même endroit. Si tel est le cas, l'UMI est appelé concordant. D'autre part, si certains *reads* d'un groupe UMI présentent l'allèle de référence ou un allèle de bruit, l'UMI est appelé discordant. La différence entre un UMI concordant et discordant est expliquée dans la Figure 5.6. Chaque étiquette UMI concordante caractérise un seul fragment d'ADN. En utilisant les trois listes ref_umi, alt_umi et noise_umi, nous pouvons calculer le nombre d'UMI concordants et discordants pour chaque variant. Le nombre d'UMI concordants d'un variant est un indicateur exact du nombre de molécules d'ADN initiales portant la mutation. UMI-VarCal utilise un seuil d'UMI concordants afin de filtrer les variants ayant peu de comptages UMI concordants. Ce filtre basé sur les UMI garantit que les variants qui passent à travers ne sont pas des artefacts techniques, et donc assure une très grande spécificité de détection. Ces variants sont appelés variants potentiels car ils n'ont pas encore franchi toutes les étapes de post-traitement.

5.5.2.7 Filtre de biais de brin

Il s'agit du deuxième filtre et il n'est applicable que pour les variants potentiels (variants qui ont réussi le test de Poisson et le processus d'analyse UMI). Il a été prouvé par Guo *et al.* qu'un biais de brin élevé (appelé SB pour *Strand Bias*) pourrait indiquer un taux potentiellement élevé de faux positifs, en particulier dans les données de séquençage à

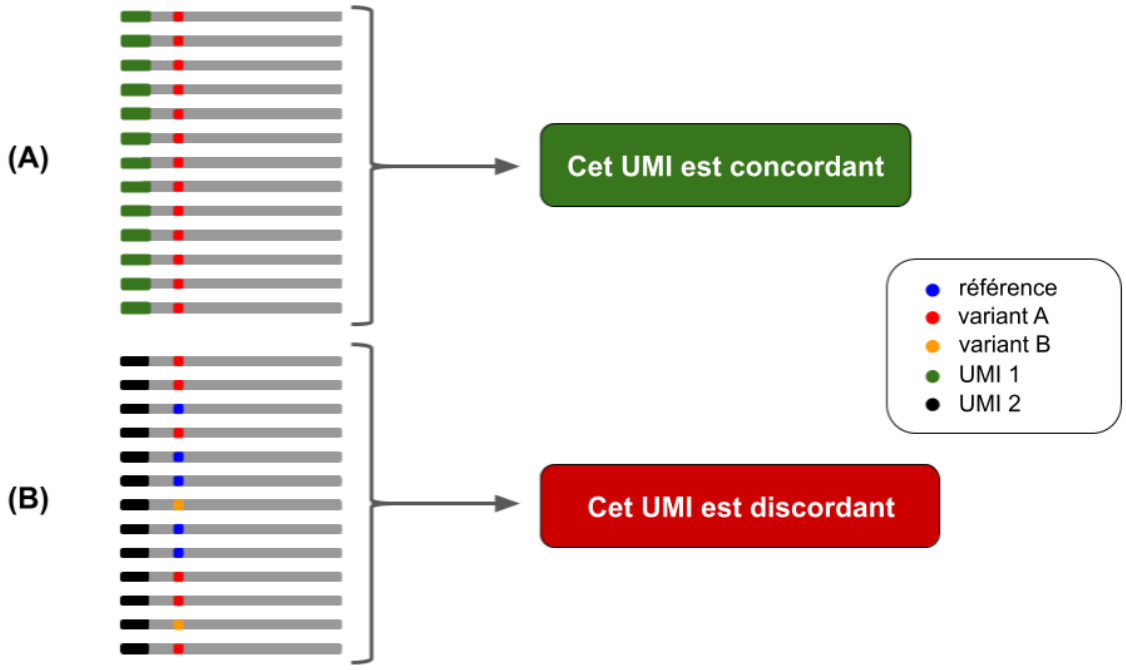


FIGURE 5.6 – La différence entre un UMI discordant et un UMI concordant. (A) Tous les *reads* associés à l'UMI 1 présentent le variant A : l'UMI 1 est concordant. (B) Le groupe UMI 2 comporte 13 *reads*. De ces 13 *reads*, six seulement présentent le variant A, cinq portent l'allèle de référence et deux présentent le variant B. Vu que tous les *reads* ne présentent pas le même allèle, nous concluons que l'UMI 2 est discordant.

reads courts d'Illumina [124]. Dans cette étape, notre filtre de biais de brin calcule un score pour chaque variant potentiel et, en le comparant à un seuil déterminé, vise à filtrer tous les faux positifs potentiels. Guo *et al.* ont comparé trois méthodes différentes pour calculer le score de biais de brin (le SB score traditionnel, le score GATK-SB et le score SB de Fisher) et ont démontré que le calcul SB traditionnel et le score Fisher permettent de mieux capturer les faux positifs que le score GATK-SB [125]. De plus, la méthode de calcul traditionnelle utilisée par Guo *et al.* pour détecter les faux positifs dans des analyses d'échantillons d'ADN mitochondrial a donné de très bons résultats avec un seuil de 1. UMI-VarCal utilise la méthode de calcul SB traditionnelle (Équation 5.8) et applique le seuil de 1 afin de filtrer le plus grand nombre de faux positifs parmi les variants potentiels sans être restrictif. Nous définissons R_f et R_r comme étant les comptes de l'allèle de référence observé sur les brins *forward* et *reverse*, et V_f et V_r comme étant les comptes de l'allèle alternatif observé sur les brins *forward* et *reverse* respectivement.

$$SB = \frac{\left| \frac{V_f}{R_f+V_f} - \frac{V_r}{R_r+V_r} \right|}{\frac{V_f+V_r}{R_f+R_r+V_f+V_r}} \tag{5.8}$$

5.5.2.8 Filtre sur les régions homopolymériques

Le pyroséquençage et le séquençage par conduction ionique (Ion Torrent) ont tous deux des difficultés à appeler correctement les bases situées dans de longues régions contenant des homopolymères. L'incertitude est due au fait que les nucléotides identiques répétitifs doivent être incorporés au cours du même cycle de synthèse. Même si la

chimie de séquençage base par base d'Illumina ne souffre pas autant dans les régions homopolymériques, en pratique, nous avons constaté que la précision du séquençage dans ces régions reste inférieure à celle des régions non homopolymériques. Ivády *et al.* ont démontré que la précision de l'appel de base chute considérablement lorsque la longueur de la région de l'homopolymère augmente (> 4 bases identiques) [126]. SomaticSniper [127] est un *variant caller* qui applique un filtre de longueur d'homopolymère afin d'éliminer les variants qui se produisent dans de longues régions homopolymériques, car de tels variants seraient très probablement des artefacts dus à des erreurs de séquençage. UMI-VarCal utilise le même filtre pour éliminer les variants trouvés dans une région homopolymérique ayant une longueur > 7.

5.5.2.9 Fichiers de sortie

Par défaut, UMI-VarCal produit automatiquement quatre fichiers :

1. un fichier VCF standard contenant tous les variants qui ont réussi les tests et qui ont été signalés avec succès. Pour chaque variant, la fréquence allélique, le nombre d'observations d'allèles alternatifs, la profondeur totale, la longueur de la région homopolymérique, le type de variant et la confiance sont fournis. Un niveau de confiance est fourni pour chaque variant et est calculé sur la base de son score de biais de brin, de la longueur de la région homopolymérique, de la *q-value* et du rapport UMI concordant/UMI discordant). Cinq niveaux sont possibles allant de faible à certain (faible < moyen < élevé < fort < certain).
2. Un fichier gVCF contenant tous les variants qui ont réussi les tests et qui ont été appelés par l'outil. En outre, le gVCF rapporte les blocs de positions pour lesquels aucun variant n'a été détecté. La profondeur moyenne et le score de qualité moyen des positions sont indiqués pour chaque bloc.
3. Un fichier VARIANTS contenant les mêmes variants du fichier VCF, en plus des métriques détaillées pour chaque variant.
4. Un fichier binaire PILEUP qui correspond au *pileup* construit par UMI-VarCal. Ce fichier peut être utilisé pour ignorer la reconstruction du *pileup* si l'analyse a déjà été effectuée sur l'échantillon.

5.5.2.10 *Workflow*

Le *workflow* global, présenté dans la Figure 5.7, est composé de modules Python qui sont appelés par un script Python principal. Tous les modules sont compilés en Cython pour obtenir de meilleures performances globales. UMI-VarCal est disponible pour Python version 2 et 3. UMI-VarCal est un logiciel autonome et ne repose sur aucun programme externe pour fonctionner. L'outil d'extraction et l'outil d'appel de variants sont exécutés via l'interface en ligne de commande UNIX/Linux. Tous les paramètres et seuils (qualité de base minimale, qualité de lecture minimale, qualité d'alignement minimale, taux d'erreur alpha de type 1, nombre minimal d'UMI concordants, biais maximal de brin et longueur maximale de la région homopolymérique) sont modifiables afin d'offrir à l'utilisateur un contrôle total sur ses résultats.

5.5.3 Résultats

À l'heure actuelle, trois *variant callers* basés sur les UMI sont disponibles publiquement : DeepSNVMiner, MAGERI et smCounter2. L'outil smCounter2 devrait être relativement rapide mais a une limite de détection théorique de seulement 0,5%. De plus, l'outil est un peu compliqué à utiliser : un des fichiers est de format inconnu et est nécessaire

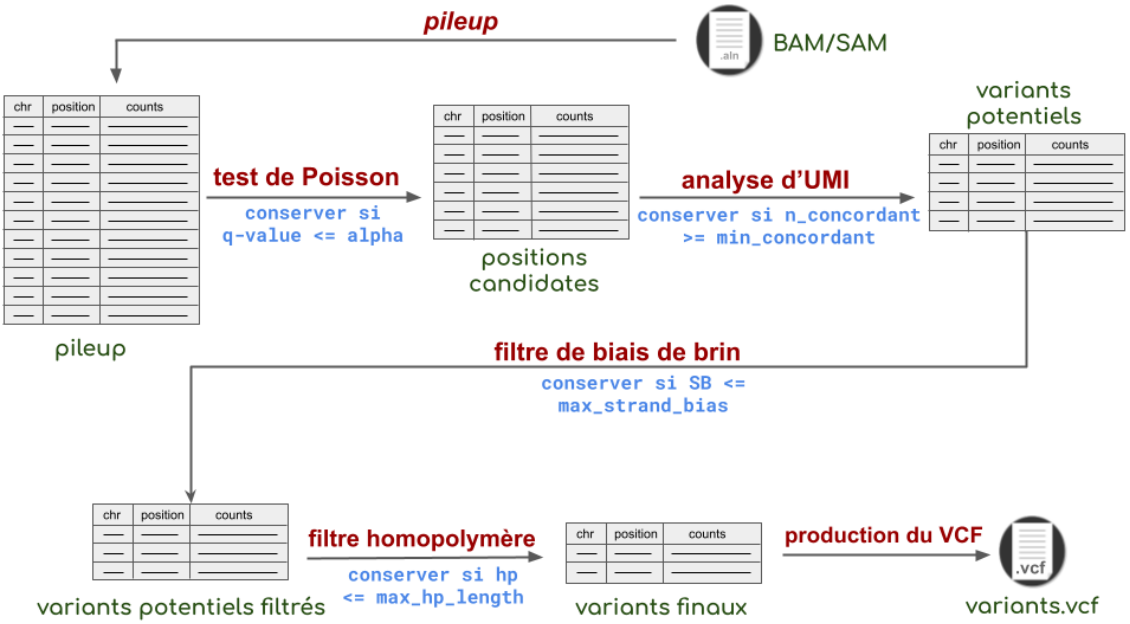


FIGURE 5.7 – Le *workflow* de l'outil de *variant calling* d'UMI-VarCal. Figure adaptée de [122].

pour lancer le programme. Nous avons essayé de contacter le développeur mais n'avons pas reçu de réponse. MAGERI a une limite de détection théorique de 0,1% mais il est très lent et consomme beaucoup de mémoire (dans nos tests, il a fallu 1 heure de temps d'exécution et un minimum de 200 Go de RAM pour analyser un échantillon). Enfin, DeepSNVMiner présente l'avantage d'avoir la même limite de détection que MAGERI (0,1%) et est beaucoup plus efficace en termes de temps d'exécution et de consommation mémoire. Pour démontrer la supériorité d'UMI-VarCal par rapport aux autres outils disponibles en termes de détection de variants ainsi que de performance, nous l'avons comparé à DeepSNVMiner et également à deux des meilleurs *variant callers* classiques, outLyzer et SiNVICT, spécialement conçus pour détecter les variants à de très faible fréquence. Dans ce qui suit, nous comparerons les performances de détection des quatre outils sur trois échantillons différents représentant des données réelles ainsi que deux échantillons simulés.

5.5.3.1 Données réelles

Le Centre Henri Becquerel de Rouen (France) a conçu un panel de séquençage ciblé pour l'analyse du lymphome diffus à grandes cellules B (DLBCL), destiné à identifier les anomalies génomiques au sein d'une liste de 36 gènes les plus fréquemment impactés dans ce type de lymphome (Table 5.2). Le panel a été spécialement conçu pour introduire des UMI lors de la construction de la librairie. Afin de tester UMI-VarCal contre les trois différents outils DeepSNVMiner, SiNVICT et outLyzer, nous avons sélectionné au hasard trois échantillons d'un très grand nombre de patients dont l'ADN a été séquencé au Centre Henri Becquerel et tous souffrant de DLBCL. L'échantillon X et l'échantillon Z sont des biopsies congelées extraites de deux patients différents du CHB tandis que l'échantillon Y est un ADN extrait d'une lignée cellulaire. Les échantillons sélectionnés ont fait l'objet d'un examen histopathologique approprié et la qualité de l'ADN a été jugée adéquate pour le séquençage.

Gène	Nombre de régions	Gène	Nombre de régions
ARID1A	85	GNA13	14
B2M	6	ID3	6
BCL2	8	IRF4	22
BRAF	2	MEF2B	21
BTK	3	MYC	16
CARD11	16	MYD88	8
CCND3	15	NOTCH1	19
CD58	17	NOTCH2	24
CD79A	5	PIM1	14
CD79B	4	PLCG2	16
CDKN2A	27	PRDM1	33
CDKN2B	47	SOCS1	8
CIITA	66	STAT6	14
CREBBP	103	TCF3	5
CXCR4	12	TNFAIP3	31
EP300	102	TNFRSF14	16
EZH2	5	TP53	24
FOXP1	22	XPO1	11

TABLE 5.2 – La liste des gènes ciblés et le nombre de régions par gène du panel Pan-lymphome du CHB.

Échantillon X

Au total, 464 variants ont été trouvés (tous les variants sont détaillés dans la Table 5.3) (Figure 5.8A). UMI-VarCal détecte 145 variants, tandis que DeepSNVMiner, outLyzer et SiNVICT en ont détecté respectivement 257, 144 et 63. Parmi ces 145 variants, 29 ont été trouvés par les trois autres outils et 63 ont été trouvés par au moins un autre *variant caller*. 214 variants n'ont été trouvés que par DeepSNVMiner : 139/214 n'ont pas réussi le test de Poisson, 60/214 n'ont pas réussi le test d'analyse UMI, 4/214 sont probablement des artefacts dus au biais de brin et 1/214 est dans une longue région homopolymérique.
75 variants ont été trouvés uniquement par outLyzer : 3/75 n'ont pas réussi le test de Poisson, 64/75 n'ont pas réussi le test d'analyse UMI, 3/75 sont probablement des artefacts dus au biais de brin et 5/75 sont dans une longue région homopolymérique.
Quinze variants ont été trouvés uniquement par SiNVICT : 3/15 n'ont pas réussi le test de Poisson, 7/15 n'ont pas réussi le test d'analyse UMI et 5/15 sont détectés dans des positions qui ne sont pas couvertes par le fichier BED fourni.
Dix variants ont été trouvés à la fois par SiNVICT et outLyzer : les 10 variants sont dans une longue région homopolymérique. Quatre variants ont été détectés à la fois par DeepSNVMiner et outLyzer : les quatre variants n'ont pas réussi le test d'analyse UMI et l'un d'eux présente un fort biais de brin. Un seul variant a été trouvé par DeepSNVMiner, SiNVICT et outLyzer : ce variant se trouve dans une zone d'homopolymère très longue (longueur = 16). 82 variants ont été détectés uniquement par UMI-VarCal : 74/82 (90,2%) ont une fréquence inférieure à 1% et 28/82 (34,1%) ont une fréquence inférieure à 0,5%. De plus, seulement 1/82 (1,2%) avait un faible niveau de confiance tandis que 73/82 (89%) avaient au moins un niveau de confiance élevé.
Enfin, nous avons lancé UMI-VarCal sans les filtres de biais de brin ni celui de longueur de la région homopolymérique afin de voir si la plupart des discordances est due à l'application de ces deux filtres (Figure 5.8B). Au total, 24/314 (7,6%) discordances sont causées soit par le filtre d'homopolymère, soit par le filtre de biais de brin, ce qui signifie

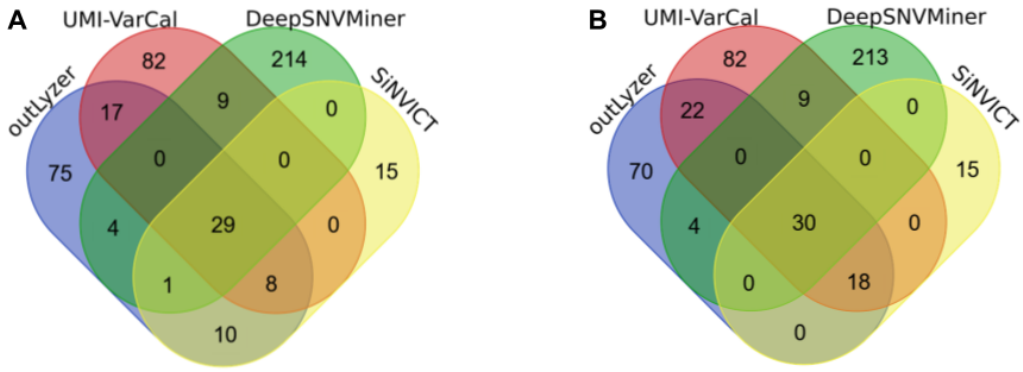


FIGURE 5.8 – **(A)** Un diagramme de Venn représentant les variants détectés par UMI-VarCal, DeepSNVMiner, SiNVICT et outLyzer dans l'échantillon X. **(B)** Un diagramme de Venn représentant les variants détectés par UMI-VarCal (sans le filtre de biais de brin ni le filtre des régions homopolymériques), DeepSNVMiner, SiNVICT et outLyzer dans l'échantillon X.

que 92,4% des discordances sont causées par une *q-value* trop élevée ou par un échec au niveau de l'analyse des UMI (le nombre de discordances par type de discordance pour l'échantillon X se trouvent dans la Table 5.4).

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
UMI-VarCal & DeepSNVMiner & SiNVICT & outLyzer (29)									
chr13	41134707	A	G	45,29	0,0000	259	0,076	3	confiance = forte (4/5)
chr16	10971186	C	T	51,14	0,0000	353	0,021	1	confiance = certaine (5/5)
chr16	10989291	C	T	54,2	0,0000	192	0,036	3	confiance = forte (4/5)
chr16	11001691	C	T	49,24	0,0000	482	0,019	1	confiance = certaine (5/5)
chr16	11001694	T	C	50,34	0,0000	481	0,008	1	confiance = certaine (5/5)
chr16	11001770	G	T	50,14	0,0000	463	0,016	2	confiance = certaine (5/5)
chr16	11001914	G	A	47,9	0,0000	405	0,005	3	confiance = forte (4/5)
chr16	11004150	C	T	46,91	0,0000	246	0,077	1	confiance = certaine (5/5)
chr16	11009587	C	A	49,68	0,0000	182	0,032	2	confiance = certaine (5/5)
chr17	7579472	G	C	99,93	0,0000	883	0,001	6	confiance = forte (4/5)
chr18	60985384	G	A	3,88	0,0000	33	0,08	1	confiance = certaine (5/5)
chr18	60985743	G	T	15,94	0,0000	120	0,066	3	confiance = forte (4/5)
chr18	60985755	A	G	22,37	0,0000	170	0,076	2	confiance = certaine (5/5)
chr18	60985879	T	C	42,74	0,0000	297	0,076	1	confiance = certaine (5/5)
chr19	42384948	C	T	49,33	0,0000	411	0,014	1	confiance = certaine (5/5)
chr1	120458924	G	A	44,44	0,0000	282	0,08	1	confiance = certaine (5/5)
chr1	23885498	T	C	99,82	0,0000	695	0	1	confiance = certaine (5/5)
chr22	41537234	G	T	99,19	0,0000	531	0,001	2	confiance = certaine (5/5)
chr22	41551039	T	A	46,08	0,0000	206	0,045	3	confiance = forte (4/5)
chr22	41564708	C	A	52,79	0,0000	292	0,007	2	confiance = certaine (5/5)
chr22	41568480	T	C	49,81	0,0000	214	0,001	1	confiance = certaine (5/5)
chr3	38182136	C	G	50,5	0,0000	335	0,041	2	confiance = certaine (5/5)
chr6	106555025	G	A	67,18	0,0000	304	0,036	2	confiance = certaine (5/5)
chr6	138197331	A	C	99,57	0,0000	327	0	1	confiance = certaine (5/5)
chr8	128751201	G	A	49,82	0,0000	528	0,008	1	confiance = certaine (5/5)
chr9	139391636	G	A	52,39	0,0000	443	0,001	1	confiance = certaine (5/5)
chr9	21975017	C	T	49,05	0,0000	407	0,01	2	confiance = certaine (5/5)
chr9	22003367	G	A	49,37	0,0000	468	0,073	2	confiance = certaine (5/5)
chr9	22005330	T	G	50,44	0,0000	443	0,022	1	confiance = certaine (5/5)
UMI-VarCal & SiNVICT & outLyzer (8)									
chr17	7579644	C	-	61,38	0,0000	180	0,057	6	confiance = forte (4/5)
chr17	7579801	G	C	99,87	0,0000	283	0,001	5	confiance = forte (4/5)
chr18	60985900	C	G	3,87	0,0000	23	0,269	2	confiance = certaine (5/5)
chr18	60985901	C	T	3,81	0,0000	22	0,292	2	confiance = forte (4/5)
chr1	2493172	G	A	44,05	0,0000	198	0,004	2	confiance = certaine (5/5)
chr1	27100182	G	-	3,9	0,0000	13	0,131	1	confiance = forte (4/5)
chr7	148506396	A	C	95,72	0,0000	305	0,005	1	confiance = certaine (5/5)
chr9	22005112	T	-	24,76	0,0000	60	0,166	1	confiance = certaine (5/5)

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
DeepSNVMiner & SiNVICT & outLyzer (1)									
chr17	7577679	T	-	42,02	0,0000	207	0,066	16	région homopolymérique (>7)
UMI-VarCal & DeepSNVMiner (9)									
chr16	10995933	A	G	99,6	0,0000	681	0	1	confiance = certaine (5/5)
chr16	11000848	G	C	99,95	0,0000	608	0	2	confiance = certaine (5/5)
chr16	11002927	A	G	99,65	0,0000	778	0,001	1	confiance = certaine (5/5)
chr16	81953081	T	C	99,91	0,0000	645	0,001	1	confiance = certaine (5/5)
chr18	60985864	C	A	13,09	0,0000	91	0,036	3	confiance = forte (4/5)
chr6	398965	C	T	47,44	0,0000	306	0,053	2	confiance = certaine (5/5)
chr7	148508833	A	G	46,07	0,0000	285	0,087	1	confiance = certaine (5/5)
chr9	21968159	G	A	51,45	0,0000	162	0,054	2	confiance = certaine (5/5)
chr9	21968199	C	G	96,39	0,0000	278	0,012	3	confiance = forte (4/5)
UMI-VarCal & outLyzer (17)									
chr12	57499106	T	C	1,17	0,0000	7	0,428	2	confiance = élevée (3/5)
chr16	3778401	T	-	1,38	0,0000	8	0,292	1	confiance = forte (4/5)
chr16	3786734	T	-	0,71	0,0000	7	0,159	6	confiance = moyenne (2/5)
chr16	3786780	A	G	0,89	0,0000	5	0,885	1	confiance = moyenne (2/5)
chr16	3820955	A	G	0,54	0,0001	5	0,404	1	confiance = élevée (3/5)
chr17	62006785	T	C	0,66	0,0000	5	0,459	1	confiance = élevée (3/5)
chr17	63052587	T	C	0,73	0,0000	5	0,854	4	confiance = élevée (3/5)
chr17	7572963	T	-	1,05	0,0000	6	0,108	6	confiance = moyenne (2/5)
chr18	60985888	A	T	1,26	0,0000	7	0,063	1	confiance = certaine (5/5)
chr18	60985907	A	T	1,64	0,0000	7	0,109	1	confiance = certaine (5/5)
chr22	41553309	G	A	0,92	0,0000	5	0,711	2	confiance = élevée (3/5)
chr22	41565643	T	-	1,15	0,0000	6	0,015	7	confiance = moyenne (2/5)
chr2	136873129	A	G	0,55	0,0000	5	0,349	1	confiance = élevée (3/5)
chr6	138192447	A	G	0,69	0,0002	5	0,427	1	confiance = élevée (3/5)
chr6	394931	A	G	0,81	0,0000	5	0,213	2	confiance = élevée (3/5)
chr9	21974396	A	G	0,7	0,0000	5	0,414	1	confiance = élevée (3/5)
chr9	21974564	A	G	0,71	0,0000	9	0,217	2	confiance = élevée (3/5)
DeepSNVMiner & outLyzer (4)									
chr16	3781808	T	C	0,47	0,0002	2	0,715	2	peu d'UMI concordants
chr1	2493239	A	G	0,88	0,0000	2	1,178	4	peu d'UMI concordants
chr22	41554396	A	G	0,74	0,0000	4	0,085	2	peu d'UMI concordants
chr6	106554898	T	C	0,83	0,0000	1	0,113	1	peu d'UMI concordants
SiNVICT & outLyzer (10)									
chr16	3808053	A	-	23,79	0,0000	136	0,039	13	région homopolymérique (>7)
chr16	3828848	T	-	15,31	0,0000	30	0,18	10	région homopolymérique (>7)
chr16	81954790	T	-	7,87	0,0000	60	0,181	9	région homopolymérique (>7)
chr1	117057449	A	-	8,4	0,0000	29	0,028	9	région homopolymérique (>7)
chr1	117087236	A	-	4,8	0,0000	17	0,038	9	région homopolymérique (>7)
chr22	41545025	T	-	32,63	0,0000	120	0,037	14	région homopolymérique (>7)
chr22	41572224	A	-	4,68	0,0000	14	0,123	8	région homopolymérique (>7)
chr6	106534485	T	-	18,54	0,0000	93	0,081	12	région homopolymérique (>7)
chr9	22003299	A	-	3,41	0,0000	6	0,229	8	région homopolymérique (>7)
chr9	22003879	T	-	3,72	0,0000	10	0,219	8	région homopolymérique (>7)
SiNVICT (15)									
chr16	3808052	T	TA	13,1	0,0000	0	0,052	13	aucun UMI concordant
chr16	3828847	C	CT	3,8	0,0000	0	0,509	10	aucun UMI concordant
chr16	81954789	C	CT	2,67	0,0000	0	0,118	9	aucun UMI concordant
chr17	7577678	C	CT	7,52	0,0000	0	0,209	16	aucun UMI concordant
chr19	19257114	A	C	1,05	0,0640	0	1,373	1	q-value > 0,05 ; aucun UMI concordant
chr19	19257143	T	G	3,12	0,0000	0	2,97	1	aucun UMI concordant
chr6	106534484	A	AT	9,86	0,0000	0	0,144	12	aucun UMI concordant
chr9	21971141	C	G	0,77	1,2000	0	0,793	1	q-value > 0,05
chr9	21971161	T	G	0,26	0,8641	0	1,846	1	q-value > 0,05
chr9	21971164	A	G	2,51	0,0000	1	0,375	1	peu d'UMI concordants
chr17	79786722	G	C	-	-	-	-	-	position hors fichier BED
chr19	29482011	G	A	-	-	-	-	-	position hors fichier BED
chr3	197024800	C	T	-	-	-	-	-	position hors fichier BED
chr6	394785	T	-	-	-	-	-	-	position hors fichier BED
chr7	158185187	T	C	-	-	-	-	-	position hors fichier BED
outLyzer (75)									
chr12	57496197	A	G	0,78	0,0000	3	0,135	1	peu d'UMI concordants
chr12	57498299	A	G	0,42	0,0007	4	0,059	1	peu d'UMI concordants
chr15	45008499	T	-	3	0,0000	9	0,199	8	région homopolymérique (>7)
chr16	10971214	T	C	0,55	0,0000	1	0,071	1	peu d'UMI concordants
chr16	10989212	T	C	0,58	0,0000	2	0,037	1	peu d'UMI concordants
chr16	10989241	A	G	0,56	0,0022	1	0,314	1	peu d'UMI concordants
chr16	10989266	A	G	0,99	0,0000	2	0,31	2	peu d'UMI concordants
chr16	10996532	T	C	0,89	0,0000	4	0,067	1	peu d'UMI concordants
chr16	10998594	T	C	0,58	0,0007	1	0,379	1	peu d'UMI concordants
chr16	11000824	A	G	0,73	0,0023	2	0,916	1	peu d'UMI concordants
chr16	11016047	A	G	0,8	0,0000	1	0,382	1	peu d'UMI concordants
chr16	11017132	A	G	0,43	0,0007	4	0,32	2	peu d'UMI concordants
chr16	11017138	A	G	0,53	0,0002	2	0,349	1	peu d'UMI concordants
chr16	11017155	A	G	0,46	0,0233	3	0,358	1	peu d'UMI concordants
chr16	11017178	A	G	0,81	0,0000	3	0,682	1	peu d'UMI concordants
chr16	3778303	G	-	0,97	0,0019	2	0,048	1	peu d'UMI concordants
chr16	3779136	T	-	1,24	0,0000	2	0,062	1	peu d'UMI concordants
chr16	3779868	T	C	0,77	0,0002	2	0,87	1	peu d'UMI concordants

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
chr16	3781805	A	G	0,83	0,0000	3	1,884	1	peu d'UMI concordants
chr16	3786761	A	G	0,54	0,0000	4	0,376	2	peu d'UMI concordants
chr16	3794913	T	C	0,68	0,0000	4	0,586	4	peu d'UMI concordants
chr16	3799608	A	G	0,72	0,0002	3	0,552	6	peu d'UMI concordants
chr16	3819234	T	C	0,75	0,0001	4	0,265	1	peu d'UMI concordants
chr16	3828046	A	G	0,66	0,0002	2	0,384	1	peu d'UMI concordants
chr16	3830839	T	C	0,56	0,0061	1	0,434	2	peu d'UMI concordants
chr16	3831189	T	C	0,74	0,0000	3	0,652	1	peu d'UMI concordants
chr16	3832867	T	C	0,66	0,0000	1	0,303	2	peu d'UMI concordants
chr16	81942079	A	G	0,69	0,0002	1	1,469	2	peu d'UMI concordants
chr16	81942092	C	T	0,54	1,2000	2	0,663	2	q-value > 0,05
chr16	81942109	A	G	0,62	0,0114	2	1,566	2	peu d'UMI concordants
chr16	81946256	T	C	0,62	0,0002	2	1,238	2	peu d'UMI concordants
chr17	63010470	T	C	0,65	0,0000	3	1,216	1	peu d'UMI concordants
chr17	63049859	A	G	1,17	0,0000	2	0,076	5	peu d'UMI concordants
chr17	63049864	A	-	2,58	0,0000	8	0,138	8	région homopolymérique (>7)
chr17	63052468	C	-	2,37	0,0000	2	0,316	1	peu d'UMI concordants
chr1	117078855	A	-	2,39	0,0000	8	0,222	8	région homopolymérique (>7)
chr1	120458572	T	C	0,81	0,0000	8	1,178	1	biais de brin
chr1	2489216	T	C	0,63	0,0000	3	0,206	1	peu d'UMI concordants
chr1	2491340	T	C	0,55	0,0003	1	0,437	1	peu d'UMI concordants
chr1	2492093	A	G	0,49	0,0037	1	0,037	2	peu d'UMI concordants
chr1	27022977	A	-	1,01	0,0000	4	0,418	1	peu d'UMI concordants
chr1	27057727	C	-	1	0,0000	4	0,301	1	peu d'UMI concordants
chr1	27058068	T	C	0,68	0,0000	3	0,603	2	peu d'UMI concordants
chr1	27087525	T	C	0,5	0,0390	0	0,652	2	aucun UMI concordant
chr1	27099415	T	C	0,75	0,0000	3	0,198	1	peu d'UMI concordants
chr22	41521969	A	G	0,62	0,0001	4	0,212	5	peu d'UMI concordants
chr22	41522068	T	C	1,22	0,0000	3	0,327	2	peu d'UMI concordants
chr22	41523691	T	C	0,83	0,0000	3	0,423	1	peu d'UMI concordants
chr22	41527584	A	G	0,53	0,0000	3	0,016	1	peu d'UMI concordants
chr22	41527606	T	C	0,68	0,0000	4	0,378	1	peu d'UMI concordants
chr22	41574038	A	T	0,78	0,0000	4	0,251	1	peu d'UMI concordants
chr22	41574044	A	AT	0,72	0,0006	0	0,117	1	aucun UMI concordant
chr2	136872415	A	-	2,3	0,0000	8	0,215	8	région homopolymérique (>7)
chr2	61717832	T	C	0,84	0,0000	3	0,13	4	peu d'UMI concordants
chr2	61717855	T	C	0,65	0,0000	3	0,094	1	peu d'UMI concordants
chr3	38181953	C	T	0,53	0,1774	3	0,17	2	q-value > 0,05
chr6	106546968	A	G	0,81	0,0000	3	0,402	4	peu d'UMI concordants
chr6	106547029	T	C	0,85	0,0000	4	0,365	2	peu d'UMI concordants
chr6	138196050	C	T	1,69	0,0000	8	1,661	1	biais de brin
chr6	138198275	T	C	0,66	0,0000	1	0,617	5	peu d'UMI concordants
chr6	393334	A	G	1,12	0,0000	3	1,424	1	peu d'UMI concordants
chr6	395871	A	G	0,77	0,0000	2	0,237	1	peu d'UMI concordants
chr6	395882	T	C	0,5	0,0066	2	0,249	1	peu d'UMI concordants
chr6	395945	A	G	0,81	0,0000	3	0,281	1	peu d'UMI concordants
chr6	41905108	T	C	0,52	0,0001	2	0,424	2	peu d'UMI concordants
chr6	41908237	T	C	0,97	0,0000	6	1,914	3	biais de brin
chr7	148506473	A	G	0,81	0,0001	4	0,353	1	peu d'UMI concordants
chr8	128748845	T	-	0,73	0,0001	1	0,261	7	peu d'UMI concordants
chr8	128750605	C	-	1,3	0,0000	4	0,697	2	peu d'UMI concordants
chr9	139390945	G	-	1,65	0,0000	2	0,119	1	peu d'UMI concordants
chr9	139391844	T	C	0,49	0,3255	3	0,852	1	q-value > 0,05
chr9	22003169	A	G	0,6	0,0000	4	1,508	1	peu d'UMI concordants
chr9	22003277	A	-	1,92	0,0000	11	0,11	8	région homopolymérique (>7)
chr9	22003749	T	TG	1,84	0,0000	0	0,261	8	aucun UMI concordant
chrX	100611050	T	C	1,03	0,0000	3	0,271	1	peu d'UMI concordants
DeepSNVMiner (214)									
chr12	57496129	T	C	0,35	0,0502	0	0,041	1	q-value > 0,05 ; aucun UMI concordant
chr12	57496749	T	-	10,88	0,0000	14	0,498	21	région homopolymérique (>7)
chr12	57496774	T	C	0,94	0,0000	0	2,131	41	aucun UMI concordant
chr12	57498288	T	C	0,22	0,1720	0	0,116	1	q-value > 0,05
chr12	57498323	A	T	0,34	0,0146	3	0,064	1	peu d'UMI concordants
chr12	57498359	T	C	0,35	0,0502	0	0,611	2	q-value > 0,05 ; aucun UMI concordant
chr13	41133901	C	T	0,22	0,9995	1	0,265	6	q-value > 0,05
chr13	41134148	G	A	0,47	0,0004	2	0,143	2	peu d'UMI concordants
chr13	41240330	A	C	0,4	1,2000	0	1,517	1	q-value > 0,05
chr15	45007657	C	A	0,25	0,1471	0	0,272	1	q-value > 0,05
chr16	10997674	A	G	1,15	0,0000	5	1,353	1	biais de brin
chr16	10998626	C	T	0,15	1,2000	1	0,335	3	q-value > 0,05
chr16	10998643	T	C	0,9	0,0000	3	0,561	1	peu d'UMI concordants
chr16	10998666	C	T	0,38	0,0911	0	0,204	2	q-value > 0,05 ; aucun UMI concordant
chr16	10998700	T	G	0,24	0,4185	0	0,933	1	q-value > 0,05
chr16	11000381	A	T	0,15	1,2000	0	1,312	1	q-value > 0,05
chr16	11001137	G	A	0,27	1,2000	3	0,41	1	q-value > 0,05
chr16	11001690	T	C	0,46	0,3223	1	0,786	1	q-value > 0,05
chr16	11009532	T	C	0,5	0,0061	0	0,441	2	aucun UMI concordant
chr16	11010207	T	C	0,46	0,0061	1	0,338	1	peu d'UMI concordants
chr16	3777950	C	A	0,31	0,6109	1	0,476	2	q-value > 0,05
chr16	3778281	A	G	0,24	0,9529	0	0,43	1	q-value > 0,05
chr16	3778318	T	G	0,06	1,2000	0	1,381	1	q-value > 0,05
chr16	3778322	C	G	0,17	1,2000	0	2,07	2	q-value > 0,05
chr16	3778330	A	G	0,17	1,2000	1	1,178	1	q-value > 0,05
chr16	3778353	T	G	0,77	0,0000	0	2,832	1	aucun UMI concordant
chr16	3778356	A	G	0,5	0,0185	2	0,688	2	peu d'UMI concordants
chr16	3778363	C	G	0,06	1,2000	0	1,87	2	q-value > 0,05
chr16	3778575	T	C	0,69	0,0000	1	0,533	1	peu d'UMI concordants
chr16	3778951	G	A	0,31	0,2414	1	1,423	2	q-value > 0,05
chr16	3779901	C	T	0,31	0,9679	1	0,98	2	q-value > 0,05
chr16	3781920	T	C	0,27	0,0911	0	0,129	2	q-value > 0,05 ; aucun UMI concordant
chr16	3786743	C	T	0,22	0,6109	0	0,83	1	q-value > 0,05
chr16	3789727	T	C	0,25	0,3223	1	0,274	1	q-value > 0,05
chr16	3799677	A	AAA	45,72	0,0000	0	0,005	3	aucun UMI concordant
chr16	3807953	C	T	0,56	0,0233	3	0,15	1	peu d'UMI concordants
chr16	3817819	T	C	0,48	0,0001	4	0,041	4	peu d'UMI concordants

