Travail de fin d'études

Étude des différentes stratégies de « variant calling » en fonction du contexte diagnostic

Bouâfia Yassin

Bachelier en Biotechnique Bloc 3

HEH Département des Sciences et technologies

HEPH - Condorcet

Année académique 2023-2024

Travail de fin d'études

Étude des différentes stratégies de « variant calling » en fonction du contexte diagnostic

Bouâfia Yassin

Bachelier en Biotechnique Bloc 3

HEH – Département des Sciences et Technologies

HEPH – Condorcet

Année académique 2023-24

Remerciements

Je tiens à exprimer ma gratitude à David Coornaert, et à Aline Léonet pour leur soutien, encadrement et bien évidement leurs conseils tout au long de mon travail de fin d'études.

Table des matières

Table des illustrations

Table des tableaux

Résumé

L'objectif principal de ce travail de fin d'études est de tester différents programmes de variant calling et de les classer en fonction de leur utilité et de leur performance dans diverses situations. Deux expériences ont été menées pour atteindre cet objectif. La première expérience a impliqué la simulation de mutations dans une séquence de référence du poliovirus, suivie d'une analyse comparative des outils de variant calling pour évaluer leur précision et leur capacité à détecter ces mutations. La deuxième expérience s'est concentrée sur l'analyse des données séquencées réelles du virus Monkeypox, en utilisant divers logiciels de variant calling pour identifier les variants présents et en annoter les impacts potentiels sur les protéines codées. Les résultats de ces deux expériences ont permis d'identifier les forces et les faiblesses des différents outils, fournissant un guide pratique pour leur utilisation en fonction des caractéristiques spécifiques des données et des besoins de recherche.

Summary

The primary goal of this thesis is to test various variant calling programs and rank them based on their utility and performance in different scenarios. Two experiments were conducted to achieve this goal. The first experiment involved simulating mutations in a reference sequence of the poliovirus, followed by a comparative analysis of variant calling tools to assess their accuracy and ability to detect these mutations. The second experiment focused on analyzing real sequenced data from the Monkeypox virus, using different variant calling software to identify the present variants and annotate their potential impacts on the encoded proteins. The results of these two experiments provided insights into the strengths and weaknesses of the various tools, offering a practical guide for their use based on specific data characteristics and research needs.

1. Introduction

1.1. Contexte

Le séquençage de nouvelle génération, ou next-generation sequencing (NGS), a révolutionné les sciences biomédicales en offrant de nouvelles possibilités d'exploration du génome humain et de nombreux autres organismes. Apparue au début des années 2000, cette technologie a profondément modifié les domaines de la génomique, de la transcriptomique et de la métagénomique en rendant le séquençage à grande échelle plus rapide et plus abordable. Désormais, il est possible de séquencer des génomes entiers en quelques jours, alors que cela prenait des années avec les anciennes méthodes de séquençage, comme celles utilisées pour le projet du génome humain.

Au-delà de la simple accumulation de données, le NGS a permis l'émergence de nouvelles méthodes d'analyse et d'interprétation génomiques. Parmi celles-ci, le « *variant calling* », qui consiste à identifier les variations génétiques en comparant un échantillon à un génome de référence, a largement bénéficié des progrès technologiques. Les algorithmes modernes de traitement des données issues du NGS permettent de détecter avec précision divers types de variations génétiques, comme les SNPs, les indels, et des variations structurelles plus complexes. Ces analyses sont indispensables non seulement pour la recherche génétique et le diagnostic clinique, mais aussi pour des applications clés en vaccinologie.