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
chr16	3817833	C	T	0,53	0,0067	2	0,426	2	peu d'UMI concordants
chr16	3820840	G	A	0,23	0,0911	2	0,225	1	q-value > 0,05 ; peu d'UMI concordants
chr16	3823753	T	C	0,51	0,0061	0	0,477	1	aucun UMI concordant
chr16	3828813	T	C	0,21	0,2414	0	0,456	2	q-value > 0,05
chr16	3830788	T	C	0,17	0,7163	0	0,387	3	q-value > 0,05
chr16	3832828	T	C	0,7	0,0000	0	0,086	1	aucun UMI concordant
chr16	3843668	G	A	0,25	0,3037	2	0,327	1	q-value > 0,05
chr16	3860782	T	A	0,13	0,6109	1	0,435	1	q-value > 0,05
chr16	81944303	C	T	0,2	1,2000	1	0,184	4	q-value > 0,05
chr16	81954777	A	T	0,28	0,1353	1	1,763	1	q-value > 0,05
chr16	81954795	T	C	0,35	0,4259	2	0,384	9	q-value > 0,05
chr16	81954819	C	T	0,28	0,3255	2	1,301	3	q-value > 0,05
chr16	81954836	A	G	0,2	0,4070	0	0,406	1	q-value > 0,05
chr17	62006788	C	T	0,34	0,0979	1	0,738	2	q-value > 0,05 ; peu d'UMI concordants
chr17	63052593	C	T	0,15	1,2000	1	0,465	3	q-value > 0,05
chr17	7572980	T	G	9,96	0,0000	1	3,321	1	peu d'UMI concordants
chr17	7574011	T	A	0,5	0,0006	2	1,053	1	peu d'UMI concordants
chr17	7574069	A	G	0,3	0,8641	0	1,471	1	q-value > 0,05
chr17	7576504	T	C	0,93	0,0000	3	0,282	2	peu d'UMI concordants
chr17	7576892	T	A	0,31	0,0067	2	0,248	3	peu d'UMI concordants
chr17	7577595	C	T	0,19	0,6486	1	1,36	1	q-value > 0,05
chr17	7577704	T	C	0,36	0,9091	0	0,116	1	q-value > 0,05
chr17	7578588	C	T	0,26	0,1471	1	0,776	1	q-value > 0,05
chr17	7579339	A	T	0,21	0,3223	0	0,141	1	q-value > 0,05
chr18	60985417	G	A	0,35	0,9410	2	0,253	1	q-value > 0,05
chr18	60985710	C	T	0,22	1,2000	1	0,36	3	q-value > 0,05
chr19	1612356	C	T	0,36	0,5369	2	0,283	3	q-value > 0,05
chr19	1612358	C	T	0,73	0,0328	1	0,297	3	peu d'UMI concordants
chr19	19256859	G	A	0,41	0,4259	3	1,059	1	q-value > 0,05
chr19	19256888	A	G	0,05	1,2000	0	0,536	1	q-value > 0,05
chr19	19260187	T	C	0,91	0,0023	0	1,57	1	aucun UMI concordant
chr19	19260229	T	G	1,26	0,0000	1	1,987	1	peu d'UMI concordants
chr1	120458230	G	A	0,22	0,8641	1	1,183	2	q-value > 0,05
chr1	120458515	C	T	0,34	0,6109	1	1,34	3	q-value > 0,05
chr1	120458717	A	G	0,16	0,9679	0	0,909	3	q-value > 0,05
chr1	120458803	T	G	0,39	0,4070	0	1,702	2	q-value > 0,05
chr1	120459040	T	C	0,31	0,1203	2	0,913	2	q-value > 0,05
chr1	120464862	A	G	0,6	0,0000	1	0,031	1	peu d'UMI concordants
chr1	120465276	G	A	0,45	0,2414	0	0,38	3	q-value > 0,05
chr1	120466558	C	T	0,32	0,6109	2	0,093	4	q-value > 0,05
chr1	23885491	T	C	0,3	0,1203	0	0,27	2	q-value > 0,05
chr1	23885775	T	G	3,1	0,0000	3	1,712	1	peu d'UMI concordants
chr1	23885784	T	G	9,84	0,0000	14	1,306	2	biais de brin
chr1	23885793	T	C	0,45	0,8440	1	0,411	1	q-value > 0,05
chr1	2492067	C	T	0,73	0,0502	1	0,582	2	q-value > 0,05 ; peu d'UMI concordants
chr1	2494588	G	A	0,64	0,2414	2	0,072	2	q-value > 0,05
chr1	27022995	A	G	0,53	0,0911	1	0,053	1	q-value > 0,05 ; peu d'UMI concordants
chr1	27023469	T	C	0,64	0,0911	1	0,911	1	q-value > 0,05 ; peu d'UMI concordants
chr1	27023639	T	A	0,4	0,4259	1	1,523	1	q-value > 0,05
chr1	27023768	A	C	0,22	1,2000	0	1,875	1	q-value > 0,05
chr1	27057691	C	T	0,39	0,0502	1	0,377	2	q-value > 0,05 ; peu d'UMI concordants
chr1	27057734	A	G	0,32	0,2414	0	0,03	1	q-value > 0,05
chr1	27087479	A	C	1,6	0,0000	0	4,2	1	aucun UMI concordant
chr1	27087489	A	C	4,61	0,0000	0	3,801	1	aucun UMI concordant
chr1	27087512	T	C	2,59	0,0000	0	2,871	2	aucun UMI concordant
chr1	27087518	T	C	3	0,0000	1	1,588	1	peu d'UMI concordants
chr1	27087844	A	G	0,6	0,0006	0	0,745	1	aucun UMI concordant
chr1	27093048	C	T	0,26	0,4259	1	0,853	2	q-value > 0,05
chr1	27099919	T	C	0,54	0,5420	0	0,647	1	q-value > 0,05
chr1	27099999	C	T	0,27	0,7958	1	0,027	4	q-value > 0,05
chr1	27101483	G	A	0,15	0,9679	2	0,727	1	q-value > 0,05
chr1	27101603	C	T	0,31	0,7958	1	1,102	4	q-value > 0,05
chr1	27101632	T	C	0,62	0,0390	1	0,832	1	peu d'UMI concordants
chr1	27101645	C	T	0,39	0,1353	1	0,665	2	q-value > 0,05
chr1	27102089	T	G	10,65	0,0000	10	1,461	1	biais de brin
chr1	27102207	C	T	0,49	0,0001	2	1,24	1	peu d'UMI concordants
chr1	27105804	G	A	0,29	0,1720	1	0,17	1	q-value > 0,05
chr1	27106161	G	A	0,17	0,9529	0	0,749	2	q-value > 0,05
chr1	27106334	T	A	0,24	1,2000	0	0,008	1	q-value > 0,05
chr1	27106985	T	C	0,39	0,1720	0	0,778	1	q-value > 0,05
chr1	27107114	C	T	0,32	0,5369	2	0,62	2	q-value > 0,05
chr22	41513609	G	A	0,34	0,8897	0	0,354	3	q-value > 0,05
chr22	41521870	G	A	0,78	0,0004	1	0,205	3	peu d'UMI concordants
chr22	41523556	T	C	0,65	0,0016	1	0,42	2	peu d'UMI concordants
chr22	41531833	A	G	0,59	0,0023	2	0,198	1	peu d'UMI concordants
chr22	41545891	C	T	0,39	0,0640	2	0,436	2	q-value > 0,05 ; peu d'UMI concordants
chr22	41545950	A	C	1,55	0,0000	1	2,46	1	peu d'UMI concordants
chr22	41545971	T	C	0,69	0,0000	2	1,73	1	peu d'UMI concordants
chr22	41546012	A	C	0,07	1,2000	0	1,737	1	q-value > 0,05
chr22	41546038	A	G	0,2	0,9091	0	0,032	1	q-value > 0,05
chr22	41546053	A	G	0,36	0,0636	0	0,235	2	q-value > 0,05 ; aucun UMI concordant
chr22	41546066	C	T	0,22	1,2000	1	0,727	4	q-value > 0,05
chr22	41547866	A	C	0,16	0,9529	0	0,907	1	q-value > 0,05
chr22	41558792	G	A	0,47	0,3223	1	0,365	1	q-value > 0,05
chr22	41564604	G	A	0,57	0,0502	2	0,352	2	q-value > 0,05 ; peu d'UMI concordants
chr22	41564730	T	C	0,44	0,0911	1	0,191	1	q-value > 0,05 ; peu d'UMI concordants
chr22	41572380	A	G	0,41	0,0067	2	0,937	2	peu d'UMI concordants
chr22	41572485	G	A	0,25	0,1203	2	0,491	1	q-value > 0,05
chr22	41573048	G	A	0,27	0,9091	1	0,398	3	q-value > 0,05
chr22	41573121	G	A	0,69	0,3255	2	0,196	2	q-value > 0,05
chr22	41573203	A	G	0,42	0,0502	2	0,307	1	q-value > 0,05 ; peu d'UMI concordants
chr22	41573242	C	T	0,38	0,1203	1	1,065	1	q-value > 0,05
chr22	41573549	T	C	0,64	0,0000	0	1,21	1	aucun UMI concordant
chr22	41574029	G	T	0,62	0,0004	1	0,141	2	peu d'UMI concordants
chr22	41574177	C	T	0,48	0,0979	1	0,432	2	q-value > 0,05 ; peu d'UMI concordants
chr22	41574771	A	G	0,26	0,6486	0	0,174	1	q-value > 0,05
chr22	41574776	C	T	0,19	1,2000	0	1,12	3	q-value > 0,05

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
chr22	41574874	G	A	0,2	0,4259	1	0,83	2	q-value > 0,05
chr2	61717870	T	C	0,21	0,7163	1	0,189	1	q-value > 0,05
chr2	61719248	C	T	0,39	0,0979	1	1,159	2	q-value > 0,05 ; biais de brin
chr6	106536055	G	A	6,47	0,0000	1	1,734	2	peu d'UMI concordants
chr6	106536109	T	G	2,85	0,0000	0	1,723	3	aucun UMI concordant
chr6	106536117	A	G	1,07	0,0000	0	1,578	1	aucun UMI concordant
chr6	106536222	C	T	0,44	0,8098	3	0,184	1	q-value > 0,05
chr6	106547214	T	C	0,53	0,0022	2	0,182	3	peu d'UMI concordants
chr6	106552709	A	G	0,25	0,3223	0	1,095	1	q-value > 0,05
chr6	106552804	G	A	0,58	0,0022	1	0,077	2	peu d'UMI concordants
chr6	106553621	T	G	3,29	0,0000	0	2,561	1	aucun UMI concordant
chr6	106554835	G	A	0,42	0,9091	1	0,48	1	q-value > 0,05
chr6	106555172	C	T	0,37	0,0440	1	0,546	1	peu d'UMI concordants
chr6	138196068	A	G	0,33	0,0328	4	0,259	1	peu d'UMI concordants
chr6	138198246	G	A	0,53	0,6109	1	0,884	4	q-value > 0,05
chr6	138200010	G	A	0,42	0,0061	1	0,498	1	peu d'UMI concordants
chr6	138200291	T	C	0,46	0,0911	2	0,153	1	q-value > 0,05 ; peu d'UMI concordants
chr6	138200338	G	A	0,38	0,5369	1	1,203	1	q-value > 0,05
chr6	138200419	G	A	0,56	0,0233	1	0,242	1	peu d'UMI concordants
chr6	138202251	A	G	0,61	0,0000	2	0,269	2	peu d'UMI concordants
chr6	138202297	T	C	0,35	0,0543	0	0,301	1	q-value > 0,05 ; aucun UMI concordant
chr6	138202311	T	G	0,06	1,2000	0	1,175	1	q-value > 0,05
chr6	138202318	T	G	0,25	0,3223	0	2,019	1	q-value > 0,05
chr6	138202326	A	C	0,05	1,2000	0	0,638	1	q-value > 0,05
chr6	138202350	A	C	1,69	0,0056	0	1,355	1	aucun UMI concordant
chr6	138202362	A	G	1,89	0,0000	1	1,118	1	peu d'UMI concordants
chr6	37138614	G	A	0,37	1,2000	0	0,548	2	q-value > 0,05
chr6	37138716	A	G	0,34	0,6109	2	0,028	1	q-value > 0,05
chr6	37140833	G	A	0,79	0,0669	2	0,691	2	q-value > 0,05 ; peu d'UMI concordants
chr6	393157	A	G	1	0,0000	0	1,354	2	aucun UMI concordant
chr6	395908	A	G	0,3	0,1203	1	0,187	2	q-value > 0,05
chr6	397039	C	-	14,83	0,0000	61	1,399	1	biais de brin
chr6	397040	T	C	0,33	0,4259	0	0,107	1	q-value > 0,05
chr6	397044	G	A	0,32	0,4259	0	1,606	1	q-value > 0,05
chr6	397073	T	C	0,37	0,0185	0	1,963	1	aucun UMI concordant
chr6	397079	T	C	0,73	0,0000	2	1,05	2	peu d'UMI concordants
chr6	401492	G	A	0,23	1,2000	1	0,269	1	q-value > 0,05
chr6	401699	G	A	0,13	1,2000	0	0,393	2	q-value > 0,05
chr6	405054	C	T	0,36	0,0146	1	0,207	2	peu d'UMI concordants
chr6	405120	T	C	0,36	0,0979	3	0,006	1	q-value > 0,05 ; peu d'UMI concordants
chr6	407554	G	A	1,07	0,0000	0	1,874	1	aucun UMI concordant
chr6	41904334	A	G	0,49	0,0067	1	0,223	1	peu d'UMI concordants
chr6	41908136	T	G	0,49	0,5420	0	2,964	1	q-value > 0,05
chr6	41908290	A	G	0,46	0,0185	1	0,405	2	peu d'UMI concordants
chr6	41908737	T	C	0,52	0,0022	0	0,501	3	aucun UMI concordant
chr7	148508781	C	T	0,12	1,2000	0	0,918	2	q-value > 0,05
chr7	2977656	T	C	0,3	0,3223	0	0,153	1	q-value > 0,05
chr7	2979560	G	A	0,48	0,3037	1	0,02	1	q-value > 0,05
chr7	2983893	C	T	0,5	0,0006	1	0,435	1	peu d'UMI concordants
chr7	2983911	G	A	0,49	0,0502	3	0,863	1	q-value > 0,05 ; peu d'UMI concordants
chr8	128750521	G	A	0,23	1,2000	1	0,836	1	q-value > 0,05
chr8	128750568	G	A	0,38	0,0183	2	0,225	1	peu d'UMI concordants
chr8	128750680	A	C	2,58	0,0000	2	3,581	1	peu d'UMI concordants
chr8	128751197	C	-	0,68	0,0000	3	1,474	4	peu d'UMI concordants
chr9	139391006	C	T	0,21	0,4259	1	0,482	1	q-value > 0,05
chr9	139391383	G	A	0,36	0,4259	2	1,078	3	q-value > 0,05
chr9	139391583	A	G	0,34	0,1203	0	0,774	1	q-value > 0,05
chr9	139391598	G	A	0,18	1,2000	1	0,358	1	q-value > 0,05
chr9	139391621	C	T	0,17	0,7958	1	0,838	1	q-value > 0,05
chr9	139392026	G	A	1,11	0,0001	2	0,402	5	peu d'UMI concordants
chr9	21967932	G	A	0,51	0,6109	1	0,477	1	q-value > 0,05
chr9	21971161	T	C	0,95	0,0000	0	4,675	1	aucun UMI concordant
chr9	21974967	C	-	0,19	0,9455	0	1,086	3	q-value > 0,05
chr9	22003073	T	C	0,1	1,2000	0	1,055	1	q-value > 0,05
chr9	22003100	A	G	0,47	0,0002	2	0,547	3	peu d'UMI concordants
chr9	22003204	C	T	0,34	0,0328	1	1,064	1	peu d'UMI concordants
chr9	22003369	T	C	0,27	0,2414	0	0,957	2	q-value > 0,05
chr9	22004190	T	C	0,42	0,0185	1	0,905	2	peu d'UMI concordants
chr9	22004667	T	C	0,43	0,0502	0	0,539	3	q-value > 0,05 ; aucun UMI concordant
chr9	22004986	G	A	0,13	0,7163	0	0,436	1	q-value > 0,05
chr9	22005067	A	G	0,29	0,1353	0	0,458	4	q-value > 0,05
chr9	22005892	G	A	0,17	1,2000	3	0,462	1	q-value > 0,05
chr9	22005923	T	G	0,99	0,0102	1	1,003	1	peu d'UMI concordants
chr9	22006130	G	A	0,22	0,6109	1	1,626	1	q-value > 0,05
chr9	22006154	C	T	0,21	1,2000	2	1,524	2	q-value > 0,05
chr9	22009042	G	A	0,21	1,2000	0	0,385	2	q-value > 0,05
chr9	22009191	A	T	0,11	1,2000	1	0,211	1	q-value > 0,05
chr9	22009214	A	G	0,36	1,2000	1	0,687	1	q-value > 0,05
chr9	22009252	A	G	0,45	0,0920	2	0,666	1	q-value > 0,05 ; peu d'UMI concordants
chr9	22009264	C	G	0,17	1,2000	0	1,664	1	q-value > 0,05
UMI-VarCal (82)									
chr12	57496620	A	G	0,56	0,0002	5	0,236	1	confiance = forte (4/5)
chr12	57498291	C	T	0,35	0,0023	5	0,029	1	confiance = forte (4/5)
chr13	41133778	A	G	0,95	0,0000	5	0,743	1	confiance = élevée (3/5)
chr13	41134050	T	C	1	0,0000	6	0,462	1	confiance = élevée (3/5)
chr13	41134196	T	C	0,71	0,0000	5	0,585	1	confiance = élevée (3/5)
chr13	41239862	A	G	0,68	0,0000	5	0,061	1	confiance = forte (4/5)
chr16	10996523	T	C	0,6	0,0023	5	0,675	1	confiance = moyenne (2/5)
chr16	10996561	A	G	0,53	0,0019	5	0,286	1	confiance = élevée (3/5)
chr16	11002869	A	G	0,47	0,0007	5	0,137	1	confiance = élevée (3/5)
chr16	11002912	T	C	0,36	0,0146	5	0,584	1	confiance = élevée (3/5)
chr16	11003993	G	A	10,46	0,0000	10	0,034	3	confiance = élevée (3/5)
chr16	11010267	T	C	0,57	0,0002	7	0,331	1	confiance = élevée (3/5)
chr16	3789684	C	T	0,6	0,0061	5	0,942	1	confiance = faible (1/5)
chr16	3827573	A	G	0,83	0,0000	6	0,848	2	confiance = moyenne (2/5)

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
chr16	3831250	T	C	0,68	0,0000	5	0,558	3	confiance = forte (4/5)
chr16	3841984	T	C	0,59	0,0001	5	0,06	3	confiance = forte (4/5)
chr16	3843521	A	G	0,55	0,0002	5	0,822	1	confiance = moyenne (2/5)
chr16	3843604	T	C	0,5	0,0000	5	0,51	2	confiance = élevée (3/5)
chr16	3843617	T	C	0,3	0,0114	5	0,313	3	confiance = élevée (3/5)
chr16	81960712	A	G	0,44	0,0002	5	0,412	1	confiance = forte (4/5)
chr16	81960719	A	G	0,3	0,0146	5	0,081	1	confiance = élevée (3/5)
chr16	81960751	T	C	0,37	0,0050	5	0,039	1	confiance = élevée (3/5)
chr16	81960802	T	C	0,7	0,0000	5	0,11	1	confiance = élevée (3/5)
chr17	62006731	A	G	0,72	0,0000	5	0,214	1	confiance = élevée (3/5)
chr17	63014409	T	C	0,52	0,0000	6	0,351	4	confiance = élevée (3/5)
chr17	63049652	T	C	0,63	0,0001	5	0,758	2	confiance = élevée (3/5)
chr17	7576857	A	G	0,89	0,0000	5	0,995	2	confiance = moyenne (2/5)
chr17	7579924	G	A	0,76	0,0061	5	0,151	2	confiance = élevée (3/5)
chr18	60985612	G	A	3,28	0,0000	13	0,146	3	confiance = forte (4/5)
chr18	60985613	G	A	3,23	0,0000	13	0,061	3	confiance = forte (4/5)
chr18	60985695	T	C	1,25	0,0000	5	0,513	1	confiance = forte (4/5)
chr18	60985850	T	C	0,55	0,0001	5	0,209	2	confiance = élevée (3/5)
chr19	19257424	A	G	1	0,0000	6	0,054	1	confiance = élevée (3/5)
chr1	117087212	A	G	0,75	0,0000	6	0,053	5	confiance = élevée (3/5)
chr1	120458646	T	C	0,48	0,0000	5	0,146	1	confiance = élevée (3/5)
chr1	120458649	A	G	0,34	0,0328	5	0,254	1	confiance = moyenne (2/5)
chr1	120458651	A	G	0,65	0,0000	5	0,334	1	confiance = élevée (3/5)
chr1	120465303	T	C	0,46	0,0328	5	0,019	1	confiance = élevée (3/5)
chr1	2488094	A	G	0,49	0,0001	5	0,026	1	confiance = élevée (3/5)
chr1	2489219	T	C	0,43	0,0002	6	0,183	1	confiance = forte (4/5)
chr1	2492113	T	C	0,48	0,0001	5	0,238	1	confiance = élevée (3/5)
chr1	27087428	A	G	0,4	0,0023	6	0,049	1	confiance = forte (4/5)
chr1	27087562	A	G	0,74	0,0000	6	0,147	1	confiance = élevée (3/5)
chr1	27088680	T	C	0,87	0,0000	5	0,011	2	confiance = élevée (3/5)
chr1	27089527	T	A	0,54	0,0002	5	0,083	1	confiance = élevée (3/5)
chr1	27092793	T	C	0,37	0,0022	5	0,202	1	confiance = forte (4/5)
chr1	27094354	A	G	0,52	0,0001	5	0,371	2	confiance = élevée (3/5)
chr1	27094366	A	G	0,49	0,0037	5	0,465	1	confiance = élevée (3/5)
chr1	27097604	A	G	0,65	0,0183	5	0,808	1	confiance = moyenne (2/5)
chr1	27100839	A	G	0,68	0,0002	5	0,653	1	confiance = forte (4/5)
chr1	27101036	A	G	0,39	0,0007	6	0,005	1	confiance = forte (4/5)
chr1	27102147	C	T	0,55	0,0002	5	0,74	1	confiance = élevée (3/5)
chr1	27106270	A	G	0,47	0,0002	5	0,187	1	confiance = forte (4/5)
chr22	41513295	A	G	0,53	0,0001	5	0,209	2	confiance = élevée (3/5)
chr22	41523664	A	G	0,56	0,0022	5	0,337	2	confiance = élevée (3/5)
chr22	41547927	A	G	0,44	0,0002	5	0,09	2	confiance = élevée (3/5)
chr22	41551072	C	T	1,06	0,0000	8	0,822	3	confiance = élevée (3/5)
chr22	41556695	T	C	0,48	0,0022	5	0,448	3	confiance = forte (4/5)
chr22	41565554	A	G	0,46	0,0001	5	0,685	3	confiance = élevée (3/5)
chr22	41572245	T	C	0,67	0,0001	5	0,109	5	confiance = élevée (3/5)
chr2	136873052	T	C	0,77	0,0000	5	0,663	2	confiance = élevée (3/5)
chr2	136873113	A	G	0,63	0,0001	5	0,561	2	confiance = forte (4/5)
chr2	61719522	A	G	0,44	0,0061	6	0,315	6	confiance = élevée (3/5)
chr6	138192407	A	T	2,33	0,0000	5	0,772	1	confiance = forte (4/5)
chr6	138197203	T	C	0,47	0,0067	5	0,168	1	confiance = forte (4/5)
chr6	394914	A	G	0,65	0,0000	5	0,266	1	confiance = élevée (3/5)
chr6	41903786	T	C	0,66	0,0002	6	0,005	2	confiance = élevée (3/5)
chr7	140453111	A	G	0,45	0,0002	5	0,121	1	confiance = élevée (3/5)
chr7	140453116	A	G	0,6	0,0001	5	0,329	1	confiance = élevée (3/5)
chr7	140453122	T	C	0,47	0,0007	6	0,046	1	confiance = élevée (3/5)
chr7	148508789	G	A	0,6	0,0264	5	0,238	1	confiance = moyenne (2/5)
chr7	2977606	T	C	0,63	0,0000	5	0,069	3	confiance = élevée (3/5)
chr7	2984053	C	T	0,48	0,0006	5	0,573	2	confiance = forte (4/5)
chr8	128752771	T	C	0,88	0,0000	6	0,386	1	confiance = élevée (3/5)
chr8	128753099	C	T	0,82	0,0039	6	0,495	2	confiance = élevée (3/5)
chr9	139391835	A	G	0,78	0,0000	5	0,199	1	confiance = élevée (3/5)
chr9	21994288	A	T	0,82	0,0001	5	0,836	1	confiance = élevée (3/5)
chr9	22003150	A	G	0,71	0,0000	5	0,021	3	confiance = forte (4/5)
chr9	22003819	A	G	0,4	0,0061	5	0,049	2	confiance = élevée (3/5)
chr9	22003910	T	C	0,52	0,0006	5	0,43	1	confiance = élevée (3/5)
chr9	22005860	A	G	0,41	0,0002	5	0,961	1	confiance = moyenne (2/5)
chr9	22006231	T	C	0,7	0,0000	5	0,274	1	confiance = élevée (3/5)

TABLE 5.3 – Liste détaillée des variants détectés par UMI-VarCal, DeepSNVMiner, SiNVICT et outLyzer dans l'échantillon X.

q-value	UMI	UMI HP	*q-value* & UMI	UMI & SB	SB	*q-value* & SB	*q-value* & HP	UMI & HP	SB & HP	Total
120	122	17	24	23	7	1	0	0	0	314
38,22%	38,85%	5,41%	7,64%	7,32%	2,23%	0,32%	0,00%	0,00%	0,00%	

TABLE 5.4 – Nombre de discordances classées par type pour l'échantillon X.

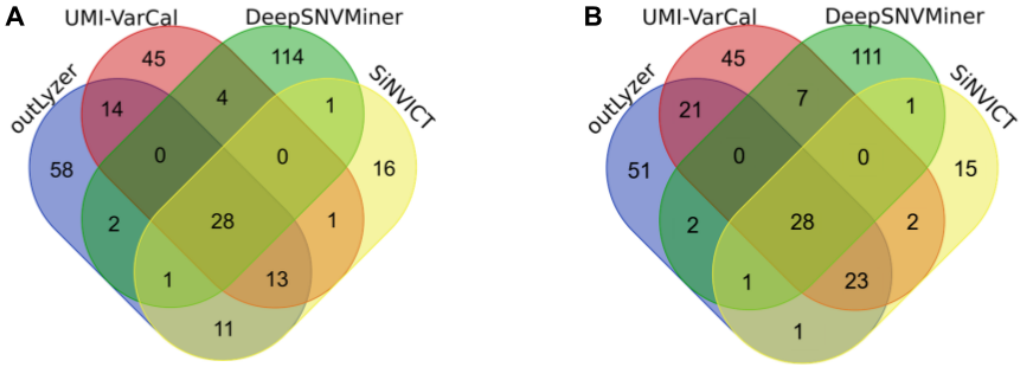


FIGURE 5.9 – **(A)** Un diagramme de Venn représentant les variants détectés par UMI-VarCal, DeepSNVMiner, SiNVICT et outLyzer dans l'échantillon Y. **(B)** Un diagramme de Venn représentant les variants détectés par UMI-VarCal (sans le filtre de biais de brin ni le filtre des régions homopolymériques), DeepSNVMiner, SiNVICT et outLyzer dans l'échantillon Y.

Échantillon Y

En ce qui concerne l'échantillon Y, 308 variants ont été retrouvés au total (tous les variants sont détaillés dans la Table 5.5) (Figure 5.9A). UMI-VarCal a détecté 105 variants, tandis que DeepSNVMiner, outLyzer et SiNVICT ont détecté 150, 127 et 71 variants respectivement. Parmi ces 105 variants, 28 sont également trouvés par les trois autres outils et 60 ont été trouvés par au moins un autre *variant caller*.
114 variants n'ont été trouvés que par DeepSNVMiner : 63/114 n'ont pas réussi le test de Poisson, 48/114 n'ont pas réussi le test d'analyse des UMI et 3/114 sont probablement des artefacts dus au biais de brin.
58 variants ont été trouvés uniquement par outLyzer : 5/58 n'ont pas réussi le test de Poisson, 46/58 n'ont pas réussi l'analyse des UMI, 2/58 sont probablement des artefacts dus au biais de brin et 5/58 sont dans une longue région homopolymérique.
Seize variants ont été trouvés uniquement par SiNVICT : 2/16 n'ont pas réussi le test de Poisson, 7/16 n'ont pas réussi le test d'analyse des UMI, 1/16 est dans une longue région homopolymérique et 6/16 sont détectés dans des positions non couvertes par le fichier BED fourni.
Onze variants ont été trouvés à la fois par SiNVICT et outLyzer : 10/11 variants sont dans une longue région homopolymérique et le dernier n'a pas d'UMI concordants et n'a donc pas réussi le test d'analyse UMI.
Deux variants ont été détectés à la fois par DeepSNVMiner et outLyzer : les deux n'ont pas réussi le test d'analyse UMI et l'un d'eux présente un biais de brin élevé.
Un seul variant a été trouvé par DeepSNVMiner, SiNVICT et outLyzer : ce variant est dans une longue région homopolymérique (longueur = 8) et n'a aucun UMI concordant.
45 variants ont été détectés uniquement par UMI-VarCal : 39/45 (86,7%) ont une fréquence inférieure à 1% et 13/45 (28,9%) ont une fréquence inférieure à 0,5%. De plus, aucun des 45 variants n'avait un faible niveau de confiance tandis que 37/45 (82,2%) avaient au moins un niveau de confiance élevé.
Enfin, pour cet échantillon également, nous avons lancé UMI-VarCal sans le filtre de biais de brin ni le filtre de longueur d'homopolymère (Figure 5.9B). Au total, 21/197 (10,7%) discordances sont causées soit par le filtre d'homopolymère, soit par le filtre de biais de brin, ce qui signifie que 89,3% des discordances sont causées par une *q-value* trop élevée

ou par un nombre d'UMI concordants insuffisants (le nombre de discordances par type de discordance pour l'échantillon Y se trouvent dans la Table 5.6).

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
UMI-VarCal & DeepSNVMiner & SiNVICT & outLyzer (28)									
chr16	10995933	A	G	99.7	0.0000	686	0.002	1	confiance = certaine (5/5)
chr16	11000848	G	C	49.43	0.0000	270	0.007	2	confiance = certaine (5/5)
chr16	11001770	G	T	51.72	0.0000	377	0.027	2	confiance = certaine (5/5)
chr16	11002904	G	A	47.05	0.0000	444	0.129	2	confiance = certaine (5/5)
chr16	11002927	A	G	99.95	0.0000	600	0	1	confiance = certaine (5/5)
chr16	11004150	C	T	49.35	0.0000	238	0.058	1	confiance = certaine (5/5)
chr16	11009541	C	T	56	0.0000	196	0.003	1	confiance = certaine (5/5)
chr16	81954789	C	G	47.41	0.0000	423	0.013	1	confiance = certaine (5/5)
chr17	7578518	C	G	99.85	0.0000	592	0	2	confiance = certaine (5/5)
chr17	7579472	G	C	99.83	0.0000	736	0	6	confiance = forte (4/5)
chr1	23885498	T	C	99.67	0.0000	602	0.001	1	confiance = certaine (5/5)
chr1	23885599	T	C	48.68	0.0000	182	0.01	1	confiance = certaine (5/5)
chr1	2488153	A	G	50.78	0.0000	545	0.004	4	confiance = forte (4/5)
chr1	2491205	C	T	47.16	0.0000	138	0.002	3	confiance = forte (4/5)
chr22	41537234	G	T	99.33	0.0000	437	0	2	confiance = certaine (5/5)
chr22	41551039	T	A	49.18	0.0000	180	0.058	3	confiance = forte (4/5)
chr22	41565575	A	G	48.82	0.0000	480	0	1	confiance = certaine (5/5)
chr22	41565578	A	C	48.54	0.0000	490	0	1	confiance = certaine (5/5)
chr22	41568480	T	C	46.5	0.0000	165	0	1	confiance = certaine (5/5)
chr6	106547372	C	G	50.91	0.0000	508	0.021	1	confiance = certaine (5/5)
chr6	138197331	A	C	51.87	0.0000	145	0.022	1	confiance = certaine (5/5)
chr6	401509	C	T	49.7	0.0000	464	0.028	3	confiance = forte (4/5)
chr9	139391321	G	A	49.07	0.0000	465	0.014	1	confiance = certaine (5/5)
chr9	139391636	G	A	50.95	0.0000	341	0.008	1	confiance = certaine (5/5)
chr9	21975017	C	T	48	0.0000	257	0.017	2	confiance = certaine (5/5)
chr9	22003223	C	T	48.17	0.0000	326	0.011	1	confiance = certaine (5/5)
chr9	22003367	G	A	99.55	0.0000	589	0.003	2	confiance = certaine (5/5)
chr9	22005330	T	G	47.43	0.0000	287	0.033	1	confiance = certaine (5/5)
UMI-VarCal & SiNVICT & outLyzer (13)									
chr17	7579644	C	-	57.03	0.0000	120	0.116	6	confiance = forte (4/5)
chr17	7579652	C	-	61.47	0.0000	129	0.161	3	confiance = forte (4/5)
chr17	7579801	G	C	99.26	0.0000	288	0.001	5	confiance = forte (4/5)
chr18	60985721	G	A	63.12	0.0000	137	0.023	2	confiance = certaine (5/5)
chr18	60985723	T	C	63.29	0.0000	137	0.017	1	confiance = certaine (5/5)
chr18	60985750	G	A	64.23	0.0000	284	0.016	2	confiance = certaine (5/5)
chr18	60985809	C	G	61.09	0.0000	249	0.075	3	confiance = forte (4/5)
chr18	60985833	G	A	61.52	0.0000	250	0.018	1	confiance = certaine (5/5)
chr18	60985901	C	G	30.87	0.0000	34	0.059	2	confiance = certaine (5/5)
chr18	60985911	G	T	30.99	0.0000	32	0.082	2	confiance = certaine (5/5)
chr19	19256870	G	A	52.53	0.0000	139	0.019	1	confiance = certaine (5/5)
chr1	27100182	G	-	3.26	0.0000	7	0.081	1	confiance = forte (4/5)
chr22	41548243	G	T	34	0.0000	61	0.484	1	confiance = certaine (5/5)
DeepSNVMiner & SiNVICT & outLyzer (1)									
chr9	22003298	C	CA	39.13	0.0000	0	0.108	8	aucun UMI concordant
UMI-VarCal & DeepSNVMiner (4)									
chr16	81953081	T	C	99.58	0.0000	600	0.002	1	confiance = certaine (5/5)
chr7	148508833	A	G	98.75	0.0000	409	0.013	1	confiance = certaine (5/5)
chr7	2977606	T	C	0.41	0.0061	5	0.229	3	confiance = moyenne (2/5)
chr9	21968199	C	G	95.94	0.0000	171	0.003	3	confiance = forte (4/5)
UMI-VarCal & SiNVICT (1)									
chr9	22005112	T	-	13.25	0.0000	19	0.085	1	confiance = forte (4/5)
UMI-VarCal & outLyzer (14)									
chr12	57499084	T	C	0.63	0.0000	5	0.329	2	confiance = élevée (3/5)
chr12	57499094	G	-	0.87	0.0000	5	0.363	7	confiance = moyenne (2/5)
chr16	3786734	T	-	1.03	0.0000	6	0.108	6	confiance = moyenne (2/5)
chr16	3817721	T	-	1.44	0.0000	5	0.044	7	confiance = moyenne (2/5)
chr16	3900932	T	C	0.69	0.0000	5	0.187	3	confiance = élevée (3/5)
chr16	81946284	A	G	0.5	0.0046	5	0.09	2	confiance = élevée (3/5)
chr17	7579638	C	T	1.41	0.0000	6	0.966	6	confiance = moyenne (2/5)
chr1	23885778	C	G	1.89	0.0000	14	0.463	1	confiance = certaine (5/5)
chr1	27022940	C	-	2.67	0.0000	6	0.105	3	confiance = élevée (3/5)
chr22	41531859	C	T	0.67	0.0484	6	0.497	1	confiance = moyenne (2/5)
chr22	41566525	A	-	1.34	0.0000	5	0.039	7	confiance = moyenne (2/5)
chr22	41574986	A	-	1.02	0.0000	5	0.311	7	confiance = moyenne (2/5)
chr6	401679	T	C	0.71	0.0000	6	0.32	1	confiance = élevée (3/5)
chr8	128750605	C	-	1.75	0.0000	6	0.37	2	confiance = forte (4/5)
DeepSNVMiner & SiNVICT (1)									
chr6	106534484	A	AT	9.82	0.0000	0	0.068	12	aucun UMI concordant
DeepSNVMiner & outLyzer (2)									

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
chr22	41531856	A	G	0.5	0.0028	0	0.171	3	aucun UMI concordant
chr9	21968219	C	T	1.06	0.0000	2	1.288	1	peu d'UMI concordants ; biais de brin
SiNVICT & outLyzer (11)									
chr16	3808053	A	-	25.54	0.0000	118	0.053	13	région homopolymérique (>7)
chr16	3828848	T	-	17.42	0.0000	28	0.059	10	région homopolymérique (>7)
chr17	7577679	T	-	46.44	0.0000	195	0.068	16	région homopolymérique (>7)
chr1	117057449	A	-	6.54	0.0000	23	0.022	9	région homopolymérique (>7)
chr1	117087236	A	-	6.48	0.0000	19	0.258	9	région homopolymérique (>7)
chr22	41545025	T	-	33.41	0.0000	90	0.012	14	région homopolymérique (>7)
chr22	41572224	A	-	5.37	0.0000	16	0.009	8	région homopolymérique (>7)
chr2	136872415	A	-	3.85	0.0000	8	0.002	8	région homopolymérique (>7)
chr6	106534485	T	-	18.29	0.0000	88	0.032	12	région homopolymérique (>7)
chr9	22003879	T	-	3.06	0.0000	9	0.155	8	région homopolymérique (>7)
chr9	22005111	A	ATG	26.8	0.0000	0	0.166	1	aucun UMI concordant
SiNVICT (16)									
chr16	3808052	T	TA	12.49	0.0000	0	0.001	13	aucun UMI concordant
chr16	81954790	T	-	5.66	0.0000	30	0.03	9	région homopolymérique (>7)
chr17	7577678	C	CT	5.41	0.0000	0	0.367	16	aucun UMI concordant
chr19	19257114	A	C	1.89	0.0000	0	1.744	1	aucun UMI concordant
chr19	19257126	C	G	1.25	0.9855	0	1.781	1	q-value > 0.05
chr19	19257137	C	G	2.33	0.0019	0	0.467	2	aucun UMI concordant
chr19	19257143	T	G	3.31	0.0000	0	1.234	1	aucun UMI concordant
chr19	19257150	C	G	2.18	0.0000	0	1.064	1	aucun UMI concordant
chr9	21971164	A	G	1.9	0.0000	0	0.544	1	aucun UMI concordant
chr9	22005933	T	G	0.19	1.2000	0	0.615	1	q-value > 0.05
chr12	57496315	T	C	-	-	-	-	-	position hors fichier BED
chr17	36577033	C	G	-	-	-	-	-	position hors fichier BED
chr17	79786722	G	C	-	-	-	-	-	position hors fichier BED
chr19	29482011	G	A	-	-	-	-	-	position hors fichier BED
chr3	197024800	C	T	-	-	-	-	-	position hors fichier BED
chr7	158185187	T	C	-	-	-	-	-	position hors fichier BED
outLyzer (58)									
chr12	57496748	C	T	1.68	0.0000	1	5.682	2	peu d'UMI concordants ; biais de brin
chr13	41134552	A	-	1.25	0.0000	3	1.013	3	peu d'UMI concordants ; biais de brin
chr16	10989204	A	G	0.83	0.0000	3	0.318	1	peu d'UMI concordants
chr16	10989593	A	G	0.91	0.0000	4	0.366	2	peu d'UMI concordants
chr16	11010241	A	G	0.62	0.0000	2	0.752	1	peu d'UMI concordants
chr16	11010274	T	C	0.45	0.0228	4	0.039	1	peu d'UMI concordants
chr16	3786142	T	C	1.3	0.0000	3	1.841	2	peu d'UMI concordants ; biais de brin
chr16	3795299	T	C	0.63	0.0083	2	0.392	1	peu d'UMI concordants
chr16	3808913	T	C	0.63	0.0000	1	0.17	1	peu d'UMI concordants
chr16	3824629	G	A	0.65	0.0228	3	0.289	1	peu d'UMI concordants
chr16	81946285	A	G	0.53	0.0019	1	0.397	2	peu d'UMI concordants
chr16	81954788	T	TG	1.09	0.0000	0	0.001	1	aucun UMI concordant
chr16	81957117	G	A	0.67	0.0000	3	0.123	1	peu d'UMI concordants
chr17	63010473	A	G	0.89	0.0000	1	1.948	2	peu d'UMI concordants ; biais de brin
chr17	63049864	A	-	2.9	0.0000	9	0.11	8	région homopolymérique (>7)
chr17	7572963	T	-	0.77	0.0011	4	0.016	6	peu d'UMI concordants
chr17	7577693	T	G	1.71	0.0000	5	0.574	16	région homopolymérique (>7)
chr17	7579858	A	G	0.63	0.0000	3	0.228	4	peu d'UMI concordants
chr18	60985808	T	C	0.61	0.0019	3	0.271	1	peu d'UMI concordants
chr19	1612276	T	C	0.61	0.0028	2	0.057	3	peu d'UMI concordants
chr19	19257008	G	A	1.27	0.0000	2	0.038	3	peu d'UMI concordants
chr19	19261566	A	G	0.51	0.0083	1	0.627	1	peu d'UMI concordants
chr1	117057369	A	G	0.89	0.0000	4	0.271	3	peu d'UMI concordants
chr1	117057375	A	G	0.8	0.0000	2	0	1	peu d'UMI concordants
chr1	117057395	A	G	0.66	0.0004	3	0.12	4	peu d'UMI concordants
chr1	117078855	A	-	2.52	0.0000	6	0.115	8	région homopolymérique (>7)
chr1	117087224	A	G	0.91	0.0000	4	0.027	3	peu d'UMI concordants
chr1	120458844	A	G	0.95	0.0000	4	0.495	1	peu d'UMI concordants
chr1	120464965	T	C	1.05	0.0000	4	0.009	3	peu d'UMI concordants
chr1	120466563	T	C	0.5	0.0046	3	0.111	1	peu d'UMI concordants
chr1	27097701	T	C	0.98	0.0003	4	0.023	1	peu d'UMI concordants
chr22	41488998	A	G	0.5	0.0605	1	0.066	3	q-value > 0.05 ; peu d'UMI concordants
chr22	41488999	A	G	0.69	0.0028	1	0.479	3	peu d'UMI concordants
chr22	41489009	A	G	0.55	0.0019	1	0.244	5	peu d'UMI concordants
chr22	41489019	A	G	0.54	0.1085	1	0.616	2	q-value > 0.05
chr22	41489024	G	A	0.59	0.0550	2	0.763	2	q-value > 0.05 ; peu d'UMI concordants
chr22	41489055	C	T	0.55	0.0228	1	0.39	2	peu d'UMI concordants
chr22	41489059	A	G	0.43	0.1443	2	0.343	3	q-value > 0.05
chr22	41523681	T	C	0.55	0.0074	4	0.389	2	peu d'UMI concordants
chr22	41533734	T	C	0.61	0.0009	2	0.44	2	peu d'UMI concordants
chr22	41542773	G	A	0.99	0.1085	2	0.014	1	q-value > 0.05
chr22	41547821	T	-	1.22	0.0000	0	1.584	8	aucun UMI concordant
chr22	41560091	C	T	0.58	0.0000	2	0.346	1	peu d'UMI concordants
chr22	41560107	T	C	0.52	0.0029	4	0.111	2	peu d'UMI concordants
chr2	136872393	A	-	2.77	0.0000	7	0.043	8	région homopolymérique (>7)
chr2	136872464	T	C	1.24	0.0000	4	0.656	1	peu d'UMI concordants
chr3	38182317	A	G	0.9	0.0000	4	0.687	3	peu d'UMI concordants
chr6	106553168	A	G	0.6	0.0009	2	0.091	2	peu d'UMI concordants
chr6	106554278	T	C	0.6	0.0002	3	0.005	3	peu d'UMI concordants
chr6	106554325	T	C	0.44	0.0004	4	0.35	1	peu d'UMI concordants
chr6	106554350	G	-	0.9	0.0180	3	0.466	3	peu d'UMI concordants
chr6	138198200	T	-	1.65	0.0000	4	0.633	7	peu d'UMI concordants
chr6	37138977	T	C	0.81	0.0001	5	1.053	1	biais de brin
chr6	393334	A	G	1.16	0.0000	5	1.067	1	biais de brin
chr9	139390824	A	G	0.53	0.0024	3	0.025	2	peu d'UMI concordants
chr9	139391574	G	A	0.82	0.0003	2	1.093	4	peu d'UMI concordants ; biais de brin
chr9	22003277	A	-	1.98	0.0000	5	0.048	8	région homopolymérique (>7)

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
chr9	22003723	A	-	0.97	0.0000	4	1.745	5	peu d'UMI concordants ; biais de brin
DeepSNVMiner (114)									
chr12	57496096	C	T	0.52	0.0228	1	1.491	1	peu d'UMI concordants ; biais de brin
chr12	57496680	T	C	0.5	0.0228	1	0.335	1	peu d'UMI concordants
chr13	41133896	A	G	0.18	1.2000	1	0.267	1	q-value > 0.05
chr13	41133953	T	C	0.33	0.1443	1	0.419	2	q-value > 0.05
chr13	41239884	G	A	0.77	0.0083	3	0.046	3	peu d'UMI concordants
chr16	10989638	G	A	0.78	0.0007	2	0.164	1	peu d'UMI concordants
chr16	10992856	A	G	0.45	0.0074	1	0.241	4	peu d'UMI concordants
chr16	10997670	C	T	0.21	0.8792	1	0.253	1	q-value > 0.05
chr16	11000800	T	C	1.46	0.0000	1	0.944	4	peu d'UMI concordants
chr16	11001185	T	C	0.17	0.6337	0	0.199	3	q-value > 0.05
chr16	11001571	C	T	0.4	0.0605	1	0.617	4	q-value > 0.05 ; peu d'UMI concordants
chr16	11001829	A	G	0.33	0.4501	1	1.58	1	q-value > 0.05
chr16	11010188	C	T	0.56	0.4501	1	0.374	1	q-value > 0.05
chr16	11010195	T	C	0.71	0.0000	0	0.523	1	aucun UMI concordant
chr16	11012360	C	T	0.18	0.9867	1	0.514	1	q-value > 0.05
chr16	11017054	A	G	0.71	0.0003	1	0.001	2	peu d'UMI concordants
chr16	3777763	C	T	0.54	0.0083	1	0.05	4	peu d'UMI concordants
chr16	3778019	C	T	0.25	0.3569	1	0.936	1	q-value > 0.05
chr16	3778457	C	T	0.19	0.7365	0	0.799	1	q-value > 0.05
chr16	3789604	A	G	0.64	0.0028	1	0.652	1	peu d'UMI concordants
chr16	3789674	A	G	0.32	0.1919	1	0.615	1	q-value > 0.05
chr16	3790419	T	C	0.33	0.0641	0	0.927	2	q-value > 0.05 ; aucun UMI concordant
chr16	3807800	A	G	0.31	0.3569	0	0.247	4	q-value > 0.05
chr16	3820892	C	T	0.34	0.1085	1	0.121	1	q-value > 0.05
chr16	3823842	G	A	0.53	0.0074	3	0.848	2	peu d'UMI concordants
chr16	3828847	C	CT	2.59	0.0000	0	0.526	10	aucun UMI concordant
chr16	3830802	C	T	0.34	0.0605	2	0.455	1	q-value > 0.05 ; peu d'UMI concordants
chr16	3831262	T	C	0.2	0.4501	0	0.307	1	q-value > 0.05
chr16	3842026	A	G	0.43	0.0761	1	0.283	3	q-value > 0.05 ; peu d'UMI concordants
chr16	3900853	A	G	0.28	0.8792	1	0.671	1	q-value > 0.05
chr17	62006787	T	C	0.62	0.0001	3	0.505	1	peu d'UMI concordants
chr17	7578259	A	G	0.27	0.1964	1	0.298	1	q-value > 0.05
chr19	19256731	C	G	7.93	0.0000	35	1.14	5	biais de brin
chr19	19256768	T	G	1.89	0.0000	0	2.276	1	aucun UMI concordant
chr19	19256780	C	G	0.23	1.2000	0	1.583	1	q-value > 0.05
chr19	42384854	G	A	0.87	0.0156	0	0.561	5	aucun UMI concordant
chr1	117113582	T	C	0.55	0.0083	1	0.096	1	peu d'UMI concordants
chr1	120464923	G	A	0.64	0.0029	2	0.514	1	peu d'UMI concordants
chr1	2491452	C	T	0.6	0.0004	1	0.741	2	peu d'UMI concordants
chr1	2492111	T	C	0.36	0.0288	2	0.209	2	peu d'UMI concordants
chr1	2494588	G	A	0.49	0.5688	0	0.066	2	q-value > 0.05
chr1	27024050	A	G	0.94	0.0074	1	0.127	1	peu d'UMI concordants
chr1	27057999	A	G	0.52	0.0083	0	1.266	1	aucun UMI concordant
chr1	27087518	T	C	3.19	0.0000	3	1.801	1	peu d'UMI concordants ; biais de brin
chr1	27087830	C	T	0.63	0.0028	2	0.129	3	peu d'UMI concordants
chr1	27089615	G	A	0.25	0.5688	0	0.518	2	q-value > 0.05
chr1	27094317	A	G	0.46	0.0028	3	0.745	1	peu d'UMI concordants
chr1	27099892	T	G	17.7	0.0000	8	1.42	1	biais de brin
chr1	27101028	C	T	0.22	0.8792	1	1.263	2	q-value > 0.05
chr1	27102089	T	G	16.08	0.0000	24	1.784	1	biais de brin
chr1	27102166	G	A	0.34	0.4305	1	0.113	1	q-value > 0.05
chr1	27102169	A	G	0.34	0.1171	1	0.099	2	q-value > 0.05
chr1	27106327	C	T	0.28	0.9855	1	0.85	1	q-value > 0.05
chr22	41513500	A	G	0.19	0.9867	1	0.022	2	q-value > 0.05
chr22	41527604	T	C	0.42	0.0407	2	0.314	1	peu d'UMI concordants
chr22	41527624	C	T	0.27	0.1740	1	0.811	3	q-value > 0.05
chr22	41531895	C	T	0.29	0.5688	0	0.425	2	q-value > 0.05
chr22	41545878	A	G	0.76	0.0003	2	0.366	1	peu d'UMI concordants
chr22	41545940	A	G	0.29	0.8235	0	0.211	1	q-value > 0.05
chr22	41547972	G	A	0.23	0.8792	0	0.346	2	q-value > 0.05
chr22	41551117	T	C	0.42	0.0228	0	0.08	1	aucun UMI concordant
chr22	41572940	T	C	0.32	0.0605	1	1.426	1	q-value > 0.05 ; is most probably str ; biased
chr22	41573074	G	T	0.43	0.8989	0	1.504	1	q-value > 0.05
chr22	41573153	T	C	0.65	0.0228	1	0.858	1	peu d'UMI concordants
chr22	41574388	C	T	0.6	0.0001	3	0.615	1	peu d'UMI concordants
chr6	106536078	T	C	0.24	0.9066	0	1.157	1	q-value > 0.05
chr6	106553193	G	A	0.46	0.1443	1	0.888	3	q-value > 0.05
chr6	106553274	C	T	0.63	0.0180	1	0.552	2	peu d'UMI concordants
chr6	106553279	T	C	0.49	0.0083	1	0.112	2	peu d'UMI concordants
chr6	106553364	G	A	0.31	0.2750	1	0.374	1	q-value > 0.05
chr6	106553379	C	T	0.15	0.9066	0	0.106	1	q-value > 0.05
chr6	106553659	A	C	3.12	0.0000	0	0.984	1	aucun UMI concordant
chr6	106554298	A	G	0.22	0.4501	0	0.205	2	q-value > 0.05
chr6	106555111	A	T	0.83	0.0000	1	0.548	1	peu d'UMI concordants
chr6	106555156	A	G	0.41	0.0083	2	0.218	3	peu d'UMI concordants
chr6	106555158	A	G	0.33	0.0605	0	0.564	3	q-value > 0.05 ; aucun UMI concordant
chr6	106555175	A	G	0.54	0.0002	3	0.037	4	peu d'UMI concordants
chr6	138192467	G	A	0.37	0.0074	2	0.001	3	peu d'UMI concordants
chr6	138197348	C	T	0.9	0.0001	1	0.436	1	peu d'UMI concordants
chr6	138200108	C	T	0.47	0.0641	1	0.119	4	q-value > 0.05 ; peu d'UMI concordants
chr6	138200363	C	T	0.49	0.0228	1	0.697	1	peu d'UMI concordants
chr6	37140863	T	C	0.2	0.2750	0	0.449	1	q-value > 0.05
chr6	37140884	T	C	0.41	0.0180	2	0.083	2	peu d'UMI concordants
chr6	393308	C	T	0.37	0.7851	2	0.027	4	q-value > 0.05
chr6	395909	A	G	0.23	0.8235	0	0.764	2	q-value > 0.05
chr6	405033	C	T	0.4	0.0761	1	0.264	1	q-value > 0.05 ; peu d'UMI concordants
chr7	148508697	A	G	0.12	1.2000	0	0.13	1	q-value > 0.05
chr7	2976681	T	G	0.56	0.0465	4	0.832	2	peu d'UMI concordants
chr7	2976688	T	C	0.16	0.9867	1	0.469	1	q-value > 0.05
chr7	2976717	T	C	1.48	0.0000	0	2.493	1	aucun UMI concordant
chr7	2978297	C	T	0.36	0.7365	2	0.338	1	q-value > 0.05
chr7	2978448	C	T	0.27	0.2750	0	0.322	1	q-value > 0.05
chr8	128750680	A	C	3.47	0.0000	0	3.502	1	aucun UMI concordant

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
chr8	128751198	C	A	0.1	1.2000	1	0.052	4	q-value > 0.05
chr8	128751210	C	T	0.36	0.1919	2	0.608	4	q-value > 0.05
chr8	128752844	T	G	0.1	1.2000	0	0.915	1	q-value > 0.05
chr8	128752965	G	A	0.23	1.2000	1	1.27	1	q-value > 0.05
chr8	128753020	A	G	0.37	0.0796	3	0.16	1	q-value > 0.05; peu d'UMI concordants
chr9	139390558	A	C	0.22	0.9867	0	0.25	1	q-value > 0.05
chr9	21967819	C	T	0.43	0.1443	1	0.832	1	q-value > 0.05
chr9	21967936	A	G	0.31	0.5688	0	0.732	1	q-value > 0.05
chr9	22003389	A	T	0.2	0.7365	1	1.399	1	q-value > 0.05
chr9	22003749	T	TG	37.91	0.0000	0	0.064	8	aucun UMI concordant
chr9	22003908	T	C	0.47	0.0550	1	0.815	1	q-value > 0.05; peu d'UMI concordants
chr9	22003996	T	C	0.45	0.0007	1	0.043	3	peu d'UMI concordants
chr9	22004990	G	A	0.26	0.7365	0	0.491	1	q-value > 0.05
chr9	22005007	T	C	0.73	0.0028	2	0.254	2	peu d'UMI concordants
chr9	22005235	T	C	0.62	0.0046	3	0.171	1	peu d'UMI concordants
chr9	22005622	G	A	0.34	0.1740	1	0.96	3	q-value > 0.05
chr9	22005769	C	T	0.27	0.9385	0	1.273	1	q-value > 0.05
chr9	22005824	G	A	0.37	0.3606	1	0.469	2	q-value > 0.05
chr9	22005841	C	T	0.64	0.0000	1	0.487	2	peu d'UMI concordants
chr9	22009155	T	C	0.64	0.0228	0	0.571	4	aucun UMI concordant
chr9	22009225	A	G	0.22	0.9385	0	0.078	1	q-value > 0.05
UMI-VarCal (45)									
chr16	10997628	A	G	0.48	0.0228	5	0.805	1	confiance = moyenne (2/5)
chr16	11003993	G	A	5.91	0.0000	7	0.908	3	confiance = élevée (3/5)
chr16	11004083	C	T	0.6	0.0003	5	0.845	3	confiance = élevée (3/5)
chr16	11017165	A	G	0.6	0.0000	7	0.396	1	confiance = forte (4/5)
chr16	3778077	T	C	0.35	0.0029	5	0.072	1	confiance = élevée (3/5)
chr16	3794968	A	G	0.67	0.0009	5	0.453	4	confiance = élevée (3/5)
chr16	3819180	A	G	1.32	0.0000	5	0.42	1	confiance = élevée (3/5)
chr16	3823894	T	C	0.47	0.0180	5	0.354	2	confiance = élevée (3/5)
chr16	3832670	A	G	1.02	0.0000	5	0.404	1	confiance = élevée (3/5)
chr16	81942168	T	C	0.73	0.0002	5	0.376	2	confiance = élevée (3/5)
chr16	81953203	T	C	0.64	0.0000	5	0.691	2	confiance = élevée (3/5)
chr16	81960759	A	G	0.44	0.0000	5	0.109	1	confiance = élevée (3/5)
chr17	62006806	A	G	0.41	0.0046	5	0.148	1	confiance = élevée (3/5)
chr17	63010689	T	C	0.61	0.0003	5	0.62	3	confiance = moyenne (2/5)
chr19	1612399	T	C	0.69	0.0000	5	0.047	3	confiance = élevée (3/5)
chr1	117078806	T	C	0.55	0.0002	5	0.208	2	confiance = forte (4/5)
chr1	120458614	A	G	0.59	0.0009	5	0.351	1	confiance = forte (4/5)
chr1	120465056	A	G	0.7	0.0000	5	0.458	1	confiance = élevée (3/5)
chr1	120465315	A	G	0.68	0.0000	7	0.345	1	confiance = forte (4/5)
chr1	120466577	T	C	0.39	0.0074	5	0.149	3	confiance = moyenne (2/5)
chr1	2488098	T	C	0.33	0.0407	5	0.07	1	confiance = élevée (3/5)
chr1	2489809	A	G	0.58	0.0180	5	0.384	1	confiance = moyenne (2/5)
chr1	2492059	T	C	0.79	0.0000	5	0.735	1	confiance = élevée (3/5)
chr1	27087413	A	G	0.47	0.0028	5	0.232	1	confiance = élevée (3/5)
chr22	41565603	A	G	0.47	0.0028	5	0.072	2	confiance = élevée (3/5)
chr22	41573581	A	G	0.47	0.0144	5	0.293	1	confiance = élevée (3/5)
chr2	136872893	T	C	0.73	0.0000	5	0.977	1	confiance = moyenne (2/5)
chr2	61719294	A	G	0.4	0.0180	5	0.041	3	confiance = élevée (3/5)
chr2	61719501	A	G	0.66	0.0004	5	0.883	3	confiance = élevée (3/5)
chr3	38182736	T	C	0.72	0.0000	5	0.512	2	confiance = élevée (3/5)
chr6	138196011	T	C	0.64	0.0004	5	0.431	2	confiance = forte (4/5)
chr6	138196031	T	C	0.77	0.0001	5	0.68	2	confiance = élevée (3/5)
chr6	405070	T	C	0.49	0.0003	5	0.077	1	confiance = élevée (3/5)
chr7	2978434	T	C	0.47	0.0028	5	0.207	2	confiance = forte (4/5)
chr7	2985482	T	C	0.57	0.0001	5	0.339	1	confiance = élevée (3/5)
chr8	128750825	T	C	0.5	0.0009	5	0.47	1	confiance = élevée (3/5)
chr8	128752820	T	C	0.58	0.0001	9	0.167	1	confiance = élevée (3/5)
chr8	128752879	T	C	0.53	0.0024	5	0.5	1	confiance = élevée (3/5)
chr8	128753099	C	T	1.02	0.0000	10	0.888	2	confiance = moyenne (2/5)
chr9	139391910	T	C	0.7	0.0074	5	0.662	1	confiance = moyenne (2/5)
chr9	139392051	A	G	42.34	0.0000	113	0.013	1	confiance = certaine (5/5)
chr9	22003084	T	C	0.68	0.0000	5	0.606	4	confiance = élevée (3/5)
chr9	22005113	G	-	16.7	0.0000	19	0.134	1	confiance = forte (4/5)
chr9	22005695	A	G	0.85	0.0000	5	0.934	4	confiance = moyenne (2/5)
chr9	22005860	A	G	0.58	0.0000	6	0.247	1	confiance = élevée (3/5)

TABLE 5.5 – Liste détaillée des variants détectés par UMI-VarCal, DeepSNVMiner, SiNVICT et outLyzer dans l'échantillon Y.

q-value	UMI	UMI HP	*q-value* & UMI	UMI & SB	SB	*q-value* & SB	*q-value* & HP	UMI & HP	SB & HP	Total
58	97	16	11	9	5	1	0	0	0	197
29?44%	49,24%	8,12%	5,58%	4,57%	2,54%	0,51%	0,00%	0,00%	0,00%	

TABLE 5.6 – Nombre de discordances classées par type pour l'échantillon Y.

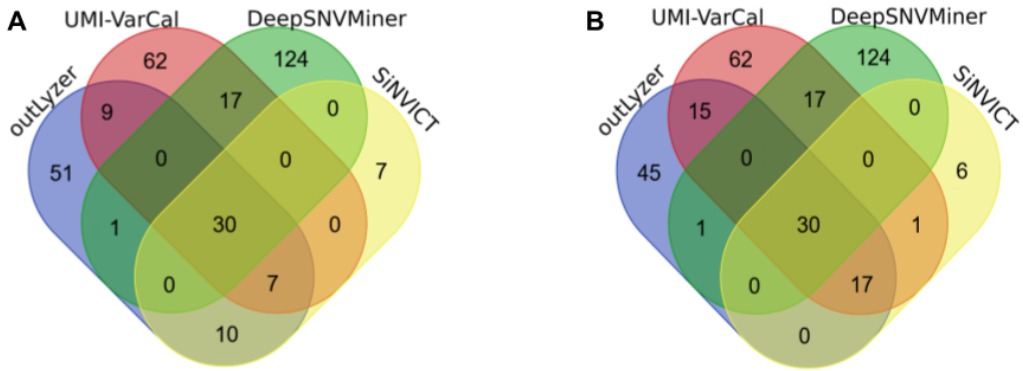


FIGURE 5.10 – **(A)** Un diagramme de Venn représentant les variants détectés par UMI-VarCal, DeepSNVMiner, SiNVICT et outLyzer dans l'échantillon Z. **(B)** Un diagramme de Venn représentant les variants détectés par UMI-VarCal (sans le filtre de biais de brin ni le filtre des régions homopolymériques), DeepSNVMiner, SiNVICT et outLyzer dans l'échantillon Z.

Échantillon Z

L'analyse effectuée sur le dernier échantillon a permis la détection de 318 variants au total (tous les variants sont détaillés dans la Table 5.7) (Figure 5.10A). UMI-VarCal a détecté 125 variants, tandis que DeepSNVMiner, outLyzer et SiNVICT en ont détecté 172, 108 et 54 respectivement. Parmi ces 145 variants, 30 sont également trouvés par les trois autres outils et 83 ont été trouvés par au moins un autre logiciel.
124 variants n'ont été trouvés que par DeepSNVMiner : 88/124 n'ont pas réussi le test de Poisson et 36/124 n'ont pas réussi le test d'analyse des UMI.
51 variants ont été détectés uniquement par outLyzer : 45/51 n'ont pas réussi le test d'analyse des UMI, 1/51 sont probablement des artefacts dus au biais de brin et 5/51 sont dans une longue région homopolymérique.
Sept variants ont été trouvés uniquement par SiNVICT : 3/7 n'ont pas réussi le test d'analyse UMI, 1/7 est dans une longue région homopolymérique et 4/7 sont détectés dans des positions qui ne sont pas couvertes par le fichier BED fourni.
Dix variants ont été appelés par SiNVICT et outLyzer : 9/10 sont dans une longue région homopolymérique et le dernier a un biais de brin élevé.
Un seul variant a été détecté à la fois par DeepSNVMiner et outLyzer : ce variant n'a pas réussi le test d'analyse UMI.
62 variants ont été détectés uniquement par UMI-VarCal : 51/62 (82,3%) ont une fréquence inférieure à 1% et 10/62 (16,1%) ont une fréquence inférieure à 0,5%. De plus, aucun des 62 variants n'avait un faible niveau de confiance alors que 55/62 (88,7%) avaient au moins un niveau de confiance élevé.
Enfin, comme nous l'avons fait pour les échantillons X et Y, nous avons lancé UMI-VarCal sans le filtre de biais de brin ni le filtre de la région homopolymérique (Figure 5.10B). Au total, 17/190 (9%) discordances sont causées soit par le filtre homopolymère, soit par le filtre de biais de brin, ce qui signifie que 91% des discordances sont causées par une *q-value* trop élevée ou un nombre insuffisants d'UMI concordants (le nombre de discordances par type de discordance pour l'échantillon Z se trouvent dans la Table 5.8).

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
UMI-VarCal & DeepSNVMiner & SiNVICT & outLyzer (30)									
chr12	57496662	C	A	39.13	0.0000	221	0.011	1	confiance = certaine (5/5)
chr16	10971222	A	C	38.89	0.0000	236	0.029	1	confiance = certaine (5/5)
chr16	10972899	G	T	55.73	0.0000	170	0.146	2	confiance = certaine (5/5)
chr16	11001743	G	A	49.61	0.0000	407	0.025	4	confiance = forte (4/5)
chr16	11002904	G	A	99.64	0.0000	958	0.001	2	confiance = certaine (5/5)
chr16	11002927	A	G	99.8	0.0000	499	0.002	1	confiance = certaine (5/5)
chr16	11004150	C	T	49.88	0.0000	196	0.111	1	confiance = certaine (5/5)
chr16	11009587	C	A	50.59	0.0000	169	0.027	2	confiance = certaine (5/5)
chr16	11017058	G	A	50.83	0.0000	331	0.019	2	confiance = certaine (5/5)
chr16	11348929	T	G	40.74	0.0000	210	0.053	1	confiance = certaine (5/5)
chr16	11349098	A	G	41.41	0.0000	303	0.004	1	confiance = certaine (5/5)
chr16	11349107	A	G	43.64	0.0000	319	0.047	1	confiance = certaine (5/5)
chr16	11349290	C	T	41.15	0.0000	173	0.063	1	confiance = certaine (5/5)
chr16	3779594	C	T	48.97	0.0000	230	0.003	1	confiance = certaine (5/5)
chr16	81954789	C	G	48.09	0.0000	434	0.025	1	confiance = certaine (5/5)
chr17	7579472	G	C	47.82	0.0000	459	0.009	6	confiance = forte (4/5)
chr19	19256870	G	A	49.44	0.0000	226	0.007	1	confiance = certaine (5/5)
chr1	23885498	T	C	49.38	0.0000	304	0.021	1	confiance = certaine (5/5)
chr1	2488153	A	G	99.73	0.0000	942	0	4	confiance = forte (4/5)
chr1	2491205	C	T	99.76	0.0000	389	0.001	3	confiance = forte (4/5)
chr1	27099906	G	A	50.37	0.0000	325	0.002	1	confiance = certaine (5/5)
chr22	41537234	G	T	99.51	0.0000	530	0.002	2	confiance = certaine (5/5)
chr22	41551039	T	A	53.01	0.0000	206	0.026	3	confiance = forte (4/5)
chr3	38182136	C	G	52.7	0.0000	261	0.013	2	confiance = certaine (5/5)
chr6	138192518	A	T	42.73	0.0000	310	0.002	1	confiance = certaine (5/5)
chr6	138197331	A	C	50.87	0.0000	194	0.018	1	confiance = certaine (5/5)
chr6	41903782	A	C	99.55	0.0000	651	0.003	1	confiance = certaine (5/5)
chr8	128750674	C	T	33.99	0.0000	187	0.076	1	confiance = certaine (5/5)
chr9	139390779	G	A	25.18	0.0000	244	0.11	1	confiance = certaine (5/5)
chr9	22003367	G	A	99.85	0.0000	910	0.001	2	confiance = certaine (5/5)
UMI-VarCal & SiNVICT & outLyzer (7)									
chr17	7579801	G	C	58.82	0.0000	142	0.024	5	confiance = forte (4/5)
chr1	117078820	G	-	29.96	0.0000	101	0.217	1	confiance = certaine (5/5)
chr1	27100182	G	-	3.74	0.0000	11	0.231	1	confiance = forte (4/5)
chr22	41568480	T	C	49.52	0.0000	194	0.002	1	confiance = certaine (5/5)
chr6	138198392	A	-	35.13	0.0000	132	0.061	3	confiance = forte (4/5)
chr9	21974741	C	T	15.43	0.0000	95	0.131	3	confiance = forte (4/5)
chr9	22005112	T	-	59.01	0.0000	276	0.138	1	confiance = certaine (5/5)
UMI-VarCal & DeepSNVMiner (17)									
chr12	57496668	T	A	39.79	0.0000	214	0.007	2	confiance = certaine (5/5)
chr16	10972922	G	A	63.14	0.0000	165	0.046	1	confiance = certaine (5/5)
chr16	10995933	A	G	99.89	0.0000	531	0.001	1	confiance = certaine (5/5)
chr16	11349018	G	T	45.15	0.0000	213	0.035	2	confiance = certaine (5/5)
chr16	11349146	A	G	43.43	0.0000	207	0.066	2	confiance = certaine (5/5)
chr16	11349332	C	G	43.9	0.0000	153	0.009	2	confiance = certaine (5/5)
chr16	11349333	C	T	44.03	0.0000	153	0.003	2	confiance = certaine (5/5)
chr16	11349340	C	T	40.99	0.0000	144	0.024	1	confiance = certaine (5/5)
chr16	81953081	T	C	99.88	0.0000	455	0.001	1	confiance = certaine (5/5)
chr1	27057621	A	C	51.95	0.0000	191	0.042	4	confiance = forte (4/5)
chr6	41905174	T	A	99.61	0.0000	516	0.002	1	confiance = certaine (5/5)
chr7	148508833	A	G	98.81	0.0000	407	0.012	1	confiance = certaine (5/5)
chr9	139391636	G	A	64.8	0.0000	704	0.015	1	confiance = certaine (5/5)
chr9	21968159	G	A	30.77	0.0000	130	0.057	2	confiance = certaine (5/5)
chr9	21968199	C	G	32.56	0.0000	97	0.081	3	confiance = forte (4/5)
chr9	22005330	T	G	34.06	0.0000	352	0.07	1	confiance = certaine (5/5)
chr9	22006273	G	T	44.74	0.0000	200	0.162	1	confiance = certaine (5/5)
UMI-VarCal & SiNVICT (2)									
chr19	19257137	C	G	2.31	0.0021	0	0.322	2	aucun UMI concordant
chr22	41545906	A	C	1.6	0.0000	0	0.291	1	aucun UMI concordant
UMI-VarCal & outLyzer (9)									
chr16	10989611	T	C	0.51	0.0002	5	0.45	1	confiance = élevée (3/5)
chr16	11348889	C	G	2.34	0.0000	10	0.294	1	confiance = certaine (5/5)
chr16	3779136	T	-	1.82	0.0000	6	0.115	1	confiance = forte (4/5)
chr16	3799608	A	G	0.81	0.0000	6	0.318	6	confiance = élevée (3/5)
chr16	3817721	T	-	1.54	0.0000	5	0.076	7	confiance = moyenne (2/5)
chr17	63052468	C	-	2.93	0.0000	5	0.384	1	confiance = forte (4/5)
chr19	42384952	T	C	0.75	0.0030	5	0.295	1	confiance = élevée (3/5)
chr9	139390945	G	-	2.17	0.0000	7	0.213	1	confiance = forte (4/5)
chr9	22005807	A	-	0.91	0.0000	6	0.036	7	confiance = moyenne (2/5)
DeepSNVMiner & outLyzer (1)									
chr16	3830834	G	A	0.59	0.0008	1	0.213	1	peu d'UMI concordants
SiNVICT & outLyzer (10)									
chr16	10972818	G	C	7.56	0.0000	14	2.312	1	biais de brin
chr16	3808053	A	-	24.33	0.0000	135	0.007	13	région homopolymérique (>7)
chr16	3828848	T	-	19.4	0.0000	46	0.061	10	région homopolymérique (>7)
chr17	7577679	T	-	47.04	0.0000	202	0.08	16	région homopolymérique (>7)
chr1	117057449	A	-	6.06	0.0000	11	0.23	9	région homopolymérique (>7)
chr1	117087236	A	-	6.61	0.0000	23	0.198	9	région homopolymérique (>7)
chr22	41545025	T	-	33.39	0.0000	148	0.029	14	région homopolymérique (>7)
chr22	41572224	A	-	4.06	0.0000	13	0.001	8	région homopolymérique (>7)