En vaccinologie, le variant calling a pris une importance particulièrement importante avec la pandémie de COVID-19 et le développement rapide des vaccins à ARN. Cette technologie a été cruciale pour suivre en temps réel les mutations du virus SARS-CoV-2 et adapter rapidement les vaccins aux nouvelles variantes qui ont émergé. Les vaccins à ARN, tels que ceux développés par Pfizer-BioNTech ou Moderna, ont la particularité de pouvoir être ajustés rapidement en réponse aux mutations du virus. Grâce au variant calling, il est possible de détecter rapidement les mutations susceptibles d'altérer l'efficacité des vaccins, permettant ainsi de réagir promptement. Par exemple, l'émergence des variants Alpha, Delta et Omicron a nécessité une surveillance génétique continue pour évaluer l'efficacité des vaccins en usage et guider la formulation de nouvelles doses de rappel. En outre, le variant calling aide à comprendre comment certaines mutations peuvent affecter la transmissibilité du virus et la résistance aux vaccins, ce qui est crucial pour élaborer des stratégies de santé publique et des plans de vaccination plus efficaces. Cette technologie permet donc de prévenir plus précisément les maladies infectieuses, d'optimiser l'efficacité des vaccins et de minimiser les risques que le virus échappe à l'immunité.

Cependant, le variant calling n'est pas sans défis. La complexité biologique des échantillons, comme les régions répétitives du génome ou l'hétérogénéité qui peut apparaitre dans certaines régions, peut nuire à la précision des analyses, nécessitant des approches bioinformatiques de plus en plus sophistiquées. De plus, bien que le coût du séquençage ait baissé, la gestion, le stockage et l'analyse des vastes volumes de données générés par le NGS restent des obstacles importants, tant sur le plan logistique que financier.

1.2. Objectifs du travail

Le but principal de ce travail est de réaliser un état de l'art exhaustif des différents programmes de variant calling, en analysant en profondeur leurs spécificités, leurs

performances et leurs domaines d'application optimaux. Cette évaluation vise à identifier les outils les plus adaptés selon divers contextes d'utilisation, en tenant compte des types de variants recherchés (SNPs, indels, variants structuraux), des types de séquences (ADN, ARN), et des caractéristiques des échantillons (génomes humains, viraux, bactériens, échantillons environnementaux). Une telle analyse détaillée permettra de mettre en lumière non seulement les forces et les faiblesses de chaque méthode, mais aussi de proposer des recommandations précises pour leur application en recherche fondamentale et en diagnostic clinique.

Dans un premier temps, l'analyse portera sur une revue théorique des logiciels de variant calling les plus utilisés et reconnus dans le domaine de la génomique. Cette revue examinera les caractéristiques de chaque programme en fonction de plusieurs critères importants, tels que le type de données de séquençage (ADN, ARN, viral, bactérien, métagénomique), la disponibilité d'un génome de référence, et les types de variants recherchés (SNPs, indels, insertions, délétions, variations structurelles). En tenant compte de la qualité du séquençage, de la complexité des génomes, et de la nature des échantillons, cette analyse permettra de comparer les logiciels sur des critères tels que la précision des appels de variants, la sensibilité, la spécificité, la gestion des erreurs de séquençage, et la capacité à détecter des variants de faible fréquence. Ce benchmarking théorique s'appuiera sur des données issues de la littérature scientifique pour dresser un panorama complet des technologies disponibles et de leurs performances respectives dans divers contextes d'analyse génomique.

Ensuite, pour vérifier si les performances théoriques se traduisent effectivement en pratique, deux études de cas pratiques distinctes seront réalisées. La première expérience utilisera des données simulées, permettant de contrôler au maximum les paramètres expérimentaux. Cette approche aidera à évaluer si les outils de variant calling peuvent reproduire les performances théoriques dans un environnement où les variables sont strictement définies, telles que la complexité génomique, les taux d'erreurs de séquençage et la fréquence des variants. La deuxième expérience appliquera les outils de variant calling à des données réelles, en utilisant des reads issus de la base de données NCBI. Cette étude visera à observer si les performances théoriques des outils se maintiennent dans des conditions expérimentales avec des échantillons présentant une diversité génomique et une hétérogénéité biologique, ce qui est crucial pour leur application en recherche et en diagnostic clinique.

En combinant une revue théorique approfondie avec ces deux validations expérimentales, ce travail vise à fournir des recommandations éclairées et fondées sur des données pour le choix et l'utilisation des outils de variant calling. Ces recommandations devraient aider à choisir les outils les plus adaptés à des besoins spécifiques, mais aussi à confirmer ou à remettre en question les performances théoriques.