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
chr6	106534485	T	-	20.83	0.0000	112	0.056	12	région homopolymérique (>7)
chr9	22003879	T	-	2.58	0.0000	7	0.071	8	région homopolymérique (>7)
SiNVICT (7)									
chr16	81954790	T	-	6.48	0.0000	38	0.138	9	région homopolymérique (>7)
chr19	19257137	C	G	2.31	0.0021	0	0.322	2	aucun UMI concordant
chr19	19257143	T	G	3.51	0.0000	0	1.74	1	aucun UMI concordant
chr22	41545906	A	C	1.6	0.0000	0	0.291	1	aucun UMI concordant
chr16	3828847	C	CT	-	-	-	-	-	position hors fichier BED
chr19	29482011	G	A	-	-	-	-	-	position hors fichier BED
chr3	197024800	C	T	-	-	-	-	-	position hors fichier BED
outLyzer (51)									
chr15	45008499	T	-	3.62	0.0000	6	0.027	8	région homopolymérique (>7)
chr16	10989204	A	G	1.19	0.0000	4	0.221	1	peu d'UMI concordants
chr16	10989584	A	G	0.65	0.0000	3	0.302	2	peu d'UMI concordants
chr16	10989625	A	G	0.79	0.0001	3	0.04	1	peu d'UMI concordants
chr16	11004080	T	C	0.92	0.0000	3	0.05	1	peu d'UMI concordants
chr16	11017121	A	G	0.65	0.0001	4	0.303	2	peu d'UMI concordants
chr16	11017132	A	G	0.44	0.0089	3	0.022	2	peu d'UMI concordants
chr16	3779046	T	C	0.46	0.0030	1	0.339	2	peu d'UMI concordants
chr16	3790505	T	C	0.64	0.0008	1	1.255	2	peu d'UMI concordants ; biais de brin
chr16	3830806	A	G	0.58	0.0008	3	0.027	2	peu d'UMI concordants
chr16	81944229	T	C	0.93	0.0000	1	2.735	1	peu d'UMI concordants ; biais de brin
chr16	81954788	T	TG	1.48	0.0000	0	0.001	1	aucun UMI concordant
chr16	81954799	C	T	0.75	0.0000	3	0.027	2	peu d'UMI concordants
chr17	63049864	A	-	2.82	0.0000	7	0.192	8	région homopolymérique (>7)
chr17	63052484	G	A	1.04	0.0001	4	0.358	1	peu d'UMI concordants
chr17	63052516	T	C	0.55	0.0079	2	0.113	2	peu d'UMI concordants
chr17	63052587	T	C	0.98	0.0000	3	1.461	4	peu d'UMI concordants ; biais de brin
chr19	42384941	A	G	0.51	0.0089	1	0.145	2	peu d'UMI concordants
chr1	117078855	A	-	2.53	0.0000	6	0.243	8	région homopolymérique (>7)
chr1	2491264	A	G	0.57	0.0000	2	0.758	1	peu d'UMI concordants
chr1	2491291	T	C	0.51	0.0239	1	0.472	1	peu d'UMI concordants
chr1	2494653	A	G	0.6	0.0003	1	0.423	1	peu d'UMI concordants
chr1	27022940	C	-	2.01	0.0000	2	0.247	3	peu d'UMI concordants
chr1	27100878	A	G	0.84	0.0000	4	0.249	1	peu d'UMI concordants
chr1	27101505	A	G	0.59	0.0010	3	0.285	1	peu d'UMI concordants
chr1	27105931	G	-	1.16	0.0000	4	0.042	7	peu d'UMI concordants
chr1	27107177	T	C	1.1	0.0000	3	0.245	1	peu d'UMI concordants
chr22	41513477	A	G	0.67	0.0000	1	0.395	1	peu d'UMI concordants
chr22	41542758	A	G	0.87	0.0001	3	0.296	1	peu d'UMI concordants
chr22	41546100	T	C	0.74	0.0000	3	0.869	2	peu d'UMI concordants
chr22	41568532	A	G	0.88	0.0000	4	0.392	1	peu d'UMI concordants
chr2	136872393	A	-	2.74	0.0000	6	0.279	8	région homopolymérique (>7)
chr2	136872415	A	-	3.73	0.0000	7	0.154	8	région homopolymérique (>7)
chr2	136872447	A	G	1.16	0.0000	2	0.268	1	peu d'UMI concordants
chr6	106543510	A	G	0.75	0.0005	1	0.464	3	peu d'UMI concordants
chr6	106543552	T	C	0.81	0.0001	2	0.224	1	peu d'UMI concordants
chr6	106543557	T	C	0.75	0.0005	3	0.472	1	peu d'UMI concordants
chr6	106554854	A	G	0.58	0.0030	2	0.169	3	peu d'UMI concordants
chr6	106555263	T	C	1.04	0.0000	4	1.052	1	peu d'UMI concordants ; biais de brin
chr6	106555266	T	C	0.53	0.0003	3	0.093	1	peu d'UMI concordants
chr6	138198200	T	-	2.09	0.0000	5	1.057	7	biais de brin
chr6	37139203	G	A	0.73	0.0005	2	0.157	1	peu d'UMI concordants
chr6	393334	A	G	1.1	0.0000	2	2.366	1	peu d'UMI concordants ; biais de brin
chr6	41905081	T	C	0.57	0.0008	2	0.325	1	peu d'UMI concordants
chr6	41905108	T	C	0.55	0.0008	1	0.068	2	peu d'UMI concordants
chr7	2984211	A	G	0.91	0.0000	3	0.049	2	peu d'UMI concordants
chr8	128750605	C	-	1.56	0.0000	2	0.782	2	peu d'UMI concordants
chr8	128751091	T	C	0.52	0.0021	1	0.625	2	peu d'UMI concordants
chr9	139391515	A	G	0.66	0.0066	2	1.937	1	peu d'UMI concordants ; biais de brin
chr9	22003097	T	C	0.98	0.0000	4	0.598	2	peu d'UMI concordants
chr9	22003749	T	TG	2.15	0.0000	0	0.157	8	aucun UMI concordant
DeepSNVMiner (124)									
chr12	57498360	T	C	0.47	0.0089	2	0.574	2	peu d'UMI concordants
chr16	10972907	A	G	0.68	0.0660	0	0.474	1	q-value > 0.05 ; aucun UMI concordant
chr16	10992551	A	T	0.17	0.6287	0	0.566	1	q-value > 0.05
chr16	10992862	G	A	0.37	0.0660	1	0.16	1	q-value > 0.05 ; peu d'UMI concordants
chr16	10995974	G	A	0.25	0.3586	1	0.146	1	q-value > 0.05
chr16	10996570	G	A	0.4	0.1702	2	0.349	1	q-value > 0.05
chr16	11000437	T	G	0.09	1.2000	0	0.133	1	q-value > 0.05
chr16	11000463	A	T	0.1	1.2000	0	1.546	1	q-value > 0.05
chr16	11001169	A	G	0.33	0.0579	2	0.262	1	q-value > 0.05 ; peu d'UMI concordants
chr16	11001205	A	G	0.39	0.0631	1	0.014	1	q-value > 0.05 ; peu d'UMI concordants
chr16	11002865	T	C	0.23	0.8723	1	0.116	1	q-value > 0.05
chr16	11002973	C	T	0.24	0.1767	1	0.304	1	q-value > 0.05
chr16	11004079	C	T	0.4	0.2767	1	1.443	1	q-value > 0.05
chr16	11015993	A	G	0.47	0.1767	1	0.81	2	q-value > 0.05
chr16	11348853	C	T	0.37	0.1767	2	0.633	1	q-value > 0.05
chr16	3777767	G	A	0.38	0.7372	2	0.029	1	q-value > 0.05
chr16	3779318	T	C	0.4	0.9092	0	1.09	2	q-value > 0.05
chr16	3781727	T	C	1	0.0000	1	0.36	3	peu d'UMI concordants
chr16	3786057	C	G	0.34	0.0660	1	0.365	1	q-value > 0.05 ; peu d'UMI concordants
chr16	3807853	A	G	0.46	0.0239	1	0.197	2	peu d'UMI concordants
chr16	3808052	T	TA	12.52	0.0000	0	0.098	13	aucun UMI concordant
chr16	3808863	C	T	0.45	0.3586	1	0.069	2	q-value > 0.05
chr16	3808952	C	T	0.31	0.9791	0	0.642	1	q-value > 0.05
chr16	3819211	G	A	0.54	0.4465	1	0.072	3	q-value > 0.05
chr16	3820890	T	C	0.53	0.0030	4	1.108	1	peu d'UMI concordants ; biais de brin
chr16	3843674	C	T	0.15	0.7372	0	0.479	2	q-value > 0.05
chr16	3860778	C	T	0.28	0.2767	1	0.453	3	q-value > 0.05

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
chr16	3900422	G	A	0.58	0.1217	1	0.392	2	q-value > 0.05
chr16	81942116	C	T	0.75	0.0000	2	1.889	1	peu d'UMI concordants ; biais de brin
chr16	81944190	A	G	0.3	0.8723	1	1.154	1	q-value > 0.05
chr17	62006627	C	T	0.42	0.0660	1	1.347	4	q-value > 0.05 ; biais de brin
chr17	7572980	T	G	10.27	0.0000	1	3.433	1	peu d'UMI concordants ; biais de brin
chr17	7576610	A	G	0.62	0.0000	2	0.019	1	peu d'UMI concordants
chr17	7577007	C	CT	0.72	0.0000	0	1.377	1	aucun UMI concordant
chr17	7577602	A	G	0.3	0.1702	0	0.612	1	q-value > 0.05
chr17	7578467	T	G	0.34	0.8723	0	1.411	1	q-value > 0.05
chr17	7578480	T	G	0.04	1.2000	0	1.873	1	q-value > 0.05
chr17	7579478	G	A	0.45	0.1615	1	0.024	1	q-value > 0.05
chr18	60985782	G	A	0.04	1.2000	0	0.094	5	q-value > 0.05
chr19	19256609	C	T	0.48	0.1969	1	0.448	2	q-value > 0.05
chr19	19261606	T	C	0.43	0.1129	1	0.244	1	q-value > 0.05
chr1	120458053	T	C	0.38	0.1129	0	0.856	1	q-value > 0.05
chr1	120458396	G	A	0.33	0.2767	1	0.022	3	q-value > 0.05
chr1	120458779	G	A	0.32	0.4465	1	1.208	4	q-value > 0.05
chr1	120458988	A	G	0.38	0.0239	0	0.221	2	aucun UMI concordant
chr1	23885723	G	A	0.29	0.4465	1	0.447	2	q-value > 0.05
chr1	23885741	C	T	0.28	0.9791	1	0.382	1	q-value > 0.05
chr1	2491332	C	T	0.25	0.7372	1	0.399	1	q-value > 0.05
chr1	27023902	T	G	0.46	0.3586	0	3.716	2	q-value > 0.05
chr1	27057678	T	A	0.4	0.0162	0	0.078	1	aucun UMI concordant
chr1	27058030	C	T	0.36	0.1916	1	0.302	2	q-value > 0.05
chr1	27087479	A	C	1.75	0.0000	0	4.695	1	aucun UMI concordant
chr1	27087489	A	C	4.59	0.0000	0	4.221	1	aucun UMI concordant
chr1	27100889	G	A	0.42	0.8940	1	0.292	1	q-value > 0.05
chr1	27100985	C	T	0.44	0.1916	1	0.831	4	q-value > 0.05
chr1	27101425	A	G	0.35	0.3586	1	0.149	1	q-value > 0.05
chr1	27101668	G	A	0.12	0.8973	1	0.628	1	q-value > 0.05
chr1	27102089	T	G	9.22	0.0000	2	1.349	1	peu d'UMI concordants ; biais de brin
chr1	27102153	C	T	0.41	0.6287	1	1.246	2	q-value > 0.05
chr1	27105823	C	A	0.24	0.7372	0	0.774	1	q-value > 0.05
chr1	27106538	G	A	0.3	0.4465	1	0.633	3	q-value > 0.05
chr1	27106707	G	A	0.53	0.0000	2	0.766	3	peu d'UMI concordants
chr22	41545907	C	T	0.3	0.9791	1	0.254	4	q-value > 0.05
chr22	41551014	A	G	3.66	0.0000	1	1.389	2	peu d'UMI concordants ; biais de brin
chr22	41551094	G	A	0.18	0.9092	0	0.753	2	q-value > 0.05
chr22	41566523	A	G	0.35	0.4465	2	0.803	1	q-value > 0.05
chr22	41573090	A	G	0.32	0.8723	1	0.525	1	q-value > 0.05
chr22	41573260	T	C	0.61	0.2820	0	1.145	2	q-value > 0.05
chr22	41573873	A	G	0.79	0.0030	0	0.915	4	aucun UMI concordant
chr22	41574040	G	A	0.39	0.2767	1	0.127	4	q-value > 0.05
chr22	41574697	T	C	0.49	0.0021	2	0.083	2	peu d'UMI concordants
chr22	41574720	T	C	0.41	0.0239	2	0.32	2	peu d'UMI concordants
chr22	41574855	A	G	0.24	0.4465	0	0.75	1	q-value > 0.05
chr22	41574876	T	C	0.51	0.0030	2	0.717	1	peu d'UMI concordants
chr2	136872612	A	G	0.27	0.1767	1	0.512	2	q-value > 0.05
chr2	136873078	G	A	0.46	0.1969	1	0.422	1	q-value > 0.05
chr6	106534500	T	C	0.23	0.4465	0	0.188	2	q-value > 0.05
chr6	106553522	T	C	0.41	0.0239	1	1.394	2	peu d'UMI concordants ; biais de brin
chr6	138192590	A	C	0.06	1.2000	0	0.801	1	q-value > 0.05
chr6	138199599	C	T	2.19	0.0000	1	0.684	1	peu d'UMI concordants
chr6	37138576	A	G	0.42	0.0660	0	0.751	1	q-value > 0.05 ; aucun UMI concordant
chr6	37138702	C	T	0.12	1.2000	1	1.217	1	q-value > 0.05
chr6	37139145	G	A	0.49	0.2767	2	0.136	4	q-value > 0.05
chr6	37139148	T	C	0.42	0.1767	1	0.64	1	q-value > 0.05
chr6	393146	G	A	0.24	0.8973	1	1.479	1	q-value > 0.05
chr6	393248	C	T	0.51	0.4465	0	0.212	1	q-value > 0.05
chr6	395869	G	A	0.31	0.2767	0	0.193	2	q-value > 0.05
chr6	397047	C	T	0.68	0.0631	0	1.723	1	q-value > 0.05 ; aucun UMI concordant
chr6	398995	G	A	0.58	0.0079	2	0.101	1	peu d'UMI concordants
chr6	407522	G	C	0.25	0.8922	1	1.919	4	q-value > 0.05
chr6	41908136	T	G	0.28	1.2000	0	2.859	1	q-value > 0.05
chr6	41908140	T	G	0.49	0.0239	0	2.585	1	aucun UMI concordant
chr6	41908182	G	A	0.84	0.0239	3	0.317	1	peu d'UMI concordants
chr6	41908358	A	G	0.49	0.1504	0	0.636	1	q-value > 0.05
chr6	41909316	G	A	0.43	0.3586	1	0.421	2	q-value > 0.05
chr7	148506480	A	G	0.49	0.0032	1	0.368	4	peu d'UMI concordants
chr7	2977643	C	T	0.36	0.1504	2	0.389	2	q-value > 0.05
chr7	2983876	C	T	0.26	1.2000	0	0.136	2	q-value > 0.05
chr7	2983942	C	T	0.13	1.2000	1	0.031	2	q-value > 0.05
chr7	2984018	A	G	0.72	0.0000	0	0.153	1	aucun UMI concordant
chr8	128750531	C	T	0.36	0.1767	1	0.285	2	q-value > 0.05
chr8	128750598	C	T	0.24	0.7372	0	0.05	1	q-value > 0.05
chr8	128750680	A	C	2.9	0.0000	0	3.59	1	aucun UMI concordant
chr8	128750911	T	C	0.3	0.1969	1	0.27	1	q-value > 0.05
chr8	128752793	C	T	0.47	0.0239	3	0.252	2	peu d'UMI concordants
chr8	128753155	G	A	0.41	0.6287	1	0.2	1	q-value > 0.05
chr9	139390675	C	T	0.5	0.0010	2	0.3	1	peu d'UMI concordants
chr9	139391414	A	C	0.25	0.6287	0	2.673	1	q-value > 0.05
chr9	21967806	G	C	0.23	0.1767	1	1.068	1	q-value > 0.05
chr9	21974485	T	A	0.13	1.2000	0	0.804	2	q-value > 0.05
chr9	21975032	G	A	0.49	0.6287	3	0.639	2	q-value > 0.05
chr9	21994220	T	C	2.05	0.0000	1	3.537	1	peu d'UMI concordants ; biais de brin
chr9	21994235	C	T	0.19	1.2000	1	0.645	5	q-value > 0.05
chr9	21994251	A	G	0.21	0.6287	0	0.659	1	q-value > 0.05
chr9	22002874	A	G	0.36	0.0660	1	0.17	5	q-value > 0.05 ; peu d'UMI concordants
chr9	22003388	C	T	0.52	0.0003	2	0.024	2	peu d'UMI concordants
chr9	22004708	A	G	0.26	0.6682	0	0.271	1	q-value > 0.05
chr9	22005226	G	A	0.28	1.2000	3	0.73	3	q-value > 0.05
chr9	22005611	A	G	0.4	0.0239	1	0.279	1	peu d'UMI concordants
chr9	22005653	A	T	0.16	0.9791	1	0.708	1	q-value > 0.05
chr9	22005799	A	G	0.42	0.0008	2	0.079	2	peu d'UMI concordants
chr9	22005927	T	G	0.27	1.2000	0	0.532	1	q-value > 0.05
chr9	22006049	G	A	0.49	0.0188	1	0.465	1	peu d'UMI concordants
chr9	22006150	G	A	0.27	0.0784	3	0.64	1	q-value > 0.05 ; peu d'UMI concordants

Chromosome	Position	Ref	Alt	VAF(%)	Q-value	n_concordant_UMI	SB	HP	Commentaires
UMI-VarCal (62)									
chr12	57493849	T	C	0.56	0.0003	5	0.544	1	confiance = forte (4/5)
chr12	57499053	T	C	0.74	0.0000	5	0.299	1	confiance = élevée (3/5)
chr12	57499084	T	C	0.67	0.0001	5	0.593	2	confiance = forte (4/5)
chr13	41134681	C	T	1.44	0.0000	5	0.824	1	confiance = élevée (3/5)
chr15	45003722	A	G	0.86	0.0002	5	0.246	2	confiance = forte (4/5)
chr15	45003782	T	C	0.7	0.0000	5	0.248	1	confiance = élevée (3/5)
chr15	45003784	T	C	0.56	0.0008	5	0.306	1	confiance = forte (4/5)
chr15	45007838	A	G	0.7	0.0000	5	0.015	5	confiance = forte (4/5)
chr16	10996566	T	C	0.59	0.0003	5	0.601	1	confiance = élevée (3/5)
chr16	10997677	A	G	0.4	0.0188	5	0.135	1	confiance = moyenne (2/5)
chr16	11001157	T	C	0.5	0.0003	6	0.153	2	confiance = élevée (3/5)
chr16	11001514	T	C	0.67	0.0000	5	0.979	1	confiance = moyenne (2/5)
chr16	11003993	G	A	7.21	0.0000	5	0.787	3	confiance = élevée (3/5)
chr16	3807938	A	G	0.77	0.0001	5	0.736	2	confiance = élevée (3/5)
chr16	3820822	T	C	0.55	0.0003	5	0.463	1	confiance = élevée (3/5)
chr16	81953208	T	C	0.51	0.0089	5	0.742	1	confiance = élevée (3/5)
chr17	63010401	T	C	0.75	0.0000	5	0.404	2	confiance = élevée (3/5)
chr17	7573897	T	A	51	0.0000	266	0.052	1	confiance = certaine (5/5)
chr17	7573916	A	G	1.02	0.0000	6	0.387	1	confiance = élevée (3/5)
chr17	7576932	A	G	0.44	0.0079	5	0.071	3	confiance = élevée (3/5)
chr19	19257554	G	A	0.52	0.0089	5	0.866	2	confiance = élevée (3/5)
chr1	117087038	A	-	36.07	0.0000	118	0.064	1	confiance = certaine (5/5)
chr1	2492144	A	G	0.56	0.0026	6	0.321	1	confiance = forte (4/5)
chr1	27089446	G	C	52.32	0.0000	181	0.008	1	confiance = certaine (5/5)
chr1	27094471	A	G	1.09	0.0000	5	0.234	1	confiance = élevée (3/5)
chr1	27094476	G	A	0.84	0.0000	5	0.391	2	confiance = forte (4/5)
chr1	27101140	A	G	0.94	0.0000	5	0.945	4	confiance = moyenne (2/5)
chr1	27101679	A	G	0.48	0.0089	5	0.442	2	confiance = forte (4/5)
chr22	41513782	A	G	0.36	0.0239	5	0.031	1	confiance = élevée (3/5)
chr22	41547936	T	C	0.43	0.0079	5	0.199	2	confiance = élevée (3/5)
chr22	41560151	T	C	0.63	0.0239	5	0.444	2	confiance = élevée (3/5)
chr22	41565635	A	G	0.5	0.0188	5	0.048	1	confiance = moyenne (2/5)
chr22	41566525	A	-	1.86	0.0000	8	0.194	7	confiance = moyenne (2/5)
chr22	41572905	T	C	0.51	0.0066	5	0.289	1	confiance = élevée (3/5)
chr22	41574728	A	G	0.64	0.0000	5	0.752	1	confiance = élevée (3/5)
chr22	41574986	A	-	0.73	0.0000	5	0.067	7	confiance = moyenne (2/5)
chr2	136873239	A	G	0.55	0.0008	5	0.179	3	confiance = forte (4/5)
chr2	136873391	A	G	0.63	0.0005	5	0.517	3	confiance = forte (4/5)
chr2	61719126	A	G	0.75	0.0001	5	0.844	2	confiance = élevée (3/5)
chr2	61719182	A	G	0.51	0.0089	6	0.299	2	confiance = forte (4/5)
chr6	106554306	T	C	0.7	0.0000	5	0.026	3	confiance = forte (4/5)
chr6	138192511	A	G	0.61	0.0000	6	0.663	3	confiance = élevée (3/5)
chr6	138197210	T	A	0.76	0.0000	6	0.617	1	confiance = forte (4/5)
chr6	37139052	A	G	0.74	0.0000	5	0.461	1	confiance = élevée (3/5)
chr6	397039	C	-	65.23	0.0000	82	0.68	1	confiance = forte (4/5)
chr6	397065	A	G	0.65	0.0002	5	0.316	2	confiance = forte (4/5)
chr6	398913	T	C	0.67	0.0001	5	0.043	1	confiance = élevée (3/5)
chr6	405087	A	G	0.5	0.0010	5	0.303	1	confiance = élevée (3/5)
chr7	148508740	A	G	0.59	0.0000	7	0.135	2	confiance = élevée (3/5)
chr7	2985586	C	G	43.76	0.0000	115	0.202	2	confiance = certaine (5/5)
chr9	139390630	A	G	0.51	0.0000	5	0.404	1	confiance = élevée (3/5)
chr9	139390633	A	G	0.44	0.0003	5	0.517	1	confiance = élevée (3/5)
chr9	139391347	T	C	0.4	0.0239	6	0.229	1	confiance = élevée (3/5)
chr9	139391860	T	C	0.45	0.0110	5	0.186	1	confiance = moyenne (2/5)
chr9	21967958	A	G	0.49	0.0000	5	0.733	4	confiance = élevée (3/5)
chr9	21975066	A	G	0.57	0.0428	5	0.51	2	confiance = élevée (3/5)
chr9	22005116	T	-	3.44	0.0000	8	0.084	1	confiance = forte (4/5)
chr9	22005497	T	C	0.64	0.0000	5	0.479	4	confiance = forte (4/5)
chr9	22005641	A	G	0.81	0.0000	5	0.265	2	confiance = élevée (3/5)
chr9	22005814	T	C	0.33	0.0066	5	0.259	1	confiance = élevée (3/5)
chr9	22005912	A	G	0.58	0.0009	5	0.541	1	confiance = élevée (3/5)
chr9	22008687	A	G	0.54	0.0089	5	0.432	1	confiance = forte (4/5)

TABLE 5.7 – Liste détaillée des variants détectés par UMI-VarCal, DeepSNVMiner, SiNVICT et outLyzer dans l'échantillon Z.

q-value	UMI	UMI HP	*q-value* & UMI	UMI & SB	SB	*q-value* & SB	*q-value* & HP	UMI & HP	SB & HP	Total
79	71	15	9	13	2	1	0	0	0	190
41,58%	37,37%	7,89%	4,74%	6,84%	1,05%	0,53%	0,00%	0,00%	0,00%	

TABLE 5.8 – Nombre de discordances classées par type pour l'échantillon Z.

5.5.3.2 Données simulées

Évaluer la performance des quatre outils est un très bon moyen pour comparer le sensibilité de chacun d'eux. Cependant, vu que les vraies mutations ne sont pas connues dans les trois échantillons ci-dessus, nous ne pouvons pas conclure quant à la spécificité de ces outils. De ce fait, il est nécessaire de tester les outils sur des données simulées. Au moment où l'analyse a été réalisée, un nombre limité d'outils existaient pour simuler des *reads* avec UMI mais uniquement pour des données RNA-seq. Par contre, aucun outil ne permettait de générer des *reads* pour des données DNA-seq. Ainsi, nous avons développé UMI-Gen, un simulateur de *reads* avec UMI et l'avons utilisé pour générer deux échantillons simulés (échantillon 1 et 2). L'algorithme et le *workflow* d'UMI-Gen seront expliqués en détail dans le Chapitre 6. L'échantillon 1 présente une profondeur de 1000x et contient 13 mutations (11 substitutions à une fréquence allant de 90% à 1%, 1 insertion à 10% et une délétion à 10%). L'échantillon 2 présente une profondeur de 10 000x et contient 15 mutations (13 substitutions à une fréquence allant de 90% à 0,1%, 1 insertion à 10% et une délétion à 10%). Ensuite, nous avons lancé les analyses avec les quatre outils sur les deux échantillons pour comparer leur sensibilité et leur spécificité. Les résultats sont présentés dans la Table 5.9 pour l'échantillon 1 et dans la Table 5.10 pour l'échantillon 2.

Variant Caller	VP	FP	FN	Sensibilité (%)	Spécificité (%)
SiNVICT	8	233	5	61.5	99.7
OutLyzer	11	98	2	84.6	99.9
DeepSNVMiner	12	37	1	92.3	99.95
UMI-VarCal	13	0	0	100	100

TABLE 5.9 – Les résultats du *variant calling* sur l'échantillon 1. Quatre *variant callers* ont été testés : SiNVICT, outLyzer, DeepSNVMiner et UMI-VarCal et pour chacun d'eux, le nombre de vrais positifs (VP), faux positifs (FP), faux négatifs (FN), la sensibilité et la spécificité sont calculés.

Variant Caller	VP	FP	FN	Sensibilité (%)	Spécificité (%)
SiNVICT	8	455	7	53.4	99.4
OutLyzer	12	330	3	80	99.6
DeepSNVMiner	14	2	1	93.4	99.99
UMI-VarCal	15	0	0	100	100

TABLE 5.10 – Les résultats du *variant calling* sur l'échantillon 2. Quatre *variant callers* ont été testés : SiNVICT, outLyzer, DeepSNVMiner et UMI-VarCal et pour chacun d'eux, le nombre de vrais positifs (VP), faux positifs (FP), faux négatifs (FN), la sensibilité et la spécificité sont calculés.

Les deux échantillons ont un total de 76 630 positions séquencées, ce qui correspond à la taille du panel de séquençage. Le nombre total de positifs correspond au nombre de variants trouvés dans le fichier VCF résultat. Le nombre total de négatifs est ensuite calculé en soustrayant le nombre total de positifs du nombre total de positions (76 630). Globalement, les quatre outils ont eu des résultats relativement similaires pour les deux échantillons. En commençant par SiNVICT, il a détecté 241 variants dans l'échantillon 1 et 463 dans l'échantillon 2, mais avec le même nombre de vrais positifs. Cela correspond à des sensibilités de 61,5% et 53,4% et des spécificités de 99,7% et 99,4% pour les échantillons 1 et 2 respectivement. Passant à outLyzer, l'outil a détecté 109 variants dans l'échantillon 1

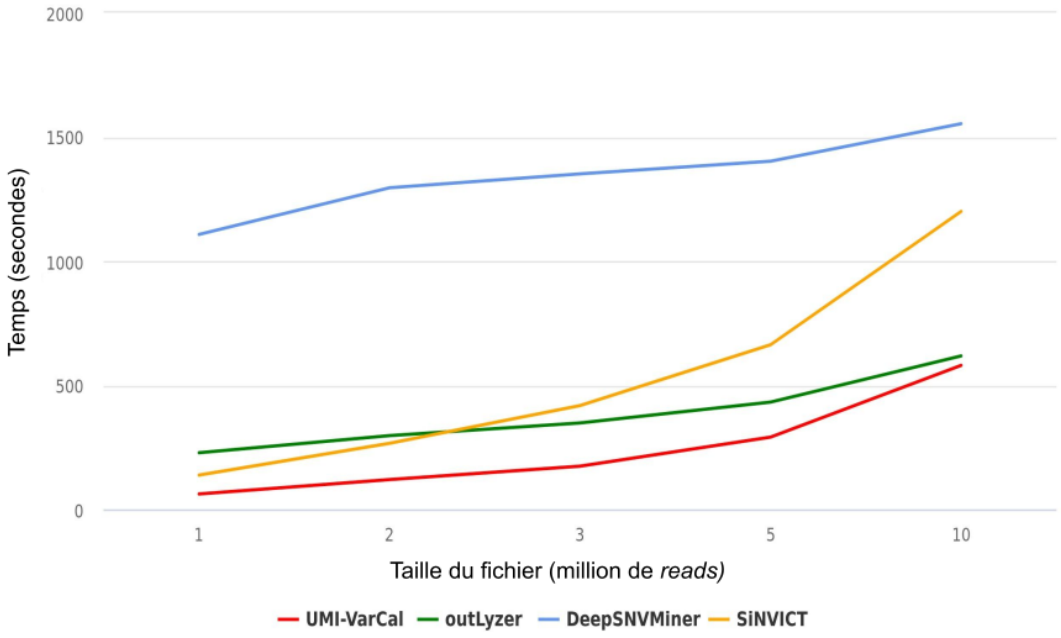


FIGURE 5.11 – Comparaison des temps d'exécution des outils UMI-VarCal, DeepSNVMiner, SiNVICT et outLyzer.

et trois fois plus de variants dans l'échantillon 2 (342). Malheureusement, parmi les 233 variants détectés en plus, un seul correspondait à un vrai positif, le reste n'étant que des faux positifs. L'outil outLyzer a obtenu de bonnes sensibilités (> 80%) et d'excellentes spécificités (99,9%/99,6%) sur les deux échantillons. En ce qui concerne DeepSNVMiner, l'outil a réussi à détecter tous les variants insérés sauf la délétion dans les deux échantillons. L'outil a obtenu des scores très élevés pour la sensibilité (92,3%/93,4%) ainsi que la spécificité (99,95%/99,99%) pour les deux ensembles de données. Enfin, UMI-VarCal a pu obtenir un score parfait (100%) en termes de sensibilité et de spécificité sur les deux échantillons en détectant tous les 13/15 variants de l'échantillon 1/2 sans aucun faux positif ni faux négatif pour les deux configurations.

5.5.3.3 Comparaison de performance

Afin de comparer les performances d'UMI-VarCal avec les trois autres *variant callers*, nous avons créé artificiellement cinq échantillons différents avec une taille croissante (1, 2, 3, 5 et 10 millions de *reads*). Cela permettra, non seulement, de comparer les performances des outils, mais aussi de voir comment les performances varient avec la taille de l'échantillon. Tous ces tests sont effectués sur un processeur à un seul cœur tournant à 2,20 GHz. Toutes les mesures ont été effectuées 3 fois et la moyenne a été utilisée pour la comparaison (Figure 5.11). Pour analyser 1 million de *reads*, UMI-VarCal est le plus rapide ayant besoin de 62 secondes pour terminer l'analyse. Il est suivi par SiNVICT qui met 138 secondes et outLyzer avec 228 secondes. L'outil le plus lent est DeepSNVMiner puisqu'il lui faut 1107 secondes pour terminer son analyse. Pour l'analyse de 2 millions de *reads*, les classements ne changent pas car UMI-VarCal est toujours le plus rapide des quatre outils et DeepSNVMiner le plus lent. De même, l'analyse de 3 millions de *reads* est la plus rapide avec UMI-VarCal et la plus lente avec DeepSNVMiner. Cependant, outLyzer arrive à battre SiNVICT qui semble avoir du retard lorsque la taille de l'échantillon augmente. Les rangs ne changent pas à la barre des 5 millions de *reads* : UMI-VarCal surpasse les trois autres logiciels et DeepSNVMiner restant le plus lent. Enfin, l'analyse de

l'échantillon de 10 millions de *reads* est encore une fois la plus rapide avec UMI-VarCal ne prenant que 580 secondes à compléter. L'outil outLyzer est de près derrière et termine l'analyse 38 secondes après UMI-VarCal (618 secondes). SiNVICT conserve la troisième place ayant besoin de 1200 secondes pour terminer son analyse. DeepSNVMiner est toujours le dernier mettant 1553 secondes pour produire les résultats finaux. Nous tenons à préciser que DeepSNVMiner est un *pipeline* d'analyse complet tandis les trois autres outils sont des *variant callers*. En tenant compte de ce fait et pour que la comparaison soit juste, les temps de DeepSNVMiner présentés dans la Figure 5.11 représentent les temps de l'étape du *variant calling* uniquement. Nous notons que UMI-VarCal et outLyzer peuvent être exécutés sur plusieurs cœurs, ce qui peut réduire considérablement le temps d'exécution, en particulier sur de très grands échantillons. En terme de consommation mémoire, UMI-VarCal n'est pas très exigeant. La consommation mémoire n'est pas seulement affectée par le nombre de *reads*, mais également par d'autres mesures tels que le facteur d'amplification de l'échantillon, la longueur des UMI et la profondeur de séquençage. Par conséquent, la mesure de la variation de la consommation mémoire avec la taille de l'échantillon n'est pas significative. Dans nos tests sur les 3 échantillons d'ADN que nous avons sélectionnés, UMI-VarCal avait besoin d'environ 5 Go de RAM.

5.6 Synthèse

Dans ce chapitre, nous avons présenté l'approche classique utilisée pour le *variant calling* et ses limites. Pour contourner ces limites, une nouvelle stratégie se basant sur les UMI est utilisée et permet de répondre à la problématique de la détection des variants à très faible fréquence (< 1%) dans des échantillons où les cellules tumorales sont minimes ou dans les biopsies liquides (*cfDNA*). Ceci nous a conduit à développer un *variant caller* appelé UMI-VarCal permettant d'utiliser efficacement les UMI pour filtrer les faux positifs. UMI-VarCal a fonctionné mieux que les trois autres outils auxquels il a été comparé. Il a pu facilement détecter les variants trouvés par les autres outils et filtrer les faux positifs grâce à ses filtres de post-traitement en plusieurs étapes (filtre d'analyse UMI, filtre de biais de brin et filtre de région homopolymérique). De plus, il a été capable de détecter un nombre élevé de variants à des VAF $\leq$ 1% et non appelés par les autres outils, dont 85% (en moyenne) ont au moins un niveau de confiance élevé. En terme de temps d'exécution, UMI-VarCal est plus rapide non seulement que les autres outils basés sur les UMI mais aussi, contre les outils adoptant l'approche de détection classique. Ces résultats démontrent comment l'approche basée sur les UMI est beaucoup plus efficace et précise que l'approche classique : elle permet de détecter des variants à des VAF aussi faibles que 0,1% sans sacrifier la spécificité ni la performance.

Chapitre 6

Simulation des reads

6.1 Introduction

Aujourd'hui, le diagnostic du cancer est un domaine de recherche très actif et l'une de ses applications les plus importantes est la détection de variants nucléotidiques ponctuels (SNV) dans les cellules tumorales. Une autre application d'une grande importance est la détection des variants structuraux (CNV) tels que les événements d'amplifications et de délétions. En effet, établir un profil précis des variants chez un patient cancéreux permet de mieux comprendre l'évolution du cancer et de personnaliser le traitement en fonction du profil établi. Ces applications sont rendues possibles grâce à des analyses de séquençage de nouvelle génération où un séquenceur se charge de produire les séquences des fragments d'ADN séquencés. Ces séquences doivent ensuite être filtrées et alignées contre un génome de référence pour pouvoir détecter les variants de type SNV ou CNV.

Avec le nombre croissant d'outils de *variant calling*, il est devenu difficile de choisir le bon logiciel adapté à une certaine expérience. La simulation de données peut jouer un rôle primordial pour tester différents outils sur un ensemble de données sur lequel nous avons le contrôle, et où la vérité est connue. C'est la différence principale avec l'utilisation des données réelles puisque dans ce cas, le vrai nombre de variants n'est pas identifié et donc des métriques comme la spécificité ne peuvent pas être calculées précisément. Dans ce qui suit, nous présenterons l'état de l'art sur les simulateurs de *reads* existants au moment de la rédaction de ce manuscrit et nous introduirons UMI-Gen, le premier simulateur de *reads* permettant de produire des *reads* avec UMI.

6.2 Simulateurs de *reads* classiques

Les simulateurs de *reads* permettent aux développeurs des nouveaux outils de tester leurs algorithmes de détection sur un ensemble de données simulé dans lequel des variants sont insérés à différentes fréquences et à différentes positions. L'utilisation de ces outils permet d'avoir un *benchmarking* très précis de la capacité de chaque outil à appeler précisément les bons variants. À l'heure actuelle, de nombreux simulateurs de *reads* courts existent. On note premièrement l'outil ART [19] dont la plus grande caractéristique est sa capacité à générer des *reads* pour les trois plateformes de séquençage majeures : Illumina, Roche/454 et SOLiD. D'autre part, les outils pIRS [128] et SInC [129] sont spécifiques à la plateforme de séquençage Illumina et donc ne peuvent pas être utilisés dans les autres cas. L'outil XS [130] est aussi capable de simuler des *reads* pour les mêmes trois plateformes prises en charge par ART mais est le seul offrant la possibilité de simuler des *reads* pour la plateforme Ion Torrent. Ces quatre simulateurs se basent un grand ensemble de données séquencées pour chaque plateforme et utilisé afin d'établir des modèles empiriques du taux d'erreur par position ou encore des modèles HMM,

surtout pour estimer le taux des insertions et des délétions. Alors que cette approche est très efficace, le problème majeur de ces outils est qu'ils ne permettent pas d'insérer des variants de type SNV dans les *reads* produits. IntSIM [131] est le seul simulateur de *reads* permettant de reproduire le bruit de fond de séquençage dans les séquences ainsi que des variants de type SNV choisis par l'utilisateur. En ce qui concerne la simulation des variants structuraux, des outils tels que SVSR [132] et VISOR [133] prennent en charge la production des *reads* avec des CNV et effectuent un très bon travail pour obtenir des profils ressemblant à ceux obtenus lors du séquençage d'échantillons réels.

6.3 Développement d'UMI-Gen

6.3.1 Introduction

Malgré le grand nombre d'outils de simulation de *reads* disponibles, aucun outil permettait de produire des *reads* artificiels avec des UMI. Ceci nous a paru étonnant vu la grande utilité que pourrait porter un tel outil pour la comparaison et pour l'évaluation des *variant callers* basés sur les UMI. Pour combler ce manque, nous avons développé UMI-Gen, un simulateur de *reads* basé sur les UMI qui peut être utilisé pour tester les *variant callers* classiques ainsi que ceux intégrant une analyse des UMI. UMI-Gen utilise plusieurs échantillons biologiques réels pour estimer le taux du bruit de fond et les scores de qualité de base à chaque position. Ensuite, il est capable de reproduire ce bruit dans les fichiers simulés. Finalement, il est capable d'insérer dans les *reads* finaux des variants somatiques ponctuels (SNV) mais également des variants structuraux (CNV). Dans ce qui suit, nous allons expliquer en détail comment fonctionne UMI-Gen, chaque étape du *workflow* et allons présenter les résultats permettant de valider notre algorithme de simulation.

6.3.2 Différence entre bruit de fond et variant somatique

Grâce à l'utilisation des UMI, nous pouvons faire la différence entre une erreur causée par le bruit de fond et un vrai variant causé par la présence d'une mutation dans le fragment d'ADN séquencé. En effet, les séquences d'UMI ajoutées aux *fragments* doivent être aléatoires afin que chaque fragment puisse avoir une séquence oligonucléotidique courte et unique, donnant à chaque fragment une sorte d'étiquette unique. Lors de l'amplification, les UMI sont amplifiés avec leurs fragments respectifs. Après le séquençage, chaque UMI peut être identifié à partir de la séquence des *reads* vu que leur emplacement est généralement connu et prévu lors de la conception de l'expérience de séquençage. La Figure 6.1 rappelle le principe de l'utilisation des UMI pour faire la distinction entre artefact et variant somatique. En fait, si un variant est une véritable mutation somatique, cela signifie qu'il a été présent sur le fragment d'ADN initial, et lorsque nous marquons ce fragment avec un UMI, nous marquons également la mutation avec ce même UMI. Par conséquent, les fragments qui résultent de l'amplification du fragment d'ADN muté seront tous marqués par le même UMI et alors devront tous porter la même mutation somatique (Figure 6.1A). En revanche, si le variant est un artefact de séquençage, cela signifie que le fragment d'ADN initial ne portait pas la variation dès le début et que cette dernière est apparue plus tard lors de l'étape de séquençage. Alors, tous les fragments résultant de l'amplification de ce fragment d'ADN devraient théoriquement être étiquetés avec le même UMI et ne devraient pas présenter la mutation. Celle-ci sera ajoutée plus tard, lors du séquençage, n'affectant ainsi que certains *reads* et créant donc des discordances dans le même groupe UMI (Figure 6.1B).

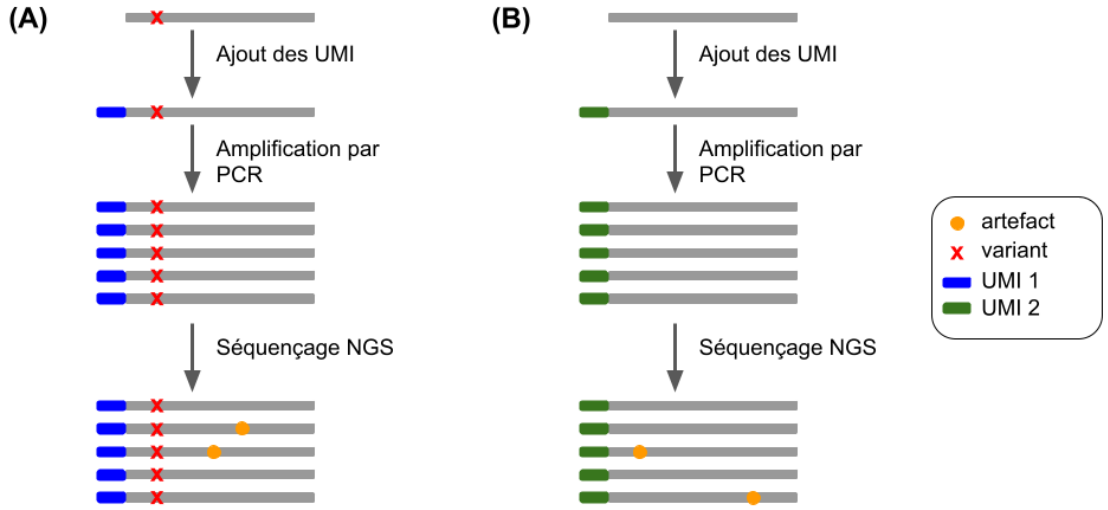


FIGURE 6.1 – La différence entre un variant somatique et un artefact d'un point de vue UMI. (A) Le variant somatique est présent initialement sur le fragment d'ADN et se retrouvera donc sur tous les fragments portant le même UMI. (B) L'artefact n'apparaît qu'après l'étape de séquençage et donc une minorité de *reads* seulement est concernée.

6.3.3 Différence entre bruit de fond et variant structural

Le problème est différent en ce qui concerne la détection des CNV. En effet, les algorithmes de détection essaient de distinguer une région amplifiée ou délétée parmi toutes les régions séquencées. Pour cela, un algorithme de segmentation est utilisé afin de créer des segments à partir des régions présentant des profondeurs similaires. Une profondeur moyenne normale est tout d'abord établie à partir d'échantillons sains et si une région de l'ADN malade présente une profondeur significativement supérieure (ou inférieure) à la profondeur moyenne, la région sera signalée comme amplifiée (ou délétée). Cependant, la difficulté de la tâche accomplie par ces outils est de distinguer entre une amplification causée par la PCR et une autre due à une amplification de la région dans l'ADN tumoral. De plus, l'amplification par PCR peut rendre une variation indétectable dans le cas d'une suramplification par PCR d'une région délétée ou d'une sous-amplification PCR d'une région amplifiée. Ceci résulte en un signal bruité et difficile à interpréter lors de la détection des CNV. Ce problème est facilement résolu grâce à l'utilisation des UMI dans les *reads*. L'idée se base sur le principe qu'une amplification tumorale par exemple est définie par une augmentation du nombre de fragments d'une région déterminée ce qui résulte en un nombre d'UMI supérieur à celui d'une zone normale présentant logiquement moins de fragments et donc moins d'UMI. En revanche, une région normale suramplifiée par PCR aura un même nombre de fragments initiaux d'une région normale et donc le même nombre d'UMI distincts mais plus de fragments avec les mêmes UMI. En résumé, la présence d'un vrai variant structural se caractérise par une modification de la profondeur d'une région ainsi que du nombre d'UMI alors que la présence d'un faux positif est soulignée par une modification de la profondeur seulement. Cette distinction est illustrée dans la Figure 6.2 qui montre un premier cas où la région étudiée est présente sur trois fragments initiaux (Figure 6.2A) alors que dans le deuxième cas, elle n'est présente que sur un seul fragment (Figure 6.2B). Les profondeurs brutes sont similaires (11/10) et ne

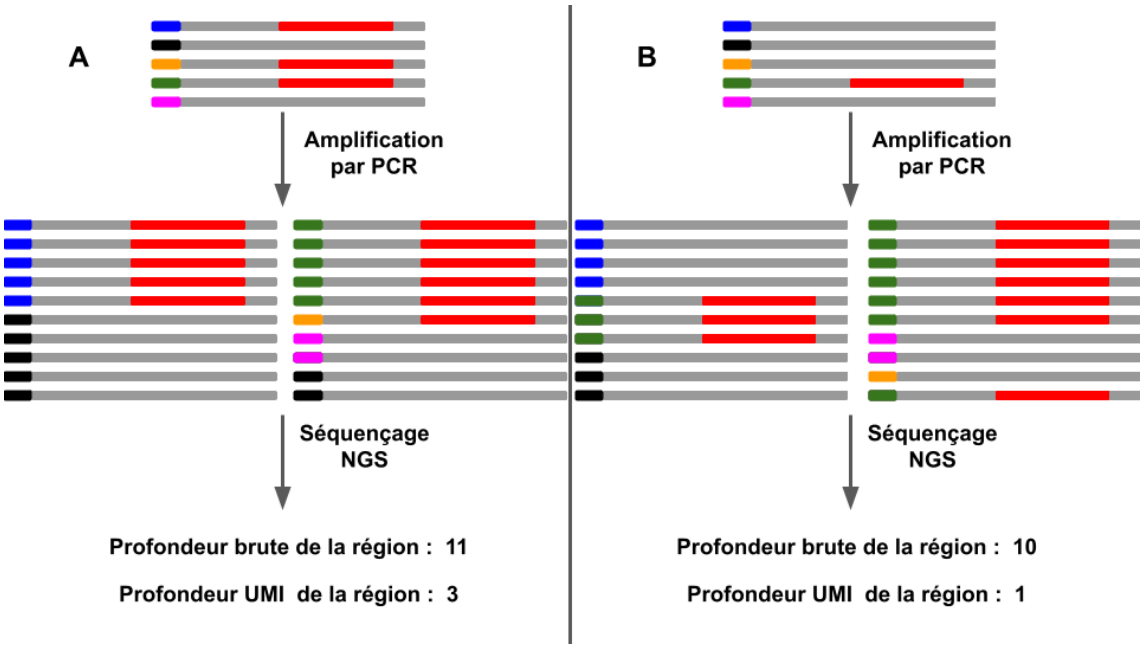


FIGURE 6.2 – La différence entre un variant structural et un artefact d'un point de vue UMI. La région en rouge est la région étudiée et les différents UMI sont marqués par des couleurs différentes. (A) La région est présente sur 3 fragments initiaux et donc marquée par trois UMI différents. La profondeur brute de la région est égale à 11. (B) La région est présente sur un seul fragment initial et donc marquée par un UMI. La profondeur brute de la région est égale à 10.

permettent pas de détecter l'amplification dans le premier cas. D'autre part, en utilisant la profondeur UMI, l'amplification est détectée puisqu'en A, trois UMI différents sont présents tandis qu'un seul est retrouvé en B.

6.3.4 Fichiers d'entrée

UMI-Gen nécessite un minimum de trois paramètres à l'exécution : une liste d'échantillons normaux BAM/SAM servant comme contrôle, le fichier BED avec les coordonnées des régions génomiques ciblées et un fichier FASTA du génome de référence. En effet, UMI-Gen est conçu pour fonctionner uniquement sur des données de séquençage ciblées, donc un fichier BED est toujours nécessaire. Comme pour UMI-VarCal, UMI-Gen peut également accepter un quatrième fichier optionnel au format PILEUP. Un cinquième fichier contenant une matrice des scores de qualité moyens calculés sur les échantillons normaux est optionnel mais peut être fourni lors de l'exécution de l'outil. Ce fichier au format MATRIX sert le même but que celui du fichier PILEUP : lors d'une première exécution, ce fichier est automatiquement produit par l'outil et en le fournissant directement lors de l'exécution, il pourra réduire le temps de l'analyse en chargeant directement les scores sans avoir besoin de les recalculer. Finalement, pour insérer des variants somatiques ou structuraux, UMI-Gen prend en charge un fichier de variants au format CSV. Le fichier contient une liste des variants que l'utilisateur souhaite insérer dans les fichiers simulés. Ce sont les seules variants qui doivent être signalés dans le fichier VCF des outils de détection lors des tests d'évaluation des *variant callers*. Le fichier de variants somatiques est composé de deux colonnes : la première contenant l'identifiant du variant écrit en suivant la nomenclature HGVS et la deuxième contenant la VAF à laquelle il sera

ajouté (Table 6.1). En ce qui concerne le fichier des variants structuraux, celui-ci est composé de cinq colonnes : les trois premières servant à préciser les coordonnées de la région à modifier en précisant le chromosome, la position du début et la position de fin respectivement. La quatrième colonne contient la fréquence de la variation et la cinquième sert à préciser son type, donc s'il s'agit d'une amplification ou d'une délétion (Table 6.2).

Colonne	Nom	Description	Type
1	ID	Identifiant du variant en nomenclature HGVS	Chaîne de caractères
2	VAF	Fréquence allélique du variant	Entier

TABLE 6.1 – Présentation des deux colonnes obligatoires d'un fichier de variants somatiques au format CSV.

Colonne	Nom	Description	Type
1	CHROM	Identifiant du chromosome	Chaîne de caractères
2	START	Position de début du variant	Entier
3	END	Position de fin du variant	Entier
4	FREQ	Fréquence du variant	Entier
5	TYPE	Type du variant (AMP/DEL)	Chaîne de caractères

TABLE 6.2 – Présentation des cinq colonnes obligatoires d'un fichier de variants structuraux au format CSV.

6.3.5 Construction du *pileup*

6.3.5.1 Construction du *pileup* initial

La construction du *pileup* est illustrée dans la Figure 6.3. La première étape consiste à générer le *pileup* initial. Pour chaque échantillon de contrôle, notre algorithme comptera les occurrences de chaque A, C, G et T. Les décomptes seront stockés pour chaque position du fichier BED ainsi que la qualité moyenne de la position et sa profondeur. Il s'agit essentiellement du même algorithme utilisé par notre *variant caller* UMI-VarCal qui a été réintégré dans cet outil pour sa grande efficacité dans le traitement des *reads* avec UMI. Un *pileup* est construit pour chaque échantillon de contrôle, et lorsqu'ils sont prêts, ils seront fusionnés dans un seul *pileup* moyen contenant les statistiques moyennes (dénombrements, profondeur et score de qualité) à chaque position en fonction des observations sur tous les échantillons de contrôle (Figure 6.3A). Une fois la fusion terminée, le résultat sera automatiquement sauvegardé sous forme de fichier PILEUP contenant toutes les informations calculées sur l'ensemble des échantillons de contrôle.

6.3.5.2 *Variant calling*

Même si théoriquement, les échantillons de contrôle ne devraient pas contenir de variations, des SNP et des mutations non détectées peuvent toujours être présents dans ces fichiers. Ces variants potentiels doivent être supprimés pour ne pas contaminer les *reads* dans les fichiers simulés. Pour ce faire, nous avons utilisé la même méthode d'appel de variants implémentée dans UMI-VarCal pour détecter des variants potentiels et les supprimer du *pileup*. Les paramètres de l'étape du *variant calling* sont toutefois plus stricts

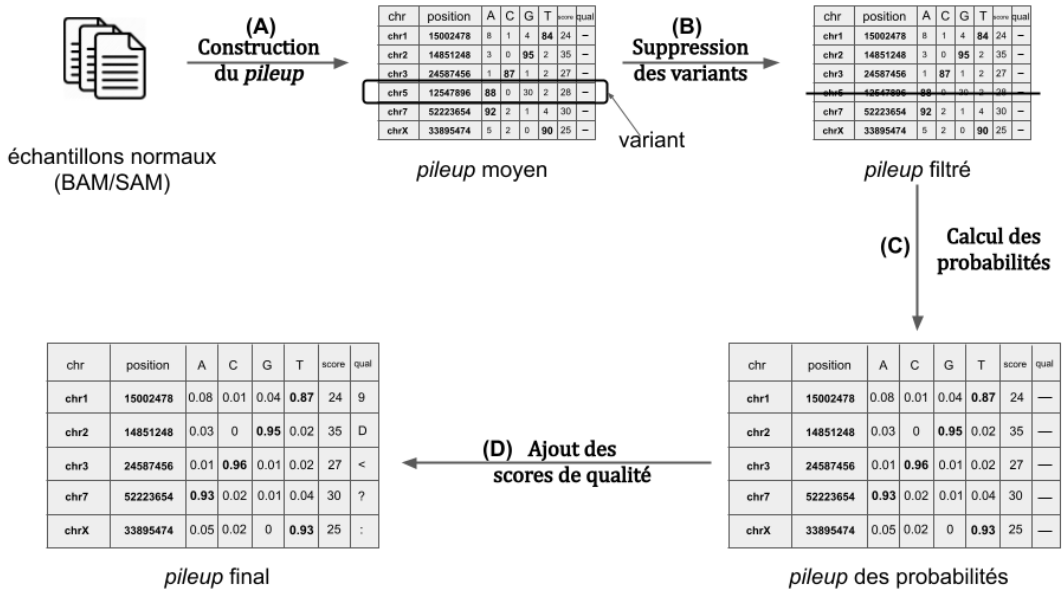


FIGURE 6.3 – Les quatre étapes nécessaires pour construire le *pileup* final à partir des échantillons de contrôle.

que ceux utilisés par défaut dans UMI-VarCal. Cette étape produira ce que nous appelons un *pileup* filtré (Figure 6.3B).

6.3.5.3 Estimation du bruit de fond

L'étape d'estimation du bruit de fond consiste à calculer la fréquence d'observation des A/C/G/T à chaque position. Sans les erreurs de bruit de fond, à chaque position, la base de référence devrait avoir une fréquence de 1 tandis que les trois bases restantes devraient être à 0. Le total des quatre fréquences doit être égal à 1. Cependant, nous savons que des artefacts existent dans nos échantillons de contrôle et ces artefacts représentent le bruit de fond que nous rencontrons normalement dans une expérience NGS normale. Puisque notre objectif est de simuler des *reads* très similaires à ceux produits lors des expériences de séquençage réelles, UMI-Gen calcule les fréquences de base réelles à partir des échantillons de contrôle à chaque position. Les fréquences seront ensuite utilisées comme matrice de probabilité lors de la production des *reads* finaux. Lorsque cette étape est terminée, un *pileup* de probabilités est généré (Figure 6.3C). Les insertions et suppressions ne sont pas prises en compte lors de l'estimation du bruit de fond et ne sont donc pas présentes dans le *pileup* final car leur occurrence a un taux beaucoup plus faible ($\sim$ 1000 fois plus faible) que celui des substitutions, en particulier dans les séquenceurs de type Illumina [134].

6.3.5.4 Estimation des scores de qualité

Notre outil a été développé sur des fichiers de séquençage produits par un séquenceur Illumina. Dans ces fichiers, les scores de qualité sont encodés sous une forme compacte, qui n'utilise qu'un octet par valeur de qualité [135, 136]. Le tableau complet du codage est disponible dans la Table 6.3. UMI-Gen n'est donc compatible qu'avec les séquenceurs utilisant le même encodage. UMI-Gen calcule le score de qualité moyen pour chaque position en fonction des qualités de tous les échantillons de contrôle, puis convertit le score de qualité en caractère ASCII correspondant et l'insère dans le fichier FASTQ final. Cette

étape constitue la dernière étape du *workflow* de la construction du *pileup* final (Figure 6.3D). De plus, UMI-Gen modélise également les scores de qualité de base par position dans les *reads* sur les échantillons de contrôle et introduit l'estimation dans les *reads* finaux. En se basant sur tous les *reads* contrôle, notre outil calcule un score de qualité de base médian pour chaque position dans le *read* afin de produire une matrice de qualité par position. Cette matrice est ensuite utilisée à la fin pour recalibrer les scores de qualité en fonction de la position de chaque base dans le *reads*. Ceci permet à UMI-Gen par exemple de reproduire la perte de qualité à la fin des *reads* lorsqu'elle est présente.

Symbole	Code ASCII	Q-Score	Symbole	Code ASCII	Q-Score
!	33	0	6	54	21
"	34	1	7	55	22
#	35	2	8	56	23
$	36	3	9	57	24
%	37	4	:	58	25
&	38	5	;	59	26
'	39	6	<	60	27
(	40	7	=	61	28
)	41	8	>	62	29
*	42	9	?	63	30
+	43	10	@	64	31
,	44	11	A	65	32
-	45	12	B	66	33
.	46	13	C	67	34
/	47	14	D	68	35
0	48	15	E	69	36
1	49	16	F	70	37
2	50	17	G	71	38
3	51	18	H	72	39
4	52	19	I	73	40
5	53	20	J	74	41

TABLE 6.3 – Tableau représentant le codage des scores de qualité dans les fichiers de séquençage produits par la plateforme Illumina.

6.3.6 Simulation des *reads* avec SNV

6.3.6.1 Production des *reads*

L'objectif principal d'UMI-Gen est de générer des *reads* pairés qui imitent ceux obtenus à partir d'expériences NGS réelles. Pour ce faire, il commence exactement de la même manière qu'une expérience réelle : obtenir les fragments d'ADN. Au début, notre outil génère un certain nombre de séquences initiales qui ne présentent que la base de référence à chaque position. L'utilisateur peut spécifier explicitement la longueur souhaitée pour tous les *reads* lors de l'exécution. Il convient de noter que l'algorithme ne créera que des *reads* qui s'alignent exactement sur les positions spécifiées dans le fichier BED, et donc l'amplification hors cible n'est pas prise en compte.

Ensuite, un UMI est attaché à chaque séquence initiale. En fonction du facteur d'amplification et de la profondeur désirée choisis par l'utilisateur, l'algorithme continuera à amplifier les séquences initiales jusqu'à ce que la profondeur désirée soit atteinte à toutes

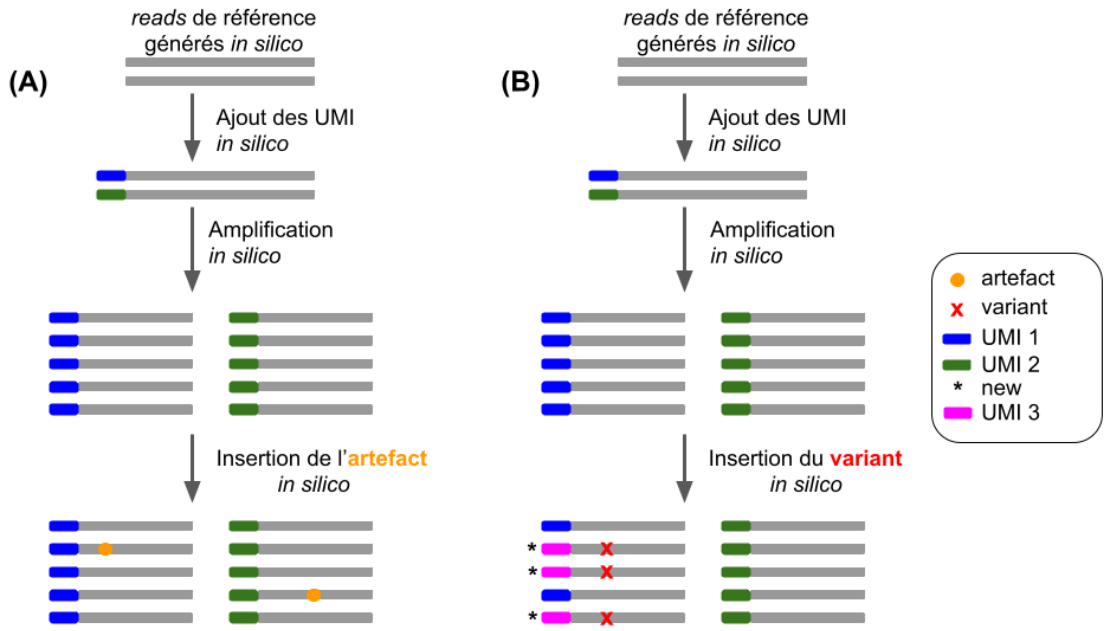


FIGURE 6.4 – LA différence entre une insertion d'un artefact (A) et une insertion d'un variant somatique (B).

les positions. En effet, à cette étape, les valeurs par défaut du facteur d'amplification et du nombre de fragments d'ADN initiaux sont automatiquement calculées afin d'assurer des performances optimales de l'outil. UMI-Gen analyse la profondeur choisie par l'utilisateur et les VAF des variants que l'utilisateur souhaite introduire et à partir de ces deux valeurs, il calcule le nombre minimum de fragments d'ADN initiaux nécessaires pour insérer un variant somatique. Même si cela garantit des performances optimales, l'utilisateur est libre de modifier ces paramètres tant qu'ils sont mathématiquement possibles. Une fois que nous avons les *reads* de référence, la deuxième étape consiste à ajouter le bruit de fond à ces *reads* (Figure 6.4A). En utilisant le *pileup* des probabilités calculé auparavant, UMI-Gen modifie les *reads* à chaque position pour qu'elles correspondent aux probabilités calculées. Ces modifications sont effectuées sans changer l'UMI des *reads* afin d'imiter les artefacts de PCR et de séquençage : ces modifications créeront donc des faux positifs qui ne doivent pas être détectés par les *variant callers*.

Enfin, UMI-Gen analyse le fichier de variants fourni par l'utilisateur afin d'insérer de vraies mutations dans les *reads* finaux. L'algorithme sélectionnera chaque position et changera la probabilité de la base alternative du variant à la fréquence correspondante à partir du fichier des variants. Dans cette étape, comme UMI-Gen ajoute un vrai variant somatique, les UMI des *reads* concernés sont également modifiés de façon à obtenir des séquences d'UMI concordantes (Figure 6.4B). Nous rappelons qu'un UMI est considéré concordant si tous les *reads* du groupe UMI portent exactement la même mutation. De plus, comme UMI-Gen génère des *reads* pairés, lors de l'ajout d'une mutation sur un *read*, le variant est automatiquement ajouté à son *mate* si les deux *reads* se chevauchent à la position du variant. Le *workflow* de la production des fichiers simulés avec des SNV est présenté Figure 6.5.

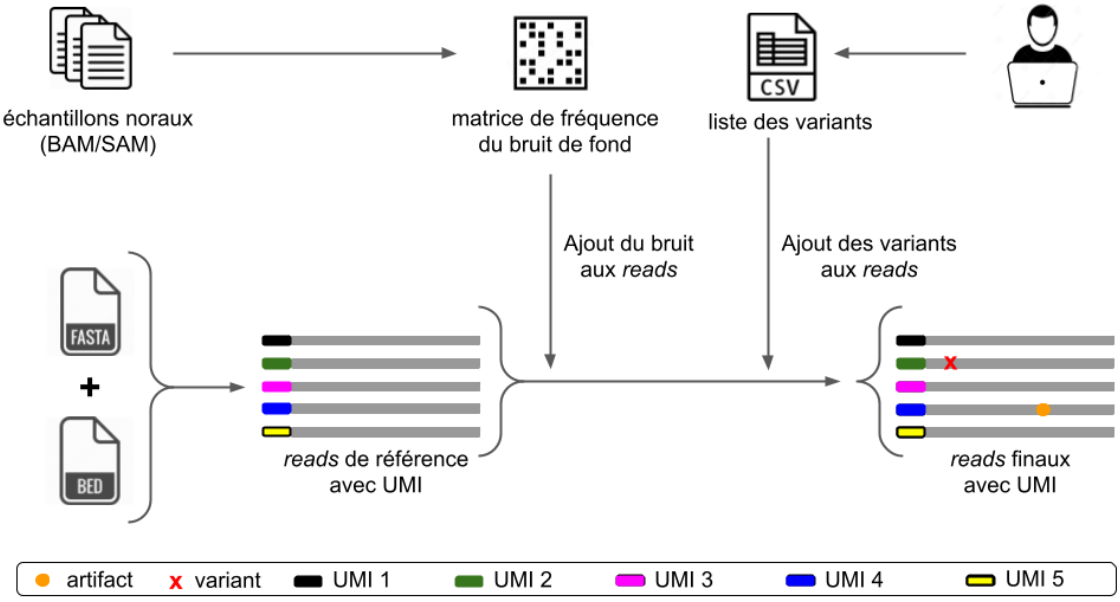


FIGURE 6.5 – Le *workflow* de la production des fichiers simulés dans lesquels des variants de type SNV ont été insérés. Figure adaptée de [137]

6.3.6.2 Résultats

Le même panel utilisé pour UMI-VarCal (76 630 bases) a été réutilisé afin de tester la capacité d'UMI-Gen à reproduire le bruit de fond moyen du séquenceur dans l'échantillon produit. Pour cela, nous avons sélectionné au hasard six échantillons d'un très grand nombre de patients dont l'ADN a été séquencé au Centre Henri Becquerel. Les six échantillons sont des biopsies liquides contenant du *cfDNA* qui a été vérifié comme étant adéquat pour le séquençage. Nous avons préféré l'utilisation de biopsies liquides car ces échantillons contiennent généralement un nombre élevé de variants et d'artefacts à très basse fréquence. L'utilisation de tels échantillons comme fichiers de contrôle produira des données simulées avec un nombre relativement élevé d'artefacts. Cela nous permettra d'avoir une estimation précise de la spécificité de chaque *variant caller* testé par la suite.

Échantillon	A	C	G	T	Profondeur totale
Contrôle 1	0	11	10	874	896
Contrôle 2	0	1	7	843	853
Contrôle 3	0	2	2	860	867
Contrôle 4	0	6	9	965	984
Contrôle 5	1	2	4	867	878
Contrôle 6	3	2	2	880	893

TABLE 6.4 – Le nombre d'observations des A, C, G, T et la profondeur totale à la position 2 493 165 du chromosome 1 pour les six échantillons de contrôle.

La Table 6.4 montre les dénombrements exacts de A, C, G, T pour la position 2 493 165 sur le chromosome 1 pour chaque échantillon de contrôle. Le premier échantillon de contrôle compte (0, 11, 10, 874), le deuxième échantillon a (0, 1, 7, 843), le troisième a (0, 2, 2, 860), le quatrième échantillon montre (1, 6, 9, 965), le cinquième a (1, 2, 4, 867) et le

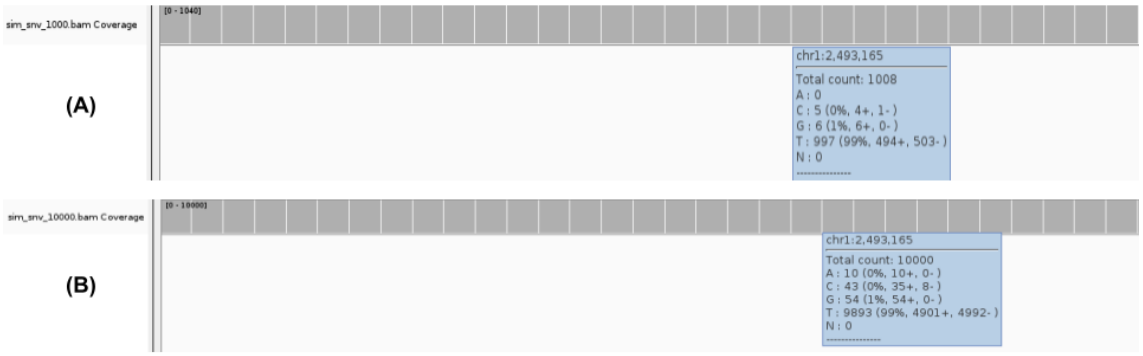


FIGURE 6.6 – Le nombre d'observations des A, C, G et T à la position 2 493 165 du chromosome 1 pour les échantillons simulés 1 et 2.

dernier compte (3, 2, 2, 880). Comme expliqué dans la Section 6.3.5.3, UMI-Gen calcule un décompte moyen pour chaque base, puis estime sa probabilité. Dans notre exemple et pour cette position, le comptage moyen obtenu présente 4 A, 24 C, 34 G et 5289 T avec une profondeur totale de 5351 bases. Pour obtenir les probabilités de cette position sur ce chromosome, il suffit simplement de diviser chaque compte de base par la profondeur, obtenant ainsi le vecteur de probabilité final (0,0007, 0,0045, 0,0064, 0,9884). Si ensuite, par exemple, nous voulons produire un fichier BAM avec une profondeur de 3000x, cette position aurait 2 A, 14 C, 19 G et 2965 T. Le *pileup* des probabilités représente essentiellement les vecteurs de probabilité de chaque position du panel, fusionnés ensemble. Dans notre test et afin de démontrer nos résultats, nous avons simulé deux échantillons artificiels dans lesquels nous avons ajouté le bruit de fond calculé. Le premier échantillon ou échantillon 1 a une profondeur moyenne de 1000x (+/- 15% à chaque position) et l'échantillon 2 a une profondeur moyenne de 10 000x. Pour nous assurer que les artefacts ont été correctement ajoutés aux *reads*, nous avons utilisé l'outil IGV (version 2.4.16) [81] pour visualiser les *reads* alignés sur le génome de référence. La Figure 6.6 montre comment le bruit de fond est correctement et très précisément ajouté à la position 2 493 165 du chromosome 1 avec les probabilités calculées à partir des six échantillons de contrôle.