2. État de l'art

2.1. Le séquençage de nouvelle génération (NGS)

Le séquençage de nouvelle génération (NGS), ou séquençage à haut débit, regroupe un ensemble de technologies qui ont transformé l'analyse de l'ADN et de l'ARN en permettant le séquençage massif et parallèle de multiples fragments génétiques. Contrairement aux méthodes de séquençage de première génération, comme le séquençage Sanger, qui analysaient les séquences d'ADN une par une, les technologies NGS permettent de traiter simultanément des millions de fragments d'ADN. Cette section examine les principales technologies et plateformes utilisées pour le NGS, en détaillant leurs avantages ainsi que les défis qu'elles posent.

Les technologies de NGS disponibles aujourd'hui se distinguent par leur capacité à produire de grandes quantités de données rapidement et avec une précision accrue. Parmi les plateformes dominantes sur le marché, trois se démarquent par leurs applications spécifiques et leurs caractéristiques techniques :

Technologie	Principe	Avantages	Inconvénients	Applications Idéales
Illumina	Séquençage par synthèse avec terminaison réversible	- Haute précision - Faible taux d'erreur - Coût réduit par base séquencée	- Lectures courtes	Séquençage de génomes entiers, exomes, profilage d'expression génique
Oxford Nanopore	Détection des variations de courant électrique à travers une nanopore	- Lectures très longues - Capacité à séquencer des régions complexes du génome	- Taux d'erreur plus élevé	Séquençage de novo, séquençage de régions complexes du génome
Pacific Biosciences	Séquençage de molécules individuelles en temps réel (SMRT)	- Lectures longues - Amélioration récente de la fidélité - Détection de variants complexes et modifications épigénétiques	- Taux d'erreur initialement élevé (amélioré avec les dernières avancées)	Détection de variants complexes, analyse des modifications épigénétiques

Tableau 1: Les différentes technologies des NGS

2.2. Le « variant calling »

Le « variant calling » est un processus bioinformatique essentiel dans l'analyse des données issues du séquençage de nouvelle génération. Ce processus permet d'identifier les variations génétiques, telles que les polymorphismes nucléotidiques simples (SNPs), les insertions et délétions (indels), les variations structurelles et les variations du nombre de copies (CNVs), par rapport à un génome de référence ou à un assemblage de novo. Cette étape joue un rôle crucial en génomique pour explorer la diversité génétique, diagnostiquer des maladies, et développer des traitements médicaux personnalisés. Une identification précise de ces variations est fondamentale pour comprendre les mécanismes génétiques sous-jacents aux pathologies, étudier les processus évolutifs, et concevoir des stratégies thérapeutiques spécifiques.

- Polymorphismes nucléotidiques simples (SNPs)
  Les polymorphismes nucléotidiques simples (SNPs) sont des variations fréquentes dans le génome humain, caractérisées par le remplacement d'un nucléotide par un autre dans une séquence d'ADN. Lorsqu'ils se situent dans des régions codantes, les SNPs peuvent être :
  - Synonymes : Ils n'affectent pas la séquence d'acides aminés de la protéine en raison de la redondance du code génétique, mais peuvent influencer des processus biologiques comme l'épissage de l'ARNm ou la stabilité de l'ARN.
  - Non synonymes : Ils modifient la séquence d'acides aminés, avec des variants faux-sens (missense) qui altèrent la fonction de la protéine et peuvent causer des dysfonctionnements cellulaires. Les variants non-sens (nonsense) introduisent un codon stop prématuré, produisant des protéines tronquées et non fonctionnelles, souvent associées à des maladies graves.