Afin de valider notre jeu de données simulé, nous l'avons comparé aux échantillons de contrôle utilisés pour le générer. Tout d'abord, nous avons comparé la distribution des scores de qualité de base dans les *reads*. La Figure 6.7A montre la variation des scores de qualité de base médians avec la position de la base dans le *read* pour les échantillons de contrôle. On voit clairement que le score médian est très élevé et très stable au début et sur toute la longueur de lecture (score $\geq$ 34). Cependant, une première baisse de qualité est notée à la position 138 et une deuxième plus importante à la position 145. Dans nos données simulées, nous avons choisi une longueur moyenne pour les *reads* d'environ 110 pb et donc le *read* le plus long avait une longueur de 127 pb. La Figure 6.7B illustre comment l'algorithme recrée parfaitement la stabilité des scores tout au long des *reads* simulés. Cependant, comme ces derniers ont une longueur maximale de 127 pb, nous ne voyons pas cette petite baisse de score à la fin des *reads* simulés. Ainsi, pour être sûr

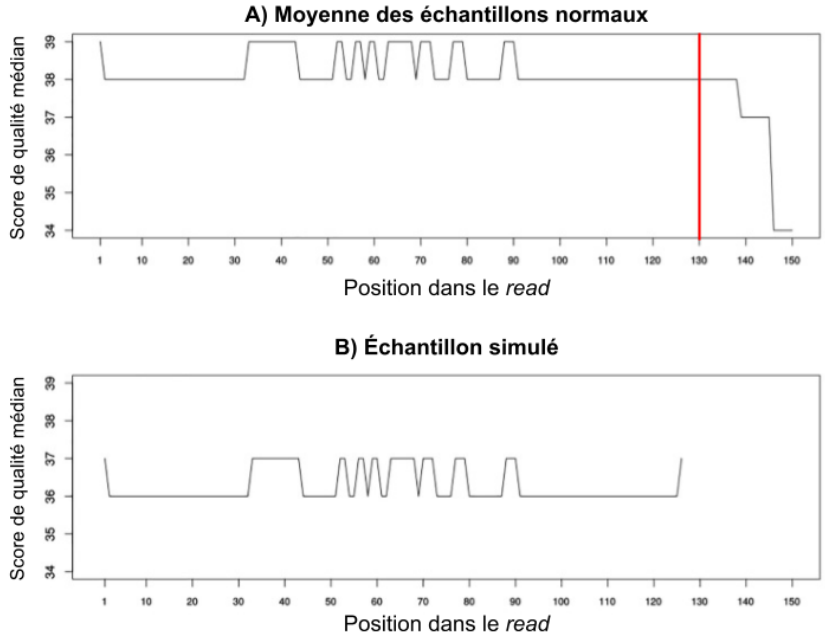


FIGURE 6.7 – La variation du score de qualité de base médian avec la position dans le *read* calculée sur les échantillons réels (A) et sur les données simulées (B)

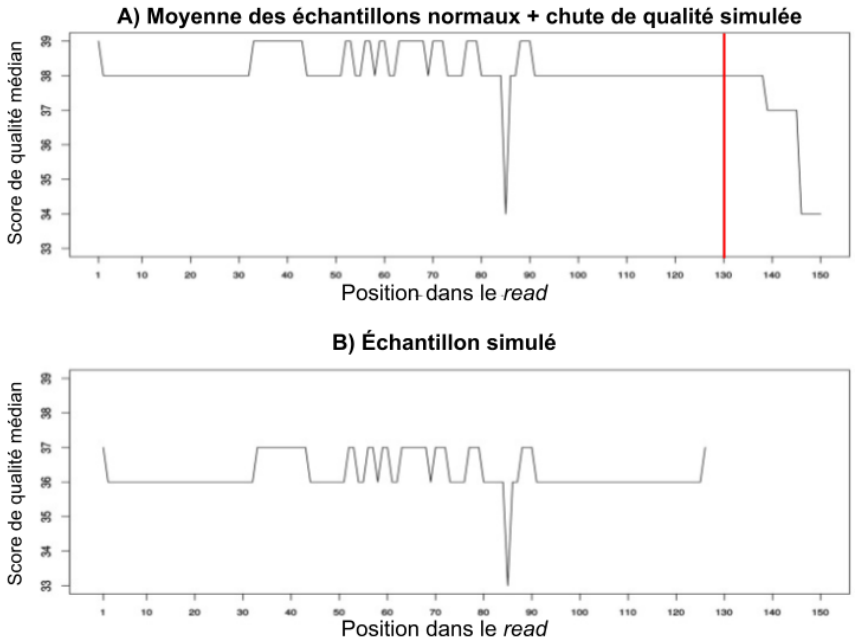


FIGURE 6.8 – La variation du score de qualité de base médian avec la position dans le *read* calculée sur les échantillons réels (A) et sur les données simulées (B). Une baisse de qualité dans les échantillons de contrôle a été simulée dans le scénario (A) et sa reproduction dans le jeu de données simulé (B).

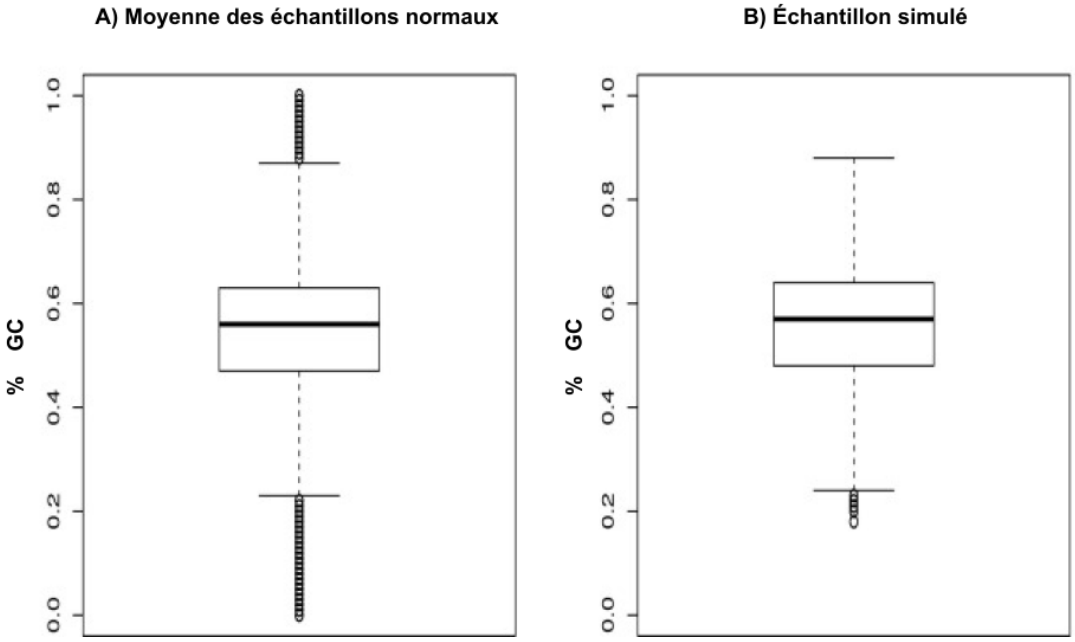


FIGURE 6.9 – La répartition du % GC des *reads* dans les données réelles (A) et dans les données simulées (B).

que notre estimation du score de qualité fonctionne correctement, nous avons simulé une baisse de qualité à la position 85 et avons voulu voir si elle sera insérée dans les *reads* simulés. La Figure 6.8 montre comment la baisse de qualité simulée (38 $\rightarrow$ 34) à la position 85 a été parfaitement reproduite dans les données simulées (36 $\rightarrow$ 33). Un autre paramètre que nous voulions vérifier est la variation en % GC entre le contrôle et les données simulées. La Figure 6.9 montre clairement comment le % GC médian des lectures dans les données de contrôle (Figure 6.9A - GC : 56%) est presque identique à celui des *reads* simulés (Figure 6.9B - GC : 57%).

Finalement, deux listes différentes de mutations ont été créées pour produire chaque échantillon simulé. La première liste contient 11 variants de substitution avec des fréquences allant de 0,9 (90%) à 0,01 (1%), une délétion à 10% et une insertion à 10%. Cette liste est utilisée pour produire l'échantillon simulé 1 avec une profondeur de 1000x. La deuxième liste contient 13 variants de substitution avec des fréquences allant de 0,9 (90%) à 0,001 (0,1%), une délétion à 10% et une insertion à 10%. Cette liste est utilisée pour produire l'échantillon simulé 2 avec une profondeur de 10 000x. Deux variants de très basse fréquence (fréquence < 1%) ont été ajoutés à la deuxième liste pour tester la précision d'insertion des variants somatiques d'UMI-Gen. Afin de vérifier que les variants choisis ont été ajoutés aux positions exactes avec les fréquences correctes, nous avons utilisé IGV pour visualiser les *reads*. La Table 6.5 présente la liste des variants ajoutés dans les deux échantillons. Les séquenceurs de nouvelle génération ont des difficultés à détecter avec précision des variants dans de longues régions d'homopolymères. De ce fait, certains *variant callers* filtrent automatiquement les variants qui se produisent dans ces régions. Afin d'éviter tout biais, nous avons soigneusement choisi l'emplacement de chaque variant pour nous assurer qu'il n'est pas inséré dans une région homopolymérique. La Figure 6.10 illustre les résultats de visualisation de quatre positions mutées par IGV et met en évidence la capacité d'UMI-Gen d'ajouter avec précision des variants dans les *reads* aux positions choisies avec les bonnes fréquences. Ces deux fichiers ont ensuite été utilisés pour tester les quatre différents *variant callers*. Les résultats ont été présentés dans la Section 5.5.3.2 (Chapitre 5).

FIGURE 6.10 – Les mutations insérées ont été correctement ajoutées aux *reads*, chacune à sa bonne position et avec la fréquence correspondante. Ici, nous voyons quatre mutations : chr1 :2491260A>G à 70%, chr1 :27022900C>A à 20%, chr1 :120458000C>CTA à 10% et chr1 :27093001G>A à 5%.

Position	Allèle de référence	Allèle alternatif	Fréquence	Échantillon
2 488 101	G	A	0,9	E1 & E2
2 489 200	C	A	0,8	E1 & E2
2 491 260	A	G	0,7	E1 & E2
2 493 201	T	A	0,6	E1 & E2
2 494 300	G	A	0,5	E1 & E2
23 885 600	C	A	0,4	E1 & E2
23 885 800	A	T	0,3	E1 & E2
27 022 900	C	A	0,2	E1 & E2
27 023 200	C	A	0,1	E1 & E2
27 093 001	G	A	0,05	E1 & E2
27 100 350	C	A	0,01	E1 & E2
27 106 500	G	A	0,005	**E2**
117 057 400	T	A	0,001	**E2**
120 458 000	C	CTA	0,1	E1 & E2
120 466 600	TGTC	T	0,1	E1 & E2

TABLE 6.5 – Liste détaillée des mutations insérées par UMI-Gen. Dans ce test, toutes les mutations sont insérées sur le chromosome 1.

6.3.7 Simulation des *reads* avec CNV

6.3.7.1 Production des *reads*

UMI-Gen est capable de simuler des *reads* avec des SNV ou des CNV. La méthode d'insertion des CNV ressemble à celle utilisée pour les SNV mais quelques différences existent puisque par exemple, la notion du bruit de fond n'est pas tout à fait pareille. Un autre exemple peut être la séquence des *reads* : les erreurs de séquençage importent peu dans ce genre d'applications puisque nous nous intéressons à des régions couvertes plutôt qu'aux séquences des *reads*. De ce fait, la notion de bruit de fond est différente pour les CNV : en effet, le bruit de fond se présente sous forme d'un changement de profondeur d'une certaine région sans que le nombre de séquences initiales d'ADN couvrant cette région ait changé. Ainsi, UMI-Gen commence toujours par produire des *reads* de référence parfaitement identiques à ceux produits lors d'une simulation de SNV. Ensuite, des séquences d'UMI sont aléatoirement générées et ajoutées au début des *reads*. Pour insérer le bruit de fond, UMI-Gen calcule d'abord la profondeur moyenne sur la totalité des régions pour les six échantillons de contrôle (les mêmes que ceux utilisés dans la partie SNV). Ensuite, toujours à partir des fichiers de contrôle, il calcule le ratio entre la profondeur de la région en question et la profondeur moyenne. Un ratio est ainsi spécifique à chaque région et en multipliant ce ratio par la profondeur désirée du fichier final (paramètre d'UMI-Gen modifiable par l'utilisateur), nous obtiendrons la profondeur finale d'une région. Par exemple, si la profondeur moyenne des échantillons de contrôle est de 2000x, la région i a une profondeur moyenne de 1700x et la profondeur désirée est 5000x, le ratio est calculé d'abord en divisant 1700 par 2000 et la profondeur de la région i dans le fichier simulé serait le produit du ratio (1700/2000) par 5000 ce qui nous donne 4250. De cette façon, le bruit est reproduit pour chaque région séquencée du panel. En ce qui concerne les UMI, même si la profondeur de la région est inférieure à la profondeur moyenne, son nombre d'UMI reste inchangé. En revanche, si la région concernée est une région où l'utilisateur souhaite insérer un variant structural, la profondeur de la région est calculée en multipliant la profondeur désirée du fichier simulé par la fréquence du variant. De même pour les UMI, leur nombre pour cette région est proportionnel à la

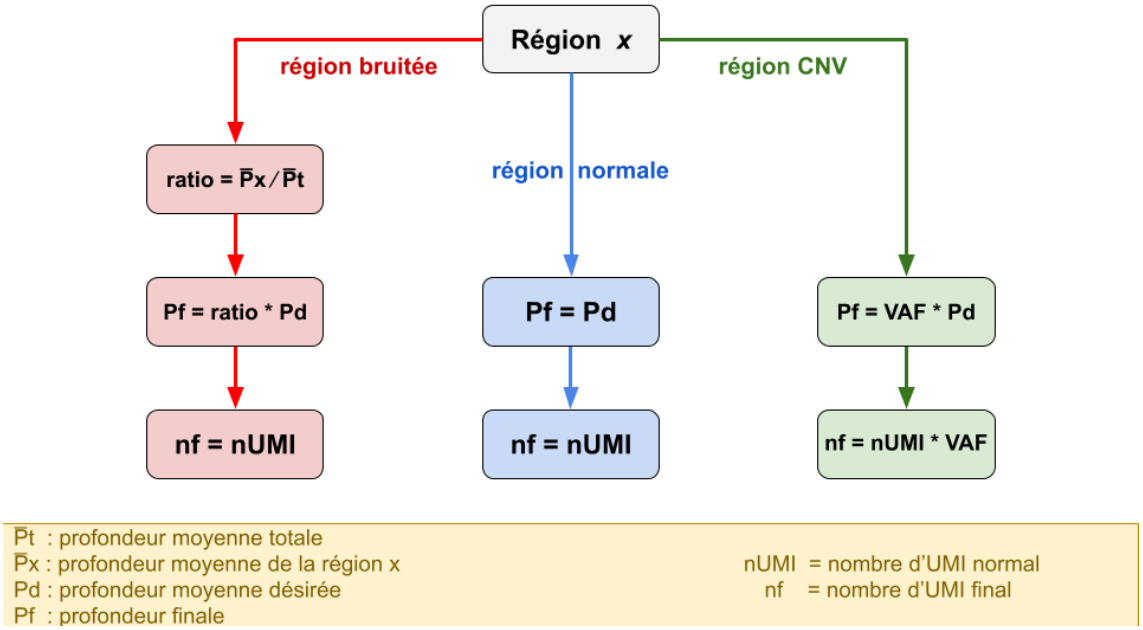


FIGURE 6.11 – La méthode de calcul d'UMI-Gen pour les trois types de région : une région bruitée, une région normale et une région CNV.

profondeur. Par exemple, si la profondeur désirée du fichier est de 5000x et l'utilisateur souhaite insérer une amplification de 30% pour la région j, sa profondeur et son nombre d'UMI seront augmentés de 30% (profondeur finale de la région j = 6500x). La méthode de calcul des profondeurs et du nombre d'UMI final appliquée par UMI-Gen pour les différentes régions est illustrée dans la Figure 6.11.

6.3.7.2 Résultats

Pour valider notre méthode de simulation de CNV, nous avons utilisé UMI-Gen pour produire un fichier BAM ayant une profondeur moyenne de 1000x. Pour ce faire, nous avons fourni un fichier de variants structuraux contenant une amplification et deux délétions. L'amplification est ajoutée sur le chromosome 1 et s'étend sur une région commençant à la position 2 491 000 jusqu'à la position 2 492 000. De même, la première délétion se situe sur le chromosome 1 de la position 27 020 000 jusqu'à 27 030 000. Enfin, le dernier CNV est une délétion ajoutée sur le chromosome 6 et s'étend sur une région de 600 pb (de la position 106 534 000 jusqu'à 106 534 600). Les trois variants ont été insérés à une fréquence de 0,3 (30%). Les comptages en UMI et en profondeur brute ont été réalisés sur les trois régions modifiées et les résultats sont présentés dans les Figures 6.12, 6.13 et 6.14. La Figure 6.12 présente en même temps l'amplification insérée sur le chromosome 1 (indiquée en vert) mais aussi un faux variant présent dans une région situé en amont (indiquée en rouge). Le nombre d'UMI est relativement stable tout au long du chromosome ayant une moyenne de 71 UMI. Dans le cas du faux variant (en rouge), on voit clairement que la profondeur de la région est supérieure à la profondeur moyenne (1400 contre 1000 en moyenne) mais que le nombre d'UMI reste stable à 70-71. Ceci veut dire que la région a été suramplifiée lors de la PCR mais le nombre de molécules initiales couvrant cette région est toujours le même. En revanche, en ce qui concerne l'amplification insérée par UMI-Gen à 30% (en vert), on remarque que la profondeur de la région a augmenté de 30% et est donc passée à 1300 et que, de même, le nombre d'UMI est passé de 71 à 93. Cette variation simultanée de profondeur brute et du nombre d'UMI est ce qui

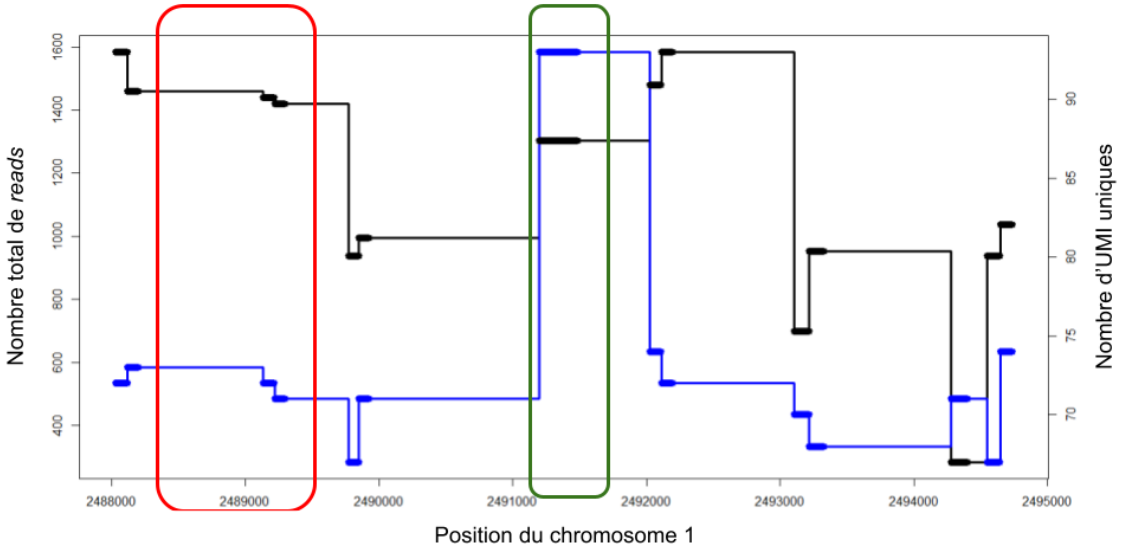


FIGURE 6.12 – La variation de la profondeur brute et du nombre d'UMI sur une région contenant un faux variant (indiqué en rouge) et une véritable amplification (indiquée en vert).

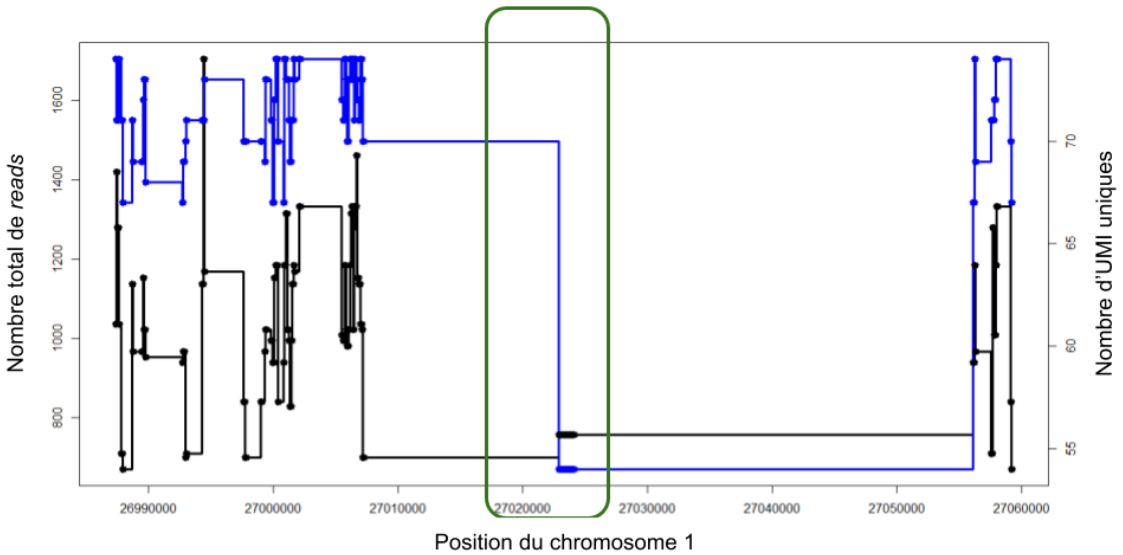


FIGURE 6.13 – La variation de la profondeur brute et du nombre d'UMI sur une région contenant un faux variant (indiqué en rouge) et une véritable délétion (indiquée en vert).

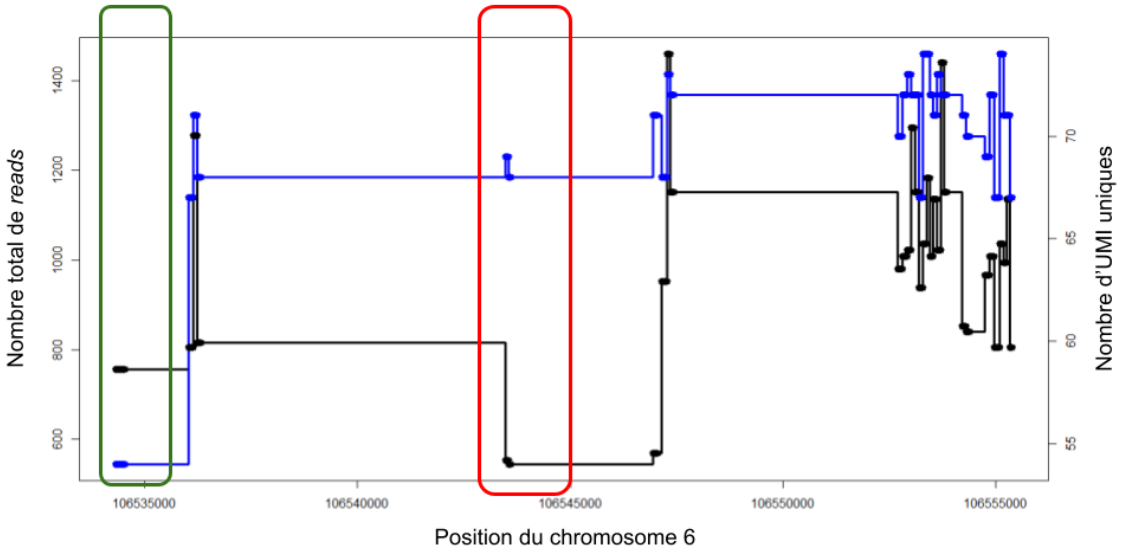


FIGURE 6.14 – La variation de la profondeur brute et du nombre d'UMI sur une région contenant un faux variant (indiqué en rouge) et une véritable délétion (indiquée en vert).

caractérise un vrai CNV. La Figure 6.13 présente la délétion insérée sur le chromosome 1 par UMI-Gen. De même, on voit que l'insertion du variant a provoqué une chute de profondeur ainsi qu'une chute du nombre d'UMI qui diminuent d'environ 30%. De la même façon, la Figure 6.14 présente le cas de deux délétions proches l'une de l'autre, la première (en vert) correspondant au variant inséré par UMI-Gen et caractérisée par une chute simultanée de la profondeur brute et du nombre d'UMI, et la deuxième (indiquée en rouge) correspondant à un faux variant vu que seule la profondeur brute de la région est diminuée.

Finalement, nous avons utilisé UMI-Gen pour simuler un échantillon avec une profondeur de 1000x et en utilisant les mêmes fichiers de contrôle. Dans ce fichier, les régions amplifiées et délétées restent relativement aux mêmes coordonnées que celles utilisées précédemment mais dans des fenêtres plus larges. En effet, cet échantillon est produit pour être analysé avec un outil de détection de CNV. Ces derniers utilisent un algorithme de segmentation afin de découper la longueur des positions analysées en segments. Un segment est généralement constitué d'un gène complet et le calcul de ratio de profondeur/nombre d'UMI est effectué par segment. De ce fait, si nous insérons une amplification par exemple de 1000 pb dans un segment de longueur 25 000 pb, le segment ne sera pas considéré comme amplifié puisque la moyenne de profondeur/nombre d'UMI sera très proche de la profondeur moyenne du segment. C'est pour cela que nous avons augmenté la taille des CNV insérés afin qu'ils puissent significativement affecter le ratio du segment complet, et par conséquent, que ce dernier soit détecté comme amplifié/délété. L'amplification du chromosome 1 est ajouté dans le gène *TNFRSF14*, la délétion affectant le même chromosome touche le gène *ARID1A* et la délétion du chromosome 6 concerne le gène *PRDM1*. Nous avons utilisé l'outil mCNA [138] pour détecter les CNV insérés vu qu'il est le seul outil implémentant une analyse UMI dans son algorithme. Cet outil a été développé récemment au Centre Henri Becquerel et a été testé sur un grand nombre de données réelles [138]. Les résultats de l'analyse sont présentés dans la Figure 6.15. La figure représente le profil de l'échantillon simulé par UMI-Gen indiquant l'ensemble de

gènes détectés comme amplifié ou délété par mCNA. Les gènes *CD58* et *TNFRSF14* du chromosome 1 ainsi que le gène *EP300* du chromosome 22 sont détectés comme étant amplifiés alors que les délétions détectées concernent les gène *ARID1A* du chromosome 1, le gène *PRDM1* du chromosome 6 et le gène *NOTCH1* du chromosome 9. Ainsi, mCNA arrive à bien détecter les trois variants structuraux insérés par UMI-Gen mais détecte aussi trois faux positifs. La détection des faux positifs est probablement due au fait que mCNA ne se base pas que sur le nombre d'UMI mais aussi sur la profondeur brute des régions pour estimer la probabilité qu'une région soit réellement amplifiée ou délétée.

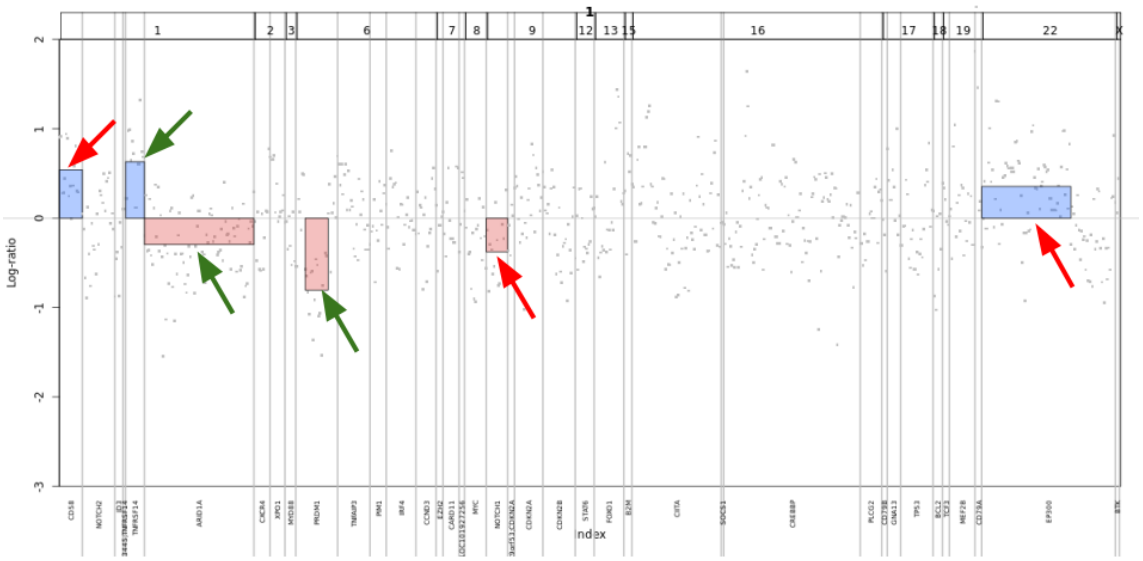


FIGURE 6.15 – Profil résultant de l'analyse CNV par mCNA d'un fichier simulé dans lequel trois variants structuraux ont été insérés par UMI-Gen. Les trois variants insérés sont indiqués par les flèches vertes alors que les flèches en rouge montrent les faux positifs. Les zones en rouge représentent des régions délétées alors que les régions amplifiées sont représentées en bleu.

6.3.8 Fichiers de sortie

UMI-Gen produit des *reads paired-end* et donc, deux fichiers FASTQ sont générés (R1 et R2) une fois tous les variants insérés. UMI-Gen appelle ensuite BWA pour faire l'alignement et produire le fichier d'alignement binaire BAM. SAMtools est enfin appelé pour créer le fichier d'index du BAM et convertir le BAM en SAM. Les cinq fichiers sont générés dans le répertoire de sortie souhaité. De plus, UMI-Gen génère un fichier binaire PILEUP qui correspond au *pileup* final des échantillons de contrôle et un fichier binaire MATRIX correspondant aux scores de qualité moyens sur les échantillons de contrôle. Ces deux fichiers peuvent être utilisés pour charger directement le *pileup* et la matrice des scores de qualité par position si l'analyse a déjà été effectuée sur les mêmes échantillons de contrôle.

6.3.9 Implémentation

L'exécution du *workflow* d'UMI-Gen est gérée par un script Python principal qui contrôle de nombreux modules Python3 et permet de lancer un des deux outils qui y sont inclus : le premier pour simuler des SNV et le second spécifique aux CNV. Afin d'obtenir

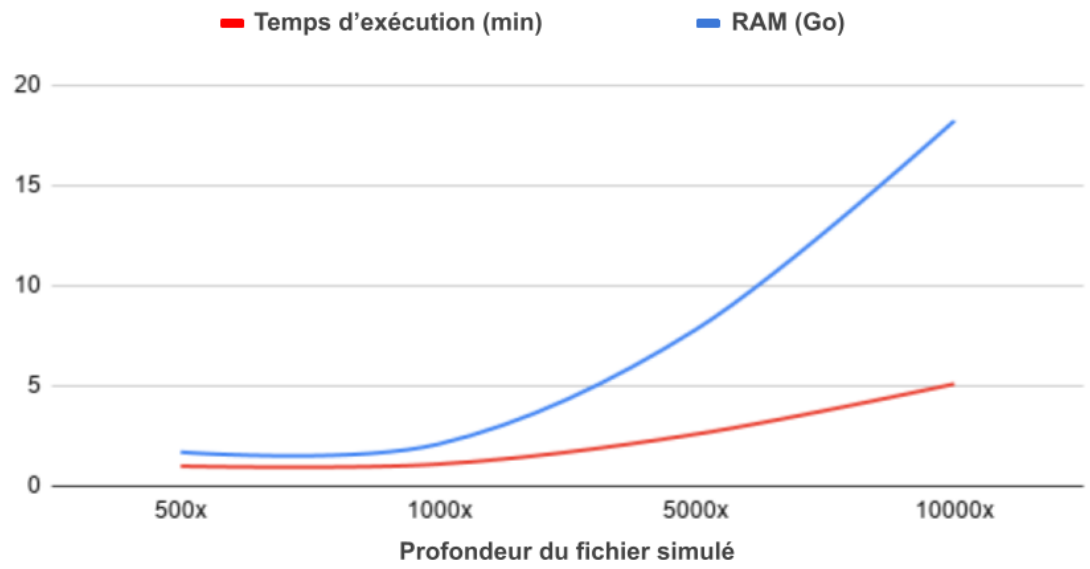


FIGURE 6.16 – Graphique montrant la variation de la performance d'UMI-Gen avec la profondeur du fichier produit en termes de temps d'exécution et de consommation en mémoire.

de meilleures performances globales, Cython a été utilisé pour compiler tous les modules Python. UMI-Gen nécessite l'installation des outils BWA et SAMtools sur l'ordinateur personnel/serveur : BWA est appelé pour l'étape d'alignement et SAMtools pour la conversion, le tri et l'indexation des fichiers BAM générés. Notre outil peut être exécuté via une interface en ligne de commande UNIX/Linux. Au total, UMI-Gen peut accepter 20 paramètres lors de l'exécution. La gestion de ces paramètres permet à l'utilisateur d'avoir un contrôle total sur ses données simulées.

6.3.10 Performance

Afin d'évaluer les performances d'UMI-Gen, nous avons simulé quatre échantillons avec des profondeurs croissantes : 500x, 1000x, 5000x et 10 000x. Pour chaque échantillon simulé, le temps d'exécution et la consommation mémoire ont été signalés. Les quatre échantillons ont été simulés en utilisant les mêmes six échantillons de contrôle. La première fois que nous exécutons UMI-Gen, l'étape de construction du *pileup* est obligatoire. Cette étape dépend uniquement des échantillons de contrôle et prend environ 1,5 min par échantillon. L'étape d'estimation de la qualité après la construction du *pileup* est également essentielle et prend en moyenne 0,5 min par échantillon. Cependant, ces deux étapes génèrent des fichiers qui peuvent être fournis directement au programme lors de l'exécution. Cela signifie que pour les autres fois, si l'utilisateur souhaite simuler des données en utilisant les mêmes échantillons de contrôle, le *pileup* et la matrice de qualité peuvent être utilisés directement, ce qui permet de gagner un temps considérable. La Table 6.6 détaille les temps d'exécution et la mémoire nécessaire pour générer chaque échantillon. La production des fichiers FASTQ ne prend que 1,57 min pour l'échantillon 500x et n'utilise que 1 Go de RAM. De l'autre côté, 16,58 min sont nécessaires pour un échantillon de 10 000x et la consommation de mémoire atteint 5,1 Go. Tous ces tests ont été réalisés sur un ordinateur fonctionnant sous Linux (Ubuntu 16.04) utilisant un seul cœur de processeur tournant à 2,20 GHz et équipé de 16 Go de RAM. Toutes les mesures ont été effectuées trois fois et la moyenne a été utilisée pour la comparaison. Après la

production du FASTQ, BWA et SAMtools sont appelés à partir de l'outil pour générer les fichiers BAM et SAM correspondants.

Échantillon	Simulation des données (min)	FASTQ en BAM (s)	Consommation mémoire (Go)
500x	1,57	8	1,0
1000x	1,87	14	1,1
5000x	6,97	52	2,6
10 000x	16,58	99	5,1

TABLE 6.6 – Analyse de performance d'UMI-Gen : la variation du temps d'exécution et de la consommation en mémoire en fonction de la profondeur des données simulées.

6.4 Synthèse

Dans ce chapitre, nous avons commencé par présenter les différents outils disponibles actuellement pour la simulation des *reads* courts. Vu l'absence d'outils permettant de simuler des *reads* avec UMI, nous avons développé UMI-Gen : un simulateur de *reads* basé sur les UMI et permettant d'insérer des CNV ou des SNV dans les fichiers simulés afin d'évaluer les algorithmes de détection de variants somatiques et structuraux. UMI-Gen se base sur un ensemble de fichiers de contrôle pour estimer le bruit de fond du séquenceur et reproduit ce dernier dans les fichiers produits. Ensuite, grâce à une liste des variants de type CNV ou SNV, il est capable d'insérer les variants fournis avant de produire les fichiers finaux. En ce qui concerne les SNV, il a été validé en simulant deux fichiers avec des profondeurs et des variants différents. Les fichiers produits ont été analysés par quatre *variant callers* différents afin de comparer leur performance et confirmer l'insertion des variants aux bonnes positions et aux bonnes fréquences alléliques. D'autre part, UMI-Gen a été utilisé pour simuler un fichier avec trois variants structuraux et un détecteur de CNV a été utilisé pour valider l'insertion des trois CNV ainsi que le bruit de fond. Actuellement et à notre connaissance, UMI-Gen est le seul outil permettant de simuler des *reads* avec des UMI et d'y insérer des CNV ou des SNV. Ainsi, UMI-Gen comble un manque essentiel et permettra de tester les nouveaux algorithmes pour la détection des SNV/CNV, surtout ceux basés sur les UMI.

Chapitre 7

Conclusion et perspectives

Ce chapitre présente la conclusion générale de cette thèse, ainsi que ses perspectives, proposées aussi bien à l'échelle des différents résultats décrits, que de manière plus générale.

7.1 Conclusion

Les objectifs de cette thèse s'inscrivent ainsi dans la large problématique de l'analyse des données de séquençage, et plus particulièrement des *reads* courts issus des technologies de séquençage de deuxième génération. Les aspects abordés dans cette problématique se sont principalement concentrés sur l'utilisation des UMI dans les deux domaines de la transcriptomique et de la génomique. En ce qui concerne la transcriptomique, les UMI ont été utilisés pour faciliter l'alignement des *reads* et pour supprimer les doublons de PCR permettant d'obtenir des résultats plus précis. Dans le cadre de la génomique, ces barcodes moléculaires ont servi à développer des nouvelles stratégies pour distinguer les vrais variants somatiques des artefacts de séquençage.

Comme présenté au sein du Chapitre 3, de nombreuses études ont déjà été réalisées implémentant efficacement une analyse des UMI, que ça soit dans des échantillons d'ARN ou d'ADN. Les travaux menés dans le cadre de cette thèse se sont donc décomposés en trois principaux objectifs. Tout d'abord, au vu des jeux de données importants produits par les technologies de NGS de deuxième génération, un premier outil a été conçu pour les analyser plus efficacement. Les données produites résultent d'une méthode innovante fusionnant une RT-MLPA classique à un séquenceur NGS. Cela a mené au développement d'un outil spécialement conçu pour analyser les données produites par cette méthode, ces dernières ne pouvant pas être traitées par des outils classiques. Le deuxième objectif s'inscrit dans le cadre de la génomique et permet de répondre à l'une des problématiques les plus importantes dans ce domaine : le *variant calling*. Ainsi, un *variant caller* a été développé pour une meilleure détection des variants somatiques de faible fréquence grâce à une implémentaion d'une analyse des UMI. Le développement de cet outil nous a permis d'identifier un manque dans le domaine du *variant calling* basé sur les UMI : l'absence d'un simulateur de *read* avec UMI. De ce fait, le troisième objectif a été rapidement établi et visait donc à développer un simulateur de *reads* avec UMI permettant de produire des données artificielles pour comparer les différentes stratégies de détection de variants somatiques.

7.1.1 RT-MLPA et NGS

Le premier objectif de cette thèse a été abordé dans le Chapitre 4 et consistait à développer un outil rapide et efficace pour analyser les données produites par une nouvelle méthode couplant une RT-MLPA classique à un séquenceur NGS. Deux applications principales ont été développées en se basant sur cette méthode : la classification

des lymphomes grâce à la mesure d'expression génique et la détection des transcrits de fusion dans les cellules tumorales. Ainsi, nous avons développé l'outil RT-MiS composé d'un logiciel permettant d'extraire les UMI, filtrer les *reads* d'un fichier FASTQ et effectuer des recherches rapides et efficaces dans chacun des *reads* pour l'attribuer au bon index et identifier la séquence du marqueur qu'il porte. RT-MiS intègre aussi une méthode de correction des UMI, indispensable dans la détection des transcrits de fusion, permettant de supprimer les doublons de PCR et retrouver le nombre de molécules uniques avec une grande efficacité. De plus, un modèle de classification de type *Random Forest* est appelé pour classer les échantillons analysés selon les niveaux d'expression de chaque gène. Cet outil a été implémenté dans une interface *web* permettant de gérer automatiquement la plupart des étapes de l'analyse, de la récupération du fichier FASTQ brut jusqu'à la production des résultats sous formes de graphiques interactifs, plus pratiques à utiliser et plus faciles à interpréter. L'outil RT-MiS est aujourd'hui l'un des principaux outils d'analyse au Centre Henri Becquerel.

7.1.2 UMI-VarCal

Le deuxième objectif de cette thèse a été présenté dans le Chapitre 5 et visait à améliorer la méthode classique du *variant calling* en utilisant les UMI. Pour cela, nous avons développé un *variant caller*, UMI-VarCal, implémentant à la fois un algorithme innovant pour un meilleur traitement des UMI dans les *reads* ainsi qu'une analyse UMI permettant de faire la distinction entre variant somatique et artefact de séquençage. L'analyse UMI devient particulièrement importante lorsque la fréquence des variants est très faible (comme dans le *cfDNA*). Nous avons testé UMI-VarCal contre trois *variant callers* différents : il a pu facilement détecter les variants trouvés par les autres outils et filtrer les faux positifs grâce à ses filtres de post-traitement en plusieurs étapes (filtre d'analyse UMI, filtre de biais de brin et filtre de région homopolymérique). De plus, il a été capable de détecter un nombre élevé de variants à des VAF $\leq 1\%$ et non appelés par les autres outils, dont 85% ont au moins un niveau de confiance élevé. Ainsi, l'analyse effectuée par UMI-VarCal génère des résultats plus précis que les autres outils tout en étant plus rapide en terme de temps d'exécution. Ces résultats démontrent comment l'approche basée sur les UMI permet de détecter avec plus de précision des variants à des VAF aussi basses que 0,1% sans sacrifier la spécificité ni la performance.

7.1.3 UMI-Gen

Le dernier objectif de cette thèse a été identifié suite au développement de l'outil UMI-VarCal, un *variant caller* basé sur les UMI. Afin de comparer objectivement sa performance avec d'autres outils, il a fallu simuler des *reads* courts dans lesquels les variants insérés sont connus. Vu l'absence d'outils permettant de simuler des *reads* avec UMI, nous avons répondu à ce manque en développant UMI-Gen : un simulateur de *reads* basé sur les UMI et permettant d'insérer des CNV ou des SNV dans les fichiers simulés afin d'évaluer les algorithmes de détection de variants somatiques et structuraux. Grâce à des fichiers de contrôle, UMI-Gen est capable d'estimer le bruit de fond moyen du séquenceur et de l'insérer dans les *reads* simulés. Ensuite, selon le mode d'utilisation souhaité, UMI-Gen permet l'insertion de variants somatiques (SNV) ou de variants structuraux (CNV) dans les *reads* simulés. Les fichiers générés contiendront alors des *reads* avec les UMI, des faux variants représentant le bruit de fond de séquençage et les vrais variants insérés à des fréquences connues et qui sont censés être détectés par les méthodes de *variant calling*. Actuellement et à notre connaissance, UMI-Gen est le seul logiciel capable de simuler des *reads* avec des UMI et d'y insérer des CNV ou des SNV. Ainsi, UMI-Gen

comble un manque essentiel dans le domaine de l'analyse des séquences nucléotidiques et permet d'évaluer les nouvelles stratégies pour la détection des SNV/CNV, particulièrement celles basées sur les UMI.

7.2 Perspectives

7.2.1 UMI-VarCal

7.2.1.1 Correction des UMI

Par définition, les UMI sont des séquences nucléotidiques dans lesquelles des erreurs de séquençage peuvent se produire aléatoirement. Bien que le taux d'erreur dans ces séquences reste faible mais augmente proportionnellement avec la taille de l'UMI, la présence de ces erreurs pourrait perturber l'analyse effectuée par UMI-VarCal pour distinguer entre vrai variant et artefact. Actuellement, UMI-VarCal effectue cette analyse sans tenir compte de ces erreurs. Pour améliorer la filtration des faux positifs, une correction des UMI se basant sur la méthode *directional* développée par UMI-tools peut être implémentée. Ceci permettra à UMI-VarCal de construire des graphes entre les UMI et de les résoudre en fusionnant ceux qui répondent à des critères bien déterminés. Par contre, cette étape est susceptible d'être plus coûteuse en termes de temps d'exécution et de consommation mémoire. Ainsi, il faudra l'implémenter de la manière la plus efficace dans UMI-VarCal et comparer les résultats sans et avec la correction pour déterminer si l'augmentation de précision des résultats obtenus justifie l'augmentation de temps et de consommation mémoire de l'outil.

7.2.1.2 Parallélisation et RAM

L'une des principales caractéristiques d'UMI-VarCal est sa haute performance qui dépasse facilement celle des autres *variant callers* basés sur les UMI mais aussi celle des outils classiques. Ceci est notamment dû à un algorithme de *pileup* très efficace conçu spécialement pour gérer les UMI dans les *reads*. De plus, le code source d'UMI-VarCal est parallélisé permettant ainsi de lancer l'analyse sur plusieurs cœurs en même temps et réduisant significativement le temps d'analyse, surtout pour les fichiers de grande taille. En effet, pour le faire, UMI-VarCal découpe le fichier initial en n sous-fichiers et lance des analyses séparées sur chaque sous-fichier. Une fois les n analyses terminées, les résultats sont fusionnés dans un résultat final. Bien que cela permette de réduire le temps d'exécution, la consommation mémoire augmente significativement et proportionnellement avec le nombre n de cœurs utilisés. De ce fait, il faudra essayer de trouver une meilleure gestion de la parallélisation qui idéalement permettra de réduire le temps d'exécution tout en achevant une gestion efficace de la consommation mémoire.

7.2.2 UMI-Gen

UMI-Gen représente aujourd'hui le premier et seul simulateur de données NGS permettant de produire des *reads* avec UMI tout en permettant une insertion de SNV ou de CNV à des fréquences et positions connues. Nous avons développé UMI-Gen pour évaluer UMI-VarCal et le comparer objectivement contre trois autres outils spécialement conçus pour la détection des variants à faible fréquence. De ce fait, UMI-Gen a été développé en se basant sur des données provenant d'un séquenceur de type Illumina MiSeq et donc produit des données compatibles avec les outils Illumina. Nous avons essayé d'intégrer une compatibilité avec les autres plateformes de séquençage telles que l'Ion

Torrent et Roche/454 (l'utilisateur peut choisir la plateforme souhaitée lors de l'exécution de l'outil) mais celle-ci n'a pas été testée rigoureusement. Ainsi, cette option reste « expérimentale » actuellement mais l'intégration d'une compatibilité avec les autres plateformes est sûrement envisageable tenant compte de son importance et du fait que pour l'instant, aucun autre outil n'est capable de simuler des *reads* avec UMI.

7.2.3 Perspectives générales

De manière plus générale, le travail réalisé dans le cadre de cette thèse s'est focalisé sur le traitement des UMI dans les données génomiques et transcriptomiques. Dans la génomique, ils ont été utilisés efficacement pour simuler des *reads* et mieux détecter les variants à faible fréquence rencontrés fréquemment dans les biopsies liquides (*cfDNA*). En ce qui concerne la transcriptomique, les UMI ont été utilisés dans des applications visant à classifier les échantillons de lymphomes en mesurant leur expression génique, ainsi que pour détecter des transcrits de fusion, marqueurs importants du cancer. Cependant, d'autres applications restent envisageables pour le futur vu l'importance des informations portées par les UMI.

Actuellement, suite à une expérience de séquençage NGS, des outils produisant des métriques de qualité telles que le taux de séquences surreprésentées, le taux des bases GC et la répartition des scores de qualité (par base et par *read*) sont indispensables pour s'assurer que l'expérience s'est bien passée et que les données produites sont exploitables. Cependant, aucun outil n'existe pour analyser ou calculer des métriques en relation avec les UMI. Par exemple, une analyse intéressante à effectuer serait le taux d'amplification de chaque UMI permettant de détecter une sous ou suramplification de certaines séquences. De plus, une autre analyse pouvant être utile serait la vérification que la complexité de la librairie d'UMI est assez élevée par rapport à la quantité initiale d'ADN utilisée. En effet, la complexité C de la librairie d'UMI est représentée par le nombre de combinaisons aléatoires possibles d'une séquence UMI. Cette valeur est calculée pour un UMI de taille n par la formule $C = 4^n$. La complexité doit être significativement supérieure au nombre de molécules initiales pour s'assurer que chaque fragment initial soit étiqueté par une séquence d'UMI différente. Ainsi, un outil permettant de détecter le taux d'UMI identiques attachés à des fragments d'ADN différents serait d'une grand utilité pour s'assurer que la taille de l'UMI est bien ajustée à la quantité d'ADN utilisée.

Finalement, en ce qui concerne la détection des variants, plusieurs *variant callers* permettent de classer les variants entre somatiques et constitutionnels. Cependant, une classification plus intéressante et plus utile serait de classer les faux variants entre erreur de PCR et erreur de séquençage. En effet, les méthodes de *variant calling* actuelles cherchent à identifier et filtrer le plus grand nombre de faux positifs dans les données analysées. Ces erreurs peuvent être des erreurs de PCR ou des erreurs de séquençage mais aucun outil ne s'intéresse à réaliser cette différence. Grâce à l'utilisation de l'information portée par les UMI, combinée aux scores de qualité des bases alternatives, cette distinction pourrait être réalisée permettant de mieux comprendre les causes probables amenant à l'apparition de chaque type d'erreur. La différence principale entre les deux types d'erreurs est le score de qualité : une erreur de séquençage est accompagnée d'un score de qualité faible alors qu'une erreur de PCR devrait avoir un score de qualité élevé. De plus, grâce aux UMI, des modèles statistiques pourraient certainement être développés afin d'estimer le numéro du cycle au bout duquel l'erreur de PCR est apparue. Un outil effectuant cette analyse pourrait éventuellement procéder à une correction de ces erreurs dans le fichier original. Il produira alors un fichier corrigé qui sera analysé par n'importe quel *variant caller*, rendant la tâche de ce dernier beaucoup plus facile et permettant de réduire considérablement son temps d'analyse.

Annexe A

Liste des publications et communications orales

Publications dans des revues internationales avec comité de lecture

[1] Vincent SATER, Pierre-Julien VIAILLY, Thierry LECROQ, Élise PRIEUR-GASTON, Élodie BOHERS, Mathieu VIENNOT, Philippe RUMINY, Hélène DAUCHEL, Pierre VERA et Fabrice JARDIN. UMI-VarCal : a new UMI-based variant caller that efficiently improves low-frequency variant detection in paired-end sequencing NGS libraries. *Bioinformatics*, 36.9 (2020), p. 2718-2724. DOI :10.1093/bioinformatics/btaa053

[2] Victor BOBÉE, Fanny DRIEUX, Vinciane MARCHAND, Vincent SATER, Liana VERESEZAN, Jean-Michel PICQUENOT, Pierre-Julien VIAILLY *et al.*. Combining gene expression profiling and machine learning to diagnose B-cell non-Hodgkin lymphoma. *Blood Cancer Journal*, 10.59 (2020). DOI : 10.1038/s41408-020-0322-5

[3] Vincent SATER, Pierre-Julien VIAILLY, Thierry LECROQ, Philippe RUMINY, Caroline BÉRARD, Élise PRIEUR-GASTON et Fabrice JARDIN. UMI-Gen : a UMI-based read simulator for variant calling evaluation in paired-end sequencing NGS libraries. *Computational and Structural Biotechnology Journal*, 18 (2020), p. 2270-2280. DOI : 10.1016/j.csbj.2020.08.011

[4] Pierre-Julien VIAILLY, Vincent SATER, Mathieu VIENNOT, Élodie BOHERS, Nicolas VERGNE, Caroline BÉRARD, Hélène DAUCHEL, Thierry LECROQ, Alison CELEBI, Philippe RUMINY et Fabrice JARDIN. Improving high-resolution copy number variation analysis from next generation sequencing using unique molecular identifiers. *BMC Bioinformatics*, 22.120 (2021). DOI : 10.1186/s12859-021-04060-4

Communications orales dans des conférences nationales

[5] Vincent SATER, Pierre-Julien VIAILLY, Thierry LECROQ, Élise PRIEUR-GASTON, Élodie BOHERS, Mathieu VIENNOT, Philippe RUMINY, Hélène DAUCHEL, Pierre VERA et Fabrice JARDIN. UMI-VarCal : a new UMI-based variant caller that efficiently improves low-frequency variant detection in paired-end sequencing NGS libraries. *Journées Ouvertes en Biologie, Informatique et Mathématiques (JOBIM)*, Montpellier, France (juillet 2020).

[6] Vincent SATER, Pierre-Julien VIAILLY, Thierry LECROQ, Philippe RUMINY, Caroline BÉRARD, Élise PRIEUR-GASTON et Fabrice JARDIN. UMI-Gen : a UMI-based read simulator for variant calling evaluation in paired-end sequencing NGS libraries. *Journées Ouvertes en Biologie, Informatique et Mathématiques (JOBIM)*, Paris, France (juillet 2021).

Communications orales

[7] Vincent SATER, Pierre-Julien VIAILLY, Thierry LECROQ, Élise PRIEUR-GASTON, Élodie BOHERS, Mathieu VIENNOT, Philippe RUMINY, Hélène DAUCHEL, Pierre VERA et Fabrice JARDIN. UMI-VarCal : a new UMI-based variant caller that efficiently improves low-frequency variant detection in paired-end sequencing NGS libraries. *SeqBIM*, Marne-la-Vallée, France (décembre 2019).

[8] Vincent SATER, Pierre-Julien VIAILLY, Thierry LECROQ, Philippe RUMINY, Caroline BÉRARD, Élise PRIEUR-GASTON et Fabrice JARDIN. UMI-Gen : a UMI-based read simulator for variant calling evaluation in paired-end sequencing NGS libraries. *SeqBIM*, Toulouse, France (novembre 2020).

Communications sous forme de poster

[9] Vincent SATER, Pierre-Julien VIAILLY, Thierry LECROQ, Élise PRIEUR-GASTON, Élodie BOHERS, Mathieu VIENNOT, Philippe RUMINY, Hélène DAUCHEL, Pierre VERA et Fabrice JARDIN. UMI-VarCal : a new UMI-based variant caller that efficiently improves low-frequency variant detection in paired-end sequencing NGS libraries. *Journées Ouvertes en Biologie, Informatique et Mathématiques (JOBIM)*, Nantes, France (juillet 2019).

[10] Vincent SATER, Pierre-Julien VIAILLY, Thierry LECROQ, Élise PRIEUR-GASTON, Élodie BOHERS, Mathieu VIENNOT, Philippe RUMINY, Hélène DAUCHEL, Pierre VERA et Fabrice JARDIN. UMI-VarCal : a new UMI-based variant caller that efficiently improves low-frequency variant detection in paired-end sequencing NGS libraries. *Journée Sciences du numérique*, Le Havre, France (octobre 2019).

Bibliographie

[1] J. D. WATSON et F. H. C. CRICK. Molecular structure of nucleic acids : a structure for deoxyribose nucleic acid. *Nature*, 171(4356) :737-738, avril 1953. ISSN : 1476-4687. DOI : 10.1038/171737a0.

[2] File :Difference DNA RNA-DE svg : Sponk / *translation : SPONK. Comparison of a single-stranded RNA and a double-stranded DNA with their corresponding nucleobases. 23 mars 2010. URL : https://commons.wikimedia.org/wiki/File:Difference_DNA_RNA-EN.svg (visité le 14/01/2021).

[3] Quo-Fata FERUNT. Français : sciences de la vie et de la terre - lycée - première s - code génétique (à 3 lettres), 29 novembre 2018. URL : https://commons.wikimedia.org/wiki/File:SVT_CodeGenetique.svg (visité le 14/01/2021).

[4] F. SANGER, S. NICKLEN et A. R. COULSON. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12) :5463-5467, 1er décembre 1977. ISSN : 0027-8424, 1091-6490. DOI : 10.1073/pnas.74.12.5463.

[5] A. M. MAXAM et W. GILBERT. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences*, 74(2) :560-564, 1er février 1977. ISSN : 0027-8424, 1091-6490. DOI : 10.1073/pnas.74.2.560.

[6] Sara GOODWIN, John D. MCPHERSON et W. Richard MCCOMBIE. Coming of age : ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6) :333-351, juin 2016. ISSN : 1471-0064. DOI : 10.1038/nrg.2016.49.

[7] Miodrag GUŽVIĆ. The history of DNA sequencing. *Journal of Medical Biochemistry*, 32(4) :301-312, 2013. ISSN : 1452-8258. DOI : 10.2478/jomb-2014-0004.

[8] Mehdi KCHOUK. Generations of sequencing technologies : from first to next generation. 9(3) :8, 2017.

[9] Jay SHENDURE, Shankar BALASUBRAMANIAN, George M. CHURCH, Walter GILBERT, Jane ROGERS, Jeffery A. SCHLOSS et Robert H. WATERSTON. DNA sequencing at 40 : past, present and future. *Nature*, 550(7676) :345-353, octobre 2017. ISSN : 1476-4687. DOI : 10.1038/nature24286.

[10] Jay SHENDURE et Hanlee JI. Next-generation DNA sequencing. *Nature Biotechnology*, 26(10) :1135-1145, octobre 2008. ISSN : 1546-1696. DOI : 10.1038/nbt1486.

[11] Christoph GOEMANS. Principle of DNA sequencing according to the dideoxy method. 20 septembre 2009. URL : https://commons.wikimedia.org/wiki/File:Didesoxy-Methode.svg (visité le 14/01/2021).

[12] Eric S. LANDER, Lauren M. LINTON, Bruce BIRREN, Chad NUSBAUM, Michael C. ZODY et BALDWIN. Initial sequencing and analysis of the human genome. *Nature*, 409(6822) :860-921, février 2001. ISSN : 1476-4687. DOI : 10.1038/35057062.

[13] w :User :Several TIMES. Diagram of an example of maxam–gilbert DNA sequencing and subsequent analysis by electrophoresis. 16 juillet 2013. URL : https://commons.wikimedia.org/wiki/File:Maxam-Gilbert_sequencing_en.svg (visité le 14/01/2021).

[14] Pierre MORISSE. *Correction de données de séquençage de troisième génération*. Theses, Normandie Université, septembre 2019. URL : https://tel.archives-ouvertes.fr/tel-02320413.

[15] Bianca K. STÖCKER, Johannes KÖSTER et Sven RAHMANN. SimLoRD : simulation of long read data. *Bioinformatics*, 32(17) :2704-2706, 1er septembre 2016. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btw286.

[16] Ze-Gang WEI et Shao-Wu ZHANG. NPBSS : a new PacBio sequencing simulator for generating the continuous long reads with an empirical model. *BMC Bioinformatics*, 19(1) :177, 22 mai 2018. ISSN : 1471-2105. DOI : 10.1186/s12859-018-2208-0.

[17] Yu LI, Renmin HAN, Chongwei BI, Mo LI, Sheng WANG et Xin GAO. DeepSimulator : a deep simulator for nanopore sequencing. *Bioinformatics*, 34(17) :2899-2908, 1er septembre 2018. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/bty223.

[18] Chen YANG, Justin CHU, René L. WARREN et Inanç BIROL. NanoSim : nanopore sequence read simulator based on statistical characterization. *GigaScience*, 6(4) :1-6, 1er avril 2017. ISSN : 2047-217X. DOI : 10.1093/gigascience/gix010.

[19] Weichun HUANG, Leping LI, Jason R. MYERS et Gabor T. MARTH. ART : a next-generation sequencing read simulator. *Bioinformatics*, 28(4) :593-594, 15 février 2012. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btr708.

[20] Ségolène CABOCHE, Christophe AUDEBERT, Yves LEMOINE et David HOT. Comparison of mapping algorithms used in high-throughput sequencing : application to ion torrent data. *BMC Genomics*, 15(1) :264, 5 avril 2014. ISSN : 1471-2164. DOI : 10.1186/1471-2164-15-264.

[21] Daniel C. RICHTER, Felix OTT, Alexander F. AUCH, Ramona SCHMID et Daniel H. HUSON. MetaSim—a sequencing simulator for genomics and metagenomics. *PLOS ONE*, 3(10) :e3373, 8 octobre 2008. ISSN : 1932-6203. DOI : 10.1371/journal.pone.0003373.

[22] Brent EWING et Phil GREEN. Base-calling of automated sequencer traces using phred. II. error probabilities. *Genome Research*, 8(3) :186-194, 3 janvier 1998. ISSN : 1088-9051, 1549-5469. DOI : 10.1101/gr.8.3.186.

[23] Brent EWING, LaDeana HILLIER, Michael C. WENDL et Phil GREEN. Base-calling of automated sequencer traces UsingPhred. i. accuracy assessment. *Genome Research*, 8(3) :175-185, 3 janvier 1998. ISSN : 1088-9051, 1549-5469. DOI : 10.1101/gr.8.3.175.

[24] Yun HEO, Xiao-Long WU, Deming CHEN, Jian MA et Wen-Mei HWU. BLESS : bloom filter-based error correction solution for high-throughput sequencing reads. *Bioinformatics*, 30(10) :1354-1362, 15 mai 2014. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btu030.

[25] David R. KELLEY, Michael C. SCHATZ et Steven L. SALZBERG. Quake : quality-aware detection and correction of sequencing errors. *Genome Biology*, 11(11) :R116, 29 novembre 2010. ISSN : 1474-760X. DOI : 10.1186/gb-2010-11-11-r116.

[26] Lucian ILIE, Farideh FAZAYELI et Silvana ILIE. HiTEC : accurate error correction in high-throughput sequencing data. *Bioinformatics*, 27(3) :295-302, 1er février 2011. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btq653.

[27] Leena SALMELA. Correction of sequencing errors in a mixed set of reads. *Bioinformatics*, 26(10) :1284-1290, 15 mai 2010. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btq151.

[28] Leena SALMELA et Jan SCHRÖDER. Correcting errors in short reads by multiple alignments. *Bioinformatics*, 27(11) :1455-1461, 1er juin 2011. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btr170.

[29] Wei-Chun KAO, Andrew H. CHAN et Yun S. SONG. ECHO : a reference-free short-read error correction algorithm. *Genome Research*, 21(7) :1181-1192, 7 janvier 2011. ISSN : 1088-9051, 1549-5469. DOI : 10.1101/gr.111351.110.

[30] X. YIN, Z. SONG, K. DORMAN et A. RAMAMOORTHY. PREMIER turbo : probabilistic error-correction using markov inference in errored reads using the turbo principle. In *2013 IEEE Global Conference on Signal and Information Processing*. 2013 IEEE Global Conference on Signal and Information Processing, pages 73-76, décembre 2013. DOI : 10.1109/GlobalSIP.2013.6736816.

[31] Ruifeng HU, Guibo SUN et Xiaobo SUN. LSCplus : a fast solution for improving long read accuracy by short read alignment. *BMC Bioinformatics*, 17(1) :451, 9 novembre 2016. ISSN : 1471-2105. DOI : 10.1186/s12859-016-1316-y.

[32] Olivia CHOUDHURY, Ankush CHAKRABARTY et Scott J. EMRICH. HECIL : a hybrid error correction algorithm for long reads with iterative learning. *Scientific Reports*, 8(1) :9936, 2 juillet 2018. ISSN : 2045-2322. DOI : 10.1038/s41598-018-28364-3.

[33] Chuan-Le XIAO, Ying CHEN, Shang-Qian XIE, Kai-Ning CHEN, Yan WANG, Yue HAN, Feng LUO et Zhi XIE. MECAT : fast mapping, error correction, and de novo assembly for single-molecule sequencing reads. *Nature Methods*, 14(11) :1072-1074, novembre 2017. ISSN : 1548-7105. DOI : 10.1038/nmeth.4432.

[34] German TISCHLER et Eugene W. MYERS. Non hybrid long read consensus using local de bruijn graph assembly. *bioRxiv* :106252, 6 février 2017. DOI : 10.1101/106252.

[35] Tom SMITH, Andreas HEGER et Ian SUDBERY. UMI-tools : modeling sequencing errors in unique molecular identifiers to improve quantification accuracy. *Genome Research*, 27(3) :491-499, mars 2017. ISSN : 1088-9051. DOI : 10.1101/gr.209601.116.

[36] T. F. SMITH et M. S. WATERMAN. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1) :195-197, 25 mars 1981. ISSN : 0022-2836. DOI : 10.1016/0022-2836(81)90087-5.

[37] Saul B. NEEDLEMAN et Christian D. WUNSCH. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3) :443-453, 28 mars 1970. ISSN : 0022-2836. DOI : 10.1016/0022-2836(70)90057-4.

[38] Stephen F. ALTSCHUL, Warren GISH, Webb MILLER, Eugene W. MYERS et David J. LIPMAN. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3) :403-410, 5 octobre 1990. ISSN : 0022-2836. DOI : 10.1016/S0022-2836(05)80360-2.

[39] Ben LANGMEAD, Cole TRAPNELL, Mihai POP et Steven L. SALZBERG. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3) :R25, 4 mars 2009. ISSN : 1474-760X. DOI : 10.1186/gb-2009-10-3-r25.

[40] Ben LANGMEAD et Steven L. SALZBERG. Fast gapped-read alignment with bowtie 2. *Nature Methods*, 9(4) :357-359, avril 2012. ISSN : 1548-7105. DOI : 10.1038/nmeth.1923.

[41] Heng LI et Richard DURBIN. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14) :1754-1760, 15 juillet 2009. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btp324.

[42] Heng LI et Richard DURBIN. Fast and accurate long-read alignment with burrows–wheeler transform. *Bioinformatics*, 26(5) :589-595, 1er mars 2010. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btp698.

[43] Heng LI. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv :1303.3997 [q-bio]*, 26 mai 2013. arXiv : 1303.3997.

[44] Mark J. CHAISSON et Glenn TESLER. Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR) : application and theory. *BMC Bioinformatics*, 13(1) :238, 19 septembre 2012. ISSN : 1471-2105. DOI : 10.1186/1471-2105-13-238.

[45] Gene MYERS. Efficient local alignment discovery amongst noisy long reads. In Dan BROWN et Burkhard MORGENSTERN, éditeurs, *Algorithms in Bioinformatics*, Lecture Notes in Computer Science, pages 52-67, Berlin, Heidelberg. Springer, 2014. ISBN : 978-3-662-44753-6. DOI : 10.1007/978-3-662-44753-6_5.

[46] Heng LI. Minimap and miniasm : fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, 32(14) :2103-2110, 15 juillet 2016. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btw152.

[47] Heng LI. Minimap2 : pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18) :3094-3100, 15 septembre 2018. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/bty191.

[48] Sequence alignment/map format and SAMtools | bioinformatics | oxford academic. URL : https://academic.oup.com/bioinformatics/article/25/16/2078/204688 (visité le 15/01/2021).

[49] Elaine R. MARDIS. Next-generation DNA sequencing methods. *Annual Review of Genomics and Human Genetics*, 9(1) :387-402, 1er septembre 2008. ISSN : 1527-8204. DOI : 10.1146/annurev.genom.9.081307.164359.

[50] Daniel C. KOBOLDT, Li DING, Elaine R. MARDIS et Richard K. WILSON. Challenges of sequencing human genomes. *Briefings in Bioinformatics*, 11(5) :484-498, 1er septembre 2010. ISSN : 1467-5463. DOI : 10.1093/bib/bbq016.

[51] Daniel FERNANDEZ-GARCIA, Allison HILLS, Karen PAGE, Robert K. HASTINGS, Bradley TOGHILL, Kate S. GODDARD, Charlotte ION, Olivia OGLE, Anna Rita BOYDELL, Kelly GLEASON, Mark RUTHERFORD, Adrian LIM, David S. GUTTERY, R. Charles COOMBES et Jacqueline A. SHAW. Plasma cell-free DNA (cfDNA) as a predictive and prognostic marker in patients with metastatic breast cancer. *Breast Cancer Research*, 21(1) :149, 19 décembre 2019. ISSN : 1465-542X. DOI : 10.1186/s13058-019-1235-8.

[52] Andreas WILM, Pauline Poh Kim AW, Denis BERTRAND, Grace Hui Ting YEO, Swee Hoe ONG, Chang Hua WONG, Chiea Chuen KHOR, Rosemary PETRIC, Martin Lloyd HIBBERD et Niranjan NAGARAJAN. LoFreq : a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets. *Nucleic Acids Research*, 40(22) :11189-11201, 1er décembre 2012. ISSN : 0305-1048. DOI : 10.1093/nar/gks918.

[53] Kristian CIBULSKIS, Michael S. LAWRENCE, Scott L. CARTER, Andrey SIVACHENKO, David JAFFE, Carrie SOUGNEZ, Stacey GABRIEL, Matthew MEYERSON, Eric S. LANDER et Gad GETZ. Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature Biotechnology*, 31(3) :213-219, mars 2013. ISSN : 1546-1696. DOI : 10.1038/nbt.2514.

[54] Jiarui DING, Ali BASHASHATI, Andrew ROTH, Arusha OLOUMI, Kane TSE, Thomas ZENG, Gholamreza HAFFARI, Martin HIRST, Marco A. MARRA, Anne CONDON, Samuel APARICIO et Sohrab P. SHAH. Feature-based classifiers for somatic mutation detection in tumour–normal paired sequencing data. *Bioinformatics*, 28(2) :167-175, 15 janvier 2012. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btr629.

[55] Etienne MULLER, Nicolas GOARDON, Baptiste BRAULT, Antoine ROUSSELIN, Germain PAIMPARAY, Angelina LEGROS, Robin FOUILLET, Olivia BRUET, Aurore TRANCHANT, Florian DOMIN, Chankannira SAN, Céline QUESNELLE, Thierry FREBOURG, Agathe RICOU, Sophie KRIEGER, Dominique VAUR et Laurent CASTERA. OutLyzer : software for extracting low-allele-frequency tumor mutations from sequencing background noise in clinical practice. *Oncotarget*, 7(48) :79485-79493, 4 novembre 2016. ISSN : 1949-2553. DOI : 10.18632/oncotarget.13103.

[56] Can KOCKAN, Faraz HACH, Iman SARRAFI, Robert H BELL, Brian MCCONEGHY, Kevin BEJA, Anne HAEGERT, Alexander W WYATT, Stanislav V VOLIK, Kim N CHI, Colin C COLLINS et S Cenk SAHINALP. SiNVICT : ultra-sensitive detection of single nucleotide variants and indels in circulating tumour DNA. *Bioinformatics*, 33(1) :26-34, 1er janvier 2017. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btw536.

[57] Mikhail SHUGAY, Andrew R. ZARETSKY, Dmitriy A. SHAGIN, Irina A. SHAGINA, Ivan A. VOLCHENKOV, Andrew A. SHELENKOV, Mikhail Y. LEBEDIN, Dmitriy V. BAGAEV, Sergey LUKYANOV et Dmitriy M. CHUDAKOV. MAGERI : computational pipeline for molecular-barcoded targeted resequencing. *PLOS Computational Biology*, 13(5) :e1005480, 5 mai 2017. ISSN : 1553-7358. DOI : 10.1371/journal.pcbi.1005480.

[58] T. Daniel ANDREWS, Yogesh JEELALL, Dipti TALAULIKAR, Christopher C. GOODNOW et Matthew A. FIELD. DeepSNVMiner : a sequence analysis tool to detect emergent, rare mutations in subsets of cell populations. *PeerJ*, 4 :e2074, 24 mai 2016. ISSN : 2167-8359. DOI : 10.7717/peerj.2074.

[59] Valentina BOEVA, Tatiana POPOVA, Maxime LIENARD, Sebastien TOFFOLI, Maud KAMAL, Christophe LE TOURNEAU, David GENTIEN, Nicolas SERVANT, Pierre GESTRAUD, Thomas RIO FRIO, Philippe HUPÉ, Emmanuel BARILLOT et Jean-François LAES. Multi-factor data normalization enables the detection of copy number aberrations in amplicon sequencing data. *Bioinformatics*, 30(24) :3443-3450, 15 décembre 2014. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btu436.

[60] Jason LI, Richard LUPAT, Kaushalya C. AMARASINGHE, Ella R. THOMPSON, Maria A. DOYLE, Georgina L. RYLAND, Richard W. TOTHILL, Saman K. HALGAMUGE, Ian G. CAMPBELL et Kylie L. GORRINGE. CONTRA : copy number analysis for targeted resequencing. *Bioinformatics*, 28(10) :1307-1313, 15 mai 2012. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/bts146.

[61] Daniel C. KOBOLDT, Qunyuan ZHANG, David E. LARSON, Dong SHEN, Michael D. MCLELLAN, Ling LIN, Christopher A. MILLER, Elaine R. MARDIS, Li DING et Richard K. WILSON. VarScan 2 : somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Research*, 22(3) :568-576, 3 janvier 2012. ISSN : 1088-9051, 1549-5469. DOI : 10.1101/gr.129684.111.

[62] Mehdi DERHOURHI. *Nouvelle technique de détection simultanée des variant ponctuels et des copy number variants dans l'obésité monogénique*. Theses, Université de Lille, mars 2019. URL : https://tel.archives-ouvertes.fr/tel-02077477.

[63] Jan O. KORBEL, Alexander Eckehart URBAN, Jason P. AFFOURTIT, Brian GODWIN, Fabian GRUBERT, Jan Fredrik SIMONS, Philip M. KIM, Dean PALEJEV, Nicholas J. CARRIERO, Lei DU, Bruce E. TAILLON, Zhoutao CHEN, Andrea TANZER, A. C. Eugenia SAUNDERS, Jianxiang CHI, Fengtang YANG, Nigel P. CARTER, Matthew E. HURLES, Sherman M. WEISSMAN, Timothy T. HARKINS, Mark B. GERSTEIN, Michael EGHOLM et Michael SNYDER. Paired-end mapping reveals extensive structural variation in the human genome. *Science*, 318(5849) :420-426, 19 octobre 2007. ISSN : 0036-8075, 1095-9203. DOI : 10.1126/science.1149504.

[64] Min ZHAO, Qingguo WANG, Quan WANG, Peilin JIA et Zhongming ZHAO. Computational tools for copy number variation (CNV) detection using next-generation sequencing data : features and perspectives. *BMC Bioinformatics*, 14(11) :S1, 13 septembre 2013. ISSN : 1471-2105. DOI : 10.1186/1471-2105-14-S11-S1.

[65] Teemu KIVIOJA, Anna VÄHÄRAUTIO, Kasper KARLSSON, Martin BONKE, Martin ENGE, Sten LINNARSSON et Jussi TAIPALE. Counting absolute numbers of molecules using unique molecular identifiers. *Nature Methods*, 9(1) :72-74, janvier 2012. ISSN : 1548-7105. DOI : 10.1038/nmeth.1778.

[66] Yoji KUKITA, Ryo MATOBA, Junji UCHIDA, Takuya HAMAKAWA, Yuichiro DOKI, Fumio IMAMURA et Kikuya KATO. High-fidelity target sequencing of individual molecules identified using barcode sequences : de novo detection and absolute quantitation of mutations in plasma cell-free DNA from cancer patients. *DNA Research : An International Journal for Rapid Publication of Reports on Genes and Genomes*, 22(4) :269-277, août 2015. ISSN : 1340-2838. DOI : 10.1093/dnares/dsv010.

[67] Sébastien COURAUD, Felipe VACA-PANIAGUA, Stéphanie VILLAR, Javier OLIVER, Tibor SCHUSTER, Hélène BLANCHÉ, Nicolas GIRARD, Jean TRÉDANIEL, Laurent GUILLEMINAULT, Radj GERVAIS, Nathalie PRIM, Michel VINCENT, Jacques MARGERY, Sébastien LARIVÉ, Pascal FOUCHER, Bernard DUVERT, Maxime VALLEE, Florence Le CALVEZ-KELM, James MCKAY, Pascale MISSY, Franck MORIN, Gérard ZALCMAN, Magali OLIVIER et Pierre-Jean SOUQUET. Noninvasive diagnosis of actionable mutations by deep sequencing of circulating free DNA in lung cancer from never-smokers : a proof-of-concept study from BioCAST/IFCT-1002. *Clinical Cancer Research*, 20(17) :4613-4624, 1er septembre 2014. ISSN : 1078-0432, 1557-3265. DOI : 10.1158/1078-0432.CCR-13-3063.

[68] Jongsuk CHUNG, Ki-Wook LEE, Chung LEE, Seung-Ho SHIN, Sungkyu KYUNG, Hyo-Jeong JEON, Sook-Young KIM, Eunjung CHO, Chang Eun YOO, Dae-Soon SON, Woong-Yang PARK et Donghyun PARK. Performance evaluation of commercial library construction kits for PCR-based targeted sequencing using a unique molecular identifier. *BMC Genomics*, 20, 14 mars 2019. ISSN : 1471-2164. DOI : 10.1186/s12864-019-5583-7.

[69] Johnny A. SENA, Giulia GALOTTO, Nico P. DEVITT, Melanie C. CONNICK, Jennifer L. JACOBI, Pooja E. UMALE, Luis VIDALI et Callum J. BELL. Unique molecular identifiers reveal a novel sequencing artefact with implications for RNA-seq based gene expression analysis. *Scientific Reports*, 8, 3 septembre 2018. ISSN : 2045-2322. DOI : 10.1038/s41598-018-31064-7.

[70] Alexander DOBIN, Carrie A. DAVIS, Felix SCHLESINGER, Jorg DRENKOW, Chris ZALESKI, Sonali JHA, Philippe BATUT, Mark CHAISSON et Thomas R. GINGERAS. STAR : ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1) :15-21, 1er janvier 2013. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/bts635.

[71] Thomas D. WU et Serban NACU. Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics*, 26(7) :873-881, 1er avril 2010. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btq057.

[72] Daehwan KIM, Ben LANGMEAD et Steven L. SALZBERG. HISAT : a fast spliced aligner with low memory requirements. *Nature Methods*, 12(4) :357-360, avril 2015. ISSN : 1548-7105. DOI : 10.1038/nmeth.3317.

[73] Yu FU, Pei-Hsuan WU, Timothy BEANE, Phillip D. ZAMORE et Zhiping WENG. Elimination of PCR duplicates in RNA-seq and small RNA-seq using unique molecular identifiers. *BMC Genomics*, 19(1) :531, 13 juillet 2018. ISSN : 1471-2164. DOI : 10.1186/s12864-018-4933-1.

[74] Evan Z. MACOSKO, Anindita BASU, Rahul SATIJA, James NEMESH, Karthik SHEKHAR, Melissa GOLDMAN, Itay TIROSH, Allison R. BIALAS, Nolan KAMITAKI, Emily M. MARTERSTECK, John J. TROMBETTA, David A. WEITZ, Joshua R. SANES, Alex K. SHALEK, Aviv REGEV et Steven A. MCCARROLL. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5) :1202-1214, 21 mai 2015. ISSN : 0092-8674. DOI : 10.1016/j.cell.2015.05.002.

[75] Saiful ISLAM, Amit ZEISEL, Simon JOOST, Gioele LA MANNO, Pawel ZAJAC, Maria KASPER, Peter LÖNNERBERG et Sten LINNARSSON. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature Methods*, 11(2) :163-166, février 2014. ISSN : 1548-7105. DOI : 10.1038/nmeth.2772.

[76] Sayantan BOSE, Zhenmao WAN, Ambrose CARR, Abbas H. RIZVI, Gregory VIEIRA, Dana PE'ER et Peter A. SIMS. Scalable microfluidics for single-cell RNA printing and sequencing. *Genome Biology*, 16(1), 2015. ISSN : 1465-6906. DOI : 10.1186/s13059-015-0684-3.

[77] BROAD INSTITUTE. Picard tools. http://broadinstitute.github.io/picard/, (Accessed : 2018/02/21 ; version 2.17.8).

[78] Aaron MCKENNA, Matthew HANNA, Eric BANKS, Andrey SIVACHENKO, Kristian CIBULSKIS, Andrew KERNYTSKY, Kiran GARIMELLA, David ALTSHULER, Stacey GABRIEL, Mark DALY et Mark A. DEPRISTO. The genome analysis toolkit : a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, 20(9) :1297-1303, septembre 2010. ISSN : 1088-9051. DOI : 10.1101/gr.107524.110.

[79] Shifu CHEN, Yanqing ZHOU, Yaru CHEN, Tanxiao HUANG, Wenting LIAO, Yun XU, Zhicheng LI et Jia GU. Gencore : an efficient tool to generate consensus reads for error suppressing and duplicate removing of NGS data. *BMC Bioinformatics*, 20, Suppl 23, 27 décembre 2019. ISSN : 1471-2105. DOI : 10.1186/s12859-019-3280-9.

[80] Chang XU, Xiujing GU, Raghavendra PADMANABHAN, Zhong WU, Quan PENG, John DICARLO et Yexun WANG. smCounter2 : an accurate low-frequency variant caller for targeted sequencing data with unique molecular identifiers. *Bioinformatics*, 35(8) :1299-1309, 15 avril 2019. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/bty790.

[81] Helga THORVALDSDÓTTIR, James T. ROBINSON et Jill P. MESIROV. Integrative genomics viewer (IGV) : high-performance genomics data visualization and exploration. *Briefings in Bioinformatics*, 14(2) :178-192, mars 2013. ISSN : 1467-5463. DOI : 10.1093/bib/bbs017.

[82] Erik GARRISON et Gabor MARTH. Haplotype-based variant detection from short-read sequencing. *arXiv e-prints*, 1207 :arXiv :1207.3907, 1er juillet 2012.

[83] Heng LI. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, 27(21) :2987-2993, 1er novembre 2011. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btr509.

[84] Zhongwu LAI, Aleksandra MARKOVETS, Miika AHDESMAKI, Brad CHAPMAN, Oliver HOFMANN, Robert MCEWEN, Justin JOHNSON, Brian DOUGHERTY, J. Carl BARRETT et Jonathan R. DRY. VarDict : a novel and versatile variant caller for next-generation sequencing in cancer research. *Nucleic Acids Research*, 44(11) :e108, 20 juin 2016. ISSN : 0305-1048. DOI : 10.1093/nar/gkw227.

[85] Sydney DUBOIS, Pierre-Julien VIAILLY, Elodie BOHERS, Philippe BERTRAND, Philippe RUMINY, Vinciane MARCHAND, Catherine MAINGONNAT, Sylvain MARESCHAL, Jean-Michel PICQUENOT, Dominique PENTHER, Jean-Philippe JAIS, Bruno TESSON, Pauline PEYROUZE, Martin FIGEAC, Fabienne DESMOTS, Thierry FEST, Corinne HAIOUN, Thierry LAMY, Christiane COPIE-BERGMAN, Bettina FABIANI, Richard DELARUE, Frédéric PEYRADE, Marc ANDRÉ, Nicolas KETTERER, Karen LEROY, Gilles SALLES, Thierry J. MOLINA, Hervé TILLY et Fabrice JARDIN. Biological and Clinical Relevance of Associated Genomic Alterations in MYD88 L265p and non-L265p-Mutated Diffuse Large B-Cell Lymphoma : Analysis of 361 Cases. eng. *Clin. Cancer Res.*, 23(9) :2232-2244, mai 2017. ISSN : 1078-0432. DOI : 10.1158/1078-0432.CCR-16-1922.

[86] InfoCancer - ARCAGY - GINECO - les localisations - hémopathies malignes (cancers du sang) - lymphomes non hodgkiniens (LNH) - FORMES DE LA MALADIE- le lymphome folliculaire. URL : http://www.arcagy.org/infocancer/localisations/hemopathies-malignes-cancers-du-sang/lymphomes-non-hodgkiniens/formes-de-la-maladie/le-lymphome-folliculaire.html/ (visité le 02/03/2021).

[87] InfoCancer - ARCAGY - GINECO - localisations - cancers du sang - hémopathies - leucémie lymphoïde chronique (LLC) - formes de la maladie - plus rarement. URL : http://www.arcagy.org/infocancer/localisations/hemopathies-malignes-cancers-du-sang/leucemie-lymphoide-chronique/formes-de-la-maladie/plus-rarement.html/ (visité le 02/03/2021).

[88] Les localisations - hémopathies malignes (cancers du sang) - lymphomes non hodgkiniens (LNH) - FORMES DE LA MALADIE - le lymphome à cellules du manteau. URL : http://www.arcagy.org/infocancer/localisations/hemopathies-malignes-cancers-du-sang/lymphomes-non-hodgkiniens/formes-de-la-maladie/le-lymphome-du-manteau.html/ (visité le 02/03/2021).

[89] InfOcancer - arcagy-GINECO - les localisations hémopathies malignes (cancers du sang) lymphomes non hodgkiniens (LNH) - FORMES DE LA MALADIE - les lymphomes de la zone marginale. URL : http://www.arcagy.org/infocancer/localisations/hemopathies-malignes-cancers-du-sang/lymphomes-non-hodgkiniens/formes-de-la-maladie/les-lymphomes-de-la-zone-marginale.html/ (visité le 02/03/2021).

[90] Camille LAURENT, Marine BARON, Nadia AMARA, Corinne HAIOUN, Mylène DANDOIT, Marc MAYNADIÉ, Marie PARRENS, Beatrice VERGIER, Christiane COPIE-BERGMAN, Bettina FABIANI et OTHERS. Impact of Expert Pathologic Review of Lymphoma Diagnosis : Study of Patients From the French Lymphopath Network. *Journal of Clinical Oncology*, 35(18) :2008-2017, 2017.

[91] Steven H. SWERDLOW, Elias CAMPO, Stefano A. PILERI, Nancy Lee HARRIS, Harald STEIN, Reiner SIEBERT, Ranjana ADVANI, Michele GHIELMINI, Gilles A. SALLES, Andrew D. ZELENETZ et Elaine S. JAFFE. The 2016 revision of the World Health Organization classification of lymphoid neoplasms. eng. *Blood*, 127(20) :2375-2390, mai 2016. ISSN : 1528-0020. DOI : 10.1182/blood-2016-01-643569.

[92] Monica BELLEI, Elena SABATTINI, Emanuela Anna PESCE, Young-Hyeh KO, Won Seog KIM, Maria Elena CABRERA, Virginia MARTINEZ, Ivan DLOUHY, Roberto Pinto PAES, Tomas BARRESE, Josè VASSALLO, Vittoria TARANTINO, Julie VOSE, Dennis WEISENBURGER, Thomas RÜDIGER, Massimo FEDERICO et Stefano PILERI. Pitfalls and major issues in the histologic diagnosis of peripheral T-cell lymphomas : results of the central review of 573 cases from the T-Cell Project, an international, cooperative study : Pitfalls and issues in the histologic diagnosis of PTCLs. en. *Hematological Oncology*, 2016. ISSN : 02780232. DOI : 10.1002/hon.2316.

[93] Lucile COURONNÉ, Christian BASTARD et Olivier A. BERNARD. TET2 and DNMT3a mutations in human T-cell lymphoma. eng. *N. Engl. J. Med.*, 366(1) :95-96, janvier 2012. ISSN : 1533-4406. DOI : 10.1056/NEJMc1111708.

[94] François LEMONNIER, Lucile COURONNÉ, Marie PARRENS, Jean-Philippe JAÏS, Marion TRAVERT, Laurence LAMANT, Olivier TOURNILLAC, Therese ROUSSET, Bettina FABIANI, Rob A. CAIRNS, Tak MAK, Christian BASTARD, Olivier A. BERNARD, Laurence de LEVAL et Philippe GAULARD. Recurrent TET2 mutations in peripheral T-cell lymphomas correlate with TFH-like features and adverse clinical parameters. eng. *Blood*, 120(7) :1466-1469, août 2012. ISSN : 1528-0020. DOI : 10.1182/blood-2012-02-408542.

[95] C. WANG, T. W. MCKEITHAN, Q. GONG, W. ZHANG, A. BOUSKA, A. ROSENWALD, R. D. GASCOYNE, X. WU, J. WANG, Z. MUHAMMAD, B. JIANG, J. ROHR, A. CANNON, C. STEIDL, K. FU, Y. LI, S. HUNG, D. D. WEISENBURGER, T. C. GREINER, L. SMITH, G. OTT, E. G. ROGAN, L. M. STAUDT, J. VOSE, J. IQBAL et W. C. CHAN. IDH2r172 mutations define a unique subgroup of patients with angioimmunoblastic T-cell lymphoma. en. *Blood*, 126(15) :1741-1752, octobre 2015. ISSN : 0006-4971, 1528-0020. DOI : 10.1182/blood-2015-05-644591.

[96] Teresa PALOMERO, Lucile COURONNÉ, Hossein KHIABANIAN, Mi-Yeon KIM, Alberto AMBESI-IMPIOMBATO, Arianne PEREZ-GARCIA, Zachary CARPENTER, Francesco ABATE, Maddalena ALLEGRETTA, J Erika HAYDU, Xiaoyu JIANG, Izidore S LOSSOS, Concha NICOLAS, Milagros BALBIN, Christian BASTARD, Govind BHAGAT, Miguel A PIRIS, Elias CAMPO, Olivier A BERNARD, Raul RABADAN et Adolfo A FERRANDO. Recurrent mutations in epigenetic regulators, RHOA and FYN kinase in peripheral T cell lymphomas. *Nature Genetics*, 46(2) :166-170, janvier 2014. ISSN : 1061-4036, 1546-1718. DOI : 10.1038/ng.2873.

[97] Hae Yong YOO, Min Kyung SUNG, Seung Ho LEE, Sangok KIM, Haeseung LEE, Seongjin PARK, Sang Cheol KIM, Byungwook LEE, Kyoohyoung RHO, Jong-Eun LEE, Kwang-Hwi CHO, Wankyu KIM, Hyunjung JU, Jaesang KIM, Seok Jin KIM, Won Seog KIM, Sanghyuk LEE et Young Hyeh KO. A recurrent inactivating mutation in RHOA GTPase in angioimmunoblastic T cell lymphoma. *Nature Genetics*, 46(4) :371-375, mars 2014. ISSN : 1061-4036, 1546-1718. DOI : 10.1038/ng.2916.

[98] Lucile COURONNÉ, Christian BASTARD, Philippe GAULARD, Olivier HERMINE et Olivier BERNARD. [Molecular pathogenesis of peripheral T-cell lymphoma (1) : angioimmunoblastic T-cell lymphoma, peripheral T-cell lymphoma, not otherwise specified and anaplastic large cell lymphoma]. fre. *Med Sci (Paris)*, 31(10) :841-852, octobre 2015. ISSN : 1958-5381. DOI : 10.1051/medsci/20153110010.

[99] Edgardo R. PARRILLA CASTELLAR, Elaine S. JAFFE, Jonathan W. SAID, Steven H. SWERDLOW, Rhett P. KETTERLING, Ryan A. KNUDSON, Jagmohan S. SIDHU, Eric D. HSI, Shridevi KARIKEHALLI, Liuyan JIANG, George VASMATZIS, Sarah E. GIBSON, Sarah ONDREJKA, Alina NICOLAE, Karen L. GROGG, Cristine ALLMER, Kay M. RISTOW, Wyndham H. WILSON, William R. MACON, Mark E. LAW, James R. CERHAN, Thomas M. HABERMANN, Stephen M. ANSELL, Ahmet DOGAN, Matthew J. MAURER et Andrew L. FELDMAN. ALK-negative anaplastic large cell lymphoma is a genetically heterogeneous disease with widely disparate clinical outcomes. eng. *Blood*, 124(9) :1473-1480, août 2014. ISSN : 1528-0020. DOI : 10.1182/blood-2014-04-571091.

[100] K. SUGATA, J.-i. YASUNAGA, H. KINOSADA, Y. MITOBE, R. FURUTA, M. MAHGOUB, C. ONISHI, K. NAKASHIMA, K. OHSHIMA et M. MATSUOKA. HTLV-1 Viral Factor HBZ Induces CCR4 to Promote T-cell Migration and Proliferation. en. *Cancer Research*, 76(17) :5068-5079, septembre 2016. ISSN : 0008-5472, 1538-7445. DOI : 10.1158/0008-5472.CAN-16-0361.

[101] Lucile COURONNÉ, Christian BASTARD, Philippe GAULARD, Olivier HERMINE et Olivier BERNARD. [Molecular pathogenesis of peripheral T cell lymphoma (2) : extranodal NK/T cell lymphoma, nasal type, adult T cell leukemia/lymphoma and enteropathy associated T cell lymphoma]. fre. *Med Sci (Paris)*, 31(11) :1023-1033, novembre 2015. ISSN : 1958-5381. DOI : 10.1051/medsci/20153111017.

[102] Javeed IQBAL, Dennis D. WEISENBURGER, Timothy C. GREINER, Julie M. VOSE, Timothy MCKEITHAN, Can KUCUK, Huimin GENG, Karen DEFFENBACHER, Lynette SMITH, Karen DYBKAER et OTHERS. Molecular signatures to improve diagnosis in peripheral T-cell lymphoma and prognostication in angioimmunoblastic T-cell lymphoma. *Blood*, 115(5) :1026-1036, 2010.

[103] Javeed IQBAL, George WRIGHT, Chao WANG, Andreas ROSENWALD, Randy D. GASCOYNE, Dennis D. WEISENBURGER, Timothy C. GREINER, Lynette SMITH, Shuangping GUO, Ryan A. WILCOX, Bin Tean TEH, Soon Thye LIM, Soon Yong TAN, Lisa M. RIMSZA, Elaine S. JAFFE, Elias CAMPO, Antonio MARTINEZ, Jan DELABIE, Rita M. BRAZIEL, James R. COOK, Raymond R. TUBBS, German OTT, Eva GEISSINGER, Philippe GAULARD, Pier Paolo PICCALUGA, Stefano A. PILERI, Wing Y. AU, Shigeo NAKAMURA, Masao SETO, Francoise BERGER, Laurence de LEVAL, Joseph M. CONNORS, James ARMITAGE, Julie VOSE, Wing C. CHAN, Louis M. STAUDT et LYMPHOMA LEUKEMIA MOLECULAR PROFILING PROJECT AND THE INTERNATIONAL PERIPHERAL T-CELL LYMPHOMA PROJECT. Gene expression signatures delineate biological and prognostic subgroups in peripheral T-cell lymphoma. eng. *Blood*, 123(19) :2915-2923, mai 2014. ISSN : 1528-0020. DOI : 10.1182/blood-2013-11-536359.

[104] Tianjiao WANG, Andrew L. FELDMAN, David A. WADA, Ye LU, Avery POLK, Robert BRISKI, Kay RISTOW, Thomas M. HABERMANN, Dafydd THOMAS, Steven C. ZIESMER et OTHERS. GATA-3 expression identifies a high-risk subset of PTCL, NOS with distinct molecular and clinical features. *Blood*, 123(19) :3007-3015, 2014.

[105] Eric ELDERING, C. Arnold SPEK, Hella L. ABERSON, Annette GRUMMELS, Ingrid A. DERKS, Alex F. de VOS, Cathal J. MCELGUNN et Jan P. SCHOUTEN. Expression profiling via novel multiplex assay allows rapid assessment of gene regulation in defined signalling pathways. *Nucleic Acids Research*, 31(23) :e153-e153, 1er décembre 2003. ISSN : 0305-1048. DOI : 10.1093/nar/gng153.

[106] Victor BOBÉE, Fanny DRIEUX, Vinciane MARCHAND, Vincent SATER, Liana VERESEZAN, Jean-Michel PICQUENOT, Pierre-Julien VIAILLY, Marie-Delphine LANIC, Mathieu VIENNOT, Elodie BOHERS, Lucie OBERIC, Christiane COPIE-BERGMAN, Thierry Jo MOLINA, Philippe GAULARD, Corinne HAIOUN, Gilles SALLES, Hervé TILLY, Fabrice JARDIN et Philippe RUMINY. Combining gene expression profiling and machine learning to diagnose b-cell non-hodgkin lymphoma. *Blood Cancer Journal*, 10(5) :1-13, 22 mai 2020. ISSN : 2044-5385. DOI : 10.1038/s41408-020-0322-5.

[107] Célia DUPAIN, Anne C. HARTTRAMPF, Yannick BOURSIN, Manuel LEBEURRIER, Windy RONDOF, Guillaume ROBERT-SIEGWALD, Pierre KHOUEIRY, Birgit GEOERGER et Liliane MASSAAD-MASSADE. Discovery of new fusion transcripts in a cohort of pediatric solid cancers at relapse and relevance for personalized medicine. *Molecular Therapy*, 27(1) :200-218, 2 janvier 2019. ISSN : 1525-0016, 1525-0024. DOI : 10.1016/j.ymthe.2018.10.022.

[108] Matthew J. ELLIS, Li DING, Dong SHEN, Jingqin LUO, Vera J. SUMAN, John W. WALLIS, Brian A. VAN TINE, Jeremy HOOG, Reece J. GOIFFON, Theodore C. GOLDSTEIN, Sam NG, Li LIN, Robert CROWDER, Jacqueline SNIDER, Karla BALLMAN, Jason WEBER, Ken CHEN, Daniel C. KOBOLDT, Cyriac KANDOTH, William S. SCHIERDING, Joshua F. MCMICHAEL, Christopher A. MILLER, Charles LU, Christopher C. HARRIS, Michael D. MCLELLAN, Michael C. WENDL, Katherine DESCHRYVER, D. Craig ALLRED, Laura ESSERMAN, Gary UNZEITIG, Julie MARGENTHALER, G.V. BABIERA, P. Kelly MARCOM, J.M. GUENTHER, Marilyn LEITCH, Kelly HUNT, John OLSON, Yu TAO, Christopher A. MAHER, Lucinda L. FULTON, Robert S. FULTON, Michelle HARRISON, Ben OBERKFELL, Feiyu DU, Ryan DEMETER, Tammi L. VICKERY, Adnan ELHAMMALI, Helen PIWNICA-WORMS, Sandra MCDONALD, Mark WATSON, David J. DOOLING, David OTA, Li-Wei CHANG, Ron BOSE, Timothy J. LEY, David PIWNICA-WORMS, Joshua M. STUART, Richard K. WILSON et Elaine R. MARDIS. Whole genome analysis informs breast cancer response to aromatase inhibition. *Nature*, 486(7403) :353-360, 10 juin 2012. ISSN : 0028-0836. DOI : 10.1038/nature11143.

[109] John S. WELCH, Timothy J. LEY, Daniel C. LINK, Christopher A. MILLER, David E. LARSON, Daniel C. KOBOLDT, Lukas D. WARTMAN, Tamara L. LAMPRECHT, Fulu LIU, Jun XIA, Cyriac KANDOTH, Robert S. FULTON, Michael D. MCLELLAN, David J. DOOLING, John W. WALLIS, Ken CHEN, Christopher C. HARRIS, Heather K. SCHMIDT, Joelle M. KALICKI-VEIZER, Charles LU, Qunyuan ZHANG, Ling LIN, Michelle D. O'LAUGHLIN, Joshua F. MCMICHAEL, Kim D. DELEHAUNTY, Lucinda A. FULTON, Vincent J. MAGRINI, Sean D. MCGRATH, Ryan T. DEMETER, Tammi L. VICKERY, Jasreet HUNDAL, Lisa L. COOK, Gary W. SWIFT, Jerry P. REED, Patricia A. ALLDREDGE, Todd N. WYLIE, Jason R. WALKER, Mark A. WATSON, Sharon E. HEATH, William D. SHANNON, Nobish VARGHESE, Rakesh NAGARAJAN, Jacqueline E. PAYTON, Jack D. BATY, Shashikant KULKARNI, Jeffery M. KLCO, Michael H. TOMASSON, Peter WESTERVELT, Matthew J. WALTER, Timothy A. GRAUBERT, John F. DIPERSIO, Li DING, Elaine R. MARDIS et Richard K. WILSON. The origin and evolution of mutations in acute myeloid leukemia. *Cell*, 150(2) :264-278, 20 juillet 2012. ISSN : 0092-8674. DOI : 10.1016/j.cell.2012.06.023.

[110] Shinichi YACHIDA, Siân JONES, Ivana BOZIC, Tibor ANTAL, Rebecca LEARY, Baojin FU, Mihoko KAMIYAMA, Ralph H. HRUBAN, James R. ESHLEMAN, Martin A. NOWAK, Victor E. VELCULESCU, Kenneth W. KINZLER, Bert VOGELSTEIN et Christine A. IACOBUZIO-DONAHUE. Distant metastasis occurs late during the genetic evolution of pancreatic cancer. *Nature*, 467(7319) :1114-1117, 28 octobre 2010. ISSN : 0028-0836. DOI : 10.1038/nature09515.

[111] Serena NIK-ZAINAL, Ludmil B. ALEXANDROV, David C. WEDGE, Peter VAN LOO, Christopher D. GREENMAN, Keiran RAINE, David JONES, Jonathan HINTON, John MARSHALL, Lucy A. STEBBINGS, Andrew MENZIES, Sancha MARTIN, Kenric LEUNG, Lina CHEN, Catherine LEROY, Manasa RAMAKRISHNA, Richard RANCE, King Wai LAU, Laura J. MUDIE, Ignacio VARELA, David J. MCBRIDE, Graham R. BIGNELL, Susanna L. COOKE, Adam SHLIEN, John GAMBLE, Ian WHITMORE, Mark MADDISON, Patrick S. TARPEY, Helen R. DAVIES, Elli PAPAEMMANUIL, Philip J. STEPHENS, Stuart MCLAREN, Adam P. BUTLER, Jon W. TEAGUE, Göran JÖNSSON, Judy E. GARBER, Daniel SILVER, Penelope MIRON, Aquila FATIMA, Sandrine BOYAULT, Anita LANGERØD, Andrew TUTT, John W.M. MARTENS, Samuel A.J.R. APARICIO, Åke BORG, Anne Vincent SALOMON, Gilles THOMAS, Anne-Lise BØRRESEN-DALE, Andrea L. RICHARDSON, Michael S. NEUBERGER, P. Andrew FUTREAL, Peter J. CAMPBELL et Michael R. STRATTON. Mutational processes molding the genomes of 21 breast cancers. *Cell*, 149(5) :979-993, 25 mai 2012. ISSN : 0092-8674. DOI : 10.1016/j.cell.2012.04.024.

[112] Marco GERLINGER, Andrew J. ROWAN, Stuart HORSWELL, M. MATH, James LARKIN, David ENDESFELDER, Eva GRONROOS, Pierre MARTINEZ, Nicholas MATTHEWS, Aengus STEWART, Patrick TARPEY, Ignacio VARELA, Benjamin PHILLIMORE, Sharmin BEGUM, Neil Q. MCDONALD, Adam BUTLER, David JONES, Keiran RAINE, Calli LATIMER, Claudio R. SANTOS, Mahrokh NOHADANI, Aron C. EKLUND, Bradley SPENCER-DENE, Graham CLARK, Lisa PICKERING, Gordon STAMP, Martin GORE, Zoltan SZALLASI, Julian DOWNWARD, P. Andrew FUTREAL et Charles SWANTON. Intratumor heterogeneity and branched evolution revealed by multiregion sequencing. *The New England journal of medicine*, 366(10) :883-892, 8 mars 2012. ISSN : 0028-4793. DOI : 10.1056/NEJMoa1113205.

[113] David H. SPENCER, Manoj TYAGI, Francesco VALLANIA, Andrew J. BREDEMEYER, John D. PFEIFER, Rob D. MITRA et Eric J. DUNCAVAGE. Performance of common analysis methods for detecting low-frequency single nucleotide variants in targeted next-generation sequence data. *The Journal of Molecular Diagnostics : JMD*, 16(1) :75-88, janvier 2014. ISSN : 1525-1578. DOI : 10.1016/j.jmoldx.2013.09.003.

[114] Brandi L CANTAREL, Daniel WEAVER, Nathan MCNEILL, Jianhua ZHANG, Aaron J MACKEY et Justin REESE. BAYSIC : a bayesian method for combining sets of genome variants with improved specificity and sensitivity. *BMC Bioinformatics*, 15 :104, 12 avril 2014. ISSN : 1471-2105. DOI : 10.1186/1471-2105-15-104.

[115] Yuichi SHIRAISHI, Yusuke SATO, Kenichi CHIBA, Yusuke OKUNO, Yasunobu NAGATA, Kenichi YOSHIDA, Norio SHIBA, Yasuhide HAYASHI, Haruki KUME, Yukio HOMMA, Masashi SANADA, Seishi OGAWA et Satoru MIYANO. An empirical bayesian framework for somatic mutation detection from cancer genome sequencing data. *Nucleic Acids Research*, 41(7) :e89, avril 2013. ISSN : 0305-1048. DOI : 10.1093/nar/gkt126.

[116] Moritz GERSTUNG, Christian BEISEL, Markus RECHSTEINER, Peter WILD, Peter SCHRAML, Holger MOCH et Niko BEERENWINKEL. Reliable detection of subclonal single-nucleotide variants in tumour cell populations. *Nature Communications*, 3 :811, 1er mai 2012. ISSN : 2041-1723. DOI : 10.1038/ncomms1814.

[117] Illumina/pisces, 18 février 2021. URL : https://github.com/Illumina/Pisces (visité le 10/03/2021). original-date : 2015-12-17T22 :45 :13Z.

[118] Kyle S. SMITH, Vinod K. YADAV, Shanshan PEI, Daniel A. POLLYEA, Craig T. JORDAN et Subhajyoti DE. SomVarIUS : somatic variant identification from unpaired tissue samples. *Bioinformatics (Oxford, England)*, 32(6) :808-813, 15 mars 2016. ISSN : 1367-4811. DOI : 10.1093/bioinformatics/btv685.

[119] Francesco VALLANIA, Enrique RAMOS, Sharon CRESCI, Robi D. MITRA et Todd E. DRULEY. Detection of rare genomic variants from pooled sequencing using SPLINTER. *Journal of Visualized Experiments : JoVE*, (64), 23 juin 2012. ISSN : 1940-087X. DOI : 10.3791/3943.

[120] Yu FAN, Liu XI, Daniel S. T. HUGHES, Jianjun ZHANG, Jianhua ZHANG, P. Andrew FUTREAL, David A. WHEELER et Wenyi WANG. MuSE : accounting for tumor heterogeneity using a sample-specific error model improves sensitivity and specificity in mutation calling from sequencing data. *Genome Biology*, 17(1) :178, 24 août 2016. ISSN : 1474-760X. DOI : 10.1186/s13059-016-1029-6.

[121] Daniel C. KOBOLDT, Ken CHEN, Todd WYLIE, David E. LARSON, Michael D. MCLELLAN, Elaine R. MARDIS, George M. WEINSTOCK, Richard K. WILSON et Li DING. VarScan : variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics*, 25(17) :2283-2285, 1er septembre 2009. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btp373.

[122] Vincent SATER, Pierre-Julien VIAILLY, Thierry LECROQ, Élise PRIEUR-GASTON, Élodie BOHERS, Mathieu VIENNOT, Philippe RUMINY, Hélène DAUCHEL, Pierre VERA et Fabrice JARDIN. UMI-VarCal : a new UMI-based variant caller that efficiently improves low-frequency variant detection in paired-end sequencing NGS libraries. *Bioinformatics*, 36(9) :2718-2724, 1er mai 2020. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btaa053.

[123] Yoav BENJAMINI et Yosef HOCHBERG. Controlling the False Discovery Rate : A Practical and Powerful Approach to Multiple Testing. en. *Journal of the Royal Statistical Society : Series B (Methodological)*, 57(1) :289-300, 1995. ISSN : 2517-6161. DOI : 10.1111/j.2517-6161.1995.tb02031.x.

[124] Yan GUO, Jiang LI, Chung-I LI, Jirong LONG, David C SAMUELS et Yu SHYR. The effect of strand bias in Illumina short-read sequencing data. *BMC Genomics*, 13 :666, novembre 2012. ISSN : 1471-2164. DOI : 10.1186/1471-2164-13-666.

[125] Yan GUO, Qiuyin CAI, David C. SAMUELS, Fei YE, Jirong LONG, Chung-I LI, Jeanette F. WINTHER, E. Janet TAWN, Marilyn STOVALL, Päivi LÄHTEENMÄKI, Nea MALIA, Shawn LEVY, Christian SHAFFER, Yu SHYR, Xiao-ou SHU et John D. BOICE. The use of Next Generation Sequencing Technology to Study the Effect of Radiation Therapy on Mitochondrial DNA Mutation. *Mutat Res*, 744(2) :154-160, mai 2012. ISSN : 0027-5107. DOI : 10.1016/j.mrgentox.2012.02.006.

[126] Gergely IVÁDY, László MADAR, Erika DZSUDZSÁK, Katalin KOCZOK, János KAPPELMAYER, Veronika KRULISOVA, Milan MACEK, Attila HORVÁTH et István BALOGH. Analytical parameters and validation of homopolymer detection in a pyrosequencing-based next generation sequencing system. *BMC Genomics*, 19, février 2018. ISSN : 1471-2164. DOI : 10.1186/s12864-018-4544-x.

[127] David E. LARSON, Christopher C. HARRIS, Ken CHEN, Daniel C. KOBOLDT, Travis E. ABBOTT, David J. DOOLING, Timothy J. LEY, Elaine R. MARDIS, Richard K. WILSON et Li DING. SomaticSniper : identification of somatic point mutations in whole genome sequencing data. *Bioinformatics*, 28(3) :311-317, février 2012. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btr665.

[128] Xuesong HU, Jianying YUAN, Yujian SHI, Jianliang LU, Binghang LIU, Zhenyu LI, Yanxiang CHEN, Desheng MU, Hao ZHANG, Nan LI, Zhen YUE, Fan BAI, Heng LI et Wei FAN. pIRS : profile-based illumina pair-end reads simulator. *Bioinformatics*, 28(11) :1533-1535, 1er juin 2012. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/bts187.

[129] Swetansu PATTNAIK, Saurabh GUPTA, Arjun A RAO et Binay PANDA. SInC : an accurate and fast error-model based simulator for SNPs, indels and CNVs coupled with a read generator for short-read sequence data. *BMC Bioinformatics*, 15 :40, 5 février 2014. ISSN : 1471-2105. DOI : 10.1186/1471-2105-15-40.

[130] Diogo PRATAS, Armando J PINHO et João M O S RODRIGUES. XS : a FASTQ read simulator. *BMC Research Notes*, 7 :40, 16 janvier 2014. ISSN : 1756-0500. DOI : 10.1186/1756-0500-7-40.

[131] Xiguo YUAN, Junying ZHANG et Liying YANG. IntSIM : an integrated simulator of next-generation sequencing data. *IEEE transactions on bio-medical engineering*, 64(2) :441-451, février 2017. ISSN : 1558-2531. DOI : 10.1109/TBME.2016.2560939.

[132] Xiguo YUAN, Meihong GAO, Jun BAI et Junbo DUAN. SVSR : a program to simulate structural variations and generate sequencing reads for multiple platforms. *IEEE/ACM transactions on computational biology and bioinformatics*, 17(3) :1082-1091, juin 2020. ISSN : 1557-9964. DOI : 10.1109/TCBB.2018.2876527.

[133] Davide BOLOGNINI, Ashley SANDERS, Jan O KORBEL, Alberto MAGI, Vladimir BENES et Tobias RAUSCH. VISOR : a versatile haplotype-aware structural variant simulator for short- and long-read sequencing. *Bioinformatics*, 36(4) :1267-1269, 15 février 2020. ISSN : 1367-4803. DOI : 10.1093/bioinformatics/btz719.

[134] Melanie SCHIRMER, Rosalinda D'AMORE, Umer Z. IJAZ, Neil HALL et Christopher QUINCE. Illumina error profiles : resolving fine-scale variation in metagenomic sequencing data. *BMC Bioinformatics*, 17(1) :125, 11 mars 2016. ISSN : 1471-2105. DOI : 10.1186/s12859-016-0976-y.

[135] Brent EWING, LaDeana HILLIER, Michael C. WENDL et Phil GREEN. Base-calling of automated sequencer traces UsingPhred. i. accuracy assessment. *Genome Research*, 8(3) :175-185, 3 janvier 1998. ISSN : 1088-9051, 1549-5469. DOI : 10.1101/gr.8.3.175.

[136] Brent EWING et Phil GREEN. Base-calling of automated sequencer traces using phred. II. error probabilities. *Genome Research*, 8(3) :186-194, 3 janvier 1998. ISSN : 1088-9051, 1549-5469. DOI : 10.1101/gr.8.3.186.

[137] Vincent SATER, Pierre-Julien VIAILLY, Thierry LECROQ, Philippe RUMINY, Caroline BÉRARD, Élise PRIEUR-GASTON et Fabrice JARDIN. UMI-gen : a UMI-based read simulator for variant calling evaluation in paired-end sequencing NGS libraries. *Computational and Structural Biotechnology Journal*, 18 :2270-2280, 1er janvier 2020. ISSN : 2001-0370. DOI : 10.1016/j.csbj.2020.08.011.

[138] Pierre-Julien VIAILLY, Vincent SATER, Mathieu VIENNOT, Elodie BOHERS, Nicolas VERGNE, Caroline BERARD, Hélène DAUCHEL, Thierry LECROQ, Alison CELEBI, Philippe RUMINY, Vinciane MARCHAND, Marie-Delphine LANIC, Sydney DUBOIS, Dominique PENTHER, Hervé TILLY, Sylvain MARESCHAL et Fabrice JARDIN. Improving high-resolution copy number variation analysis from next generation sequencing using unique molecular identifiers. *BMC Bioinformatics*, 22(1) :120, 12 mars 2021. ISSN : 1471-2105. DOI : 10.1186/s12859-021-04060-4.