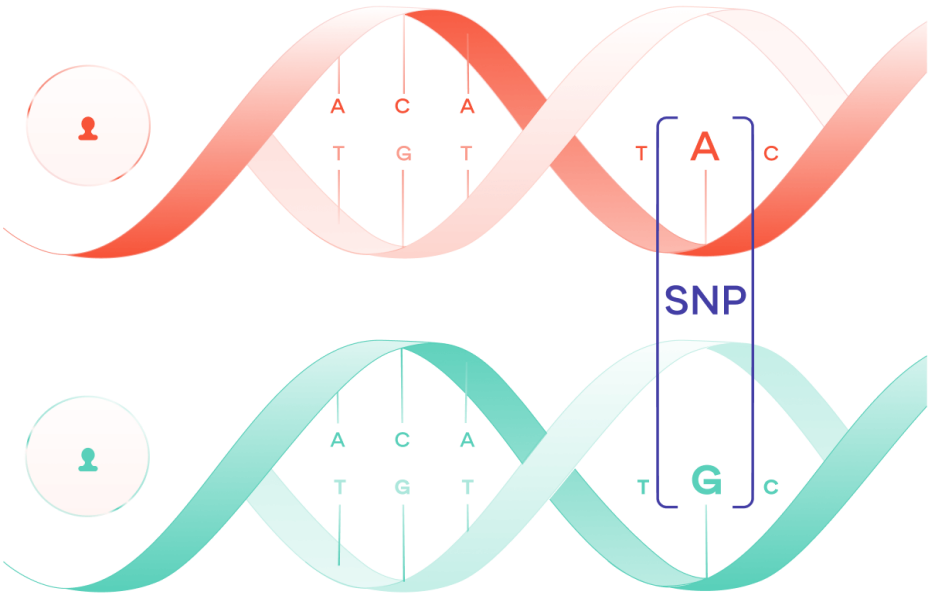


Figure 1: Illustration d'un SNP

- Insertions et délétions (Indels)
  Les indels, ou insertions et délétions, sont des variations génétiques où de courtes séquences de nucléotides sont ajoutées ou supprimées dans l'ADN. Ces mutations

Figure 2: Illustration d'un indels (délation)

  peuvent provoquer un décalage du cadre de lecture (frameshift), modifiant ainsi la structure et la fonction de la protéine, souvent avec des conséquences majeures. Les indels sont souvent liés à des maladies héréditaires, et leur détection est essentielle pour diagnostiquer des troubles génétiques et comprendre les mécanismes moléculaires sous-jacents. Bien que moins fréquents que les SNPs, les indels ont généralement des impacts plus importants, surtout lorsqu'ils affectent des régions codantes, pouvant entraîner des dysfonctionnements cellulaires graves.

- Variations structurelles complexes
  Les variations structurelles sont des réarrangements génomiques majeurs qui incluent des duplications, des inversions, des translocations et des amplifications de grands segments d'ADN. Ces modifications peuvent avoir des conséquences importantes sur l'intégrité et la fonction du génome, car elles peuvent affecter simultanément plusieurs gènes ou régions régulatrices.

- Variations du nombre de copies (CNVs)
  Les variations du nombre de copies, ou CNVs, sont des segments d'ADN dont le nombre de copies diffère par rapport au génome de référence. Ces variations peuvent correspondre à des gains (duplications) ou des pertes (délétions) de fragments d'ADN de grande taille, pouvant s'étendre sur plusieurs kilobases à mégabases. Les CNVs jouent un rôle crucial dans la diversité génétique humaine et sont fréquemment associées à des maladies complexes, comme les troubles neurodéveloppementaux (par exemple, l'autisme) et divers types de cancer. En modifiant le dosage de certains gènes, les CNVs peuvent altérer l'expression génique et perturber des réseaux génétiques entiers, ce qui peut conduire à des phénotypes pathologiques et contribuer au développement de maladies.

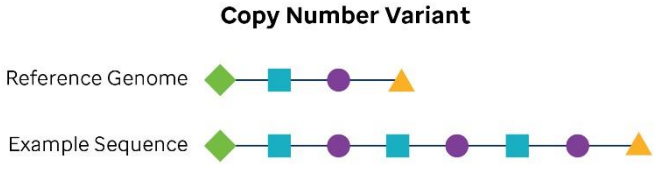


Figure 3: Illustration d'un CNV

2.2.1. Processus général du « variant calling »

Le workflow de variant calling comprend plusieurs étapes clés, qui vont de la préparation des données de séquençage à l'interprétation des variants détectés.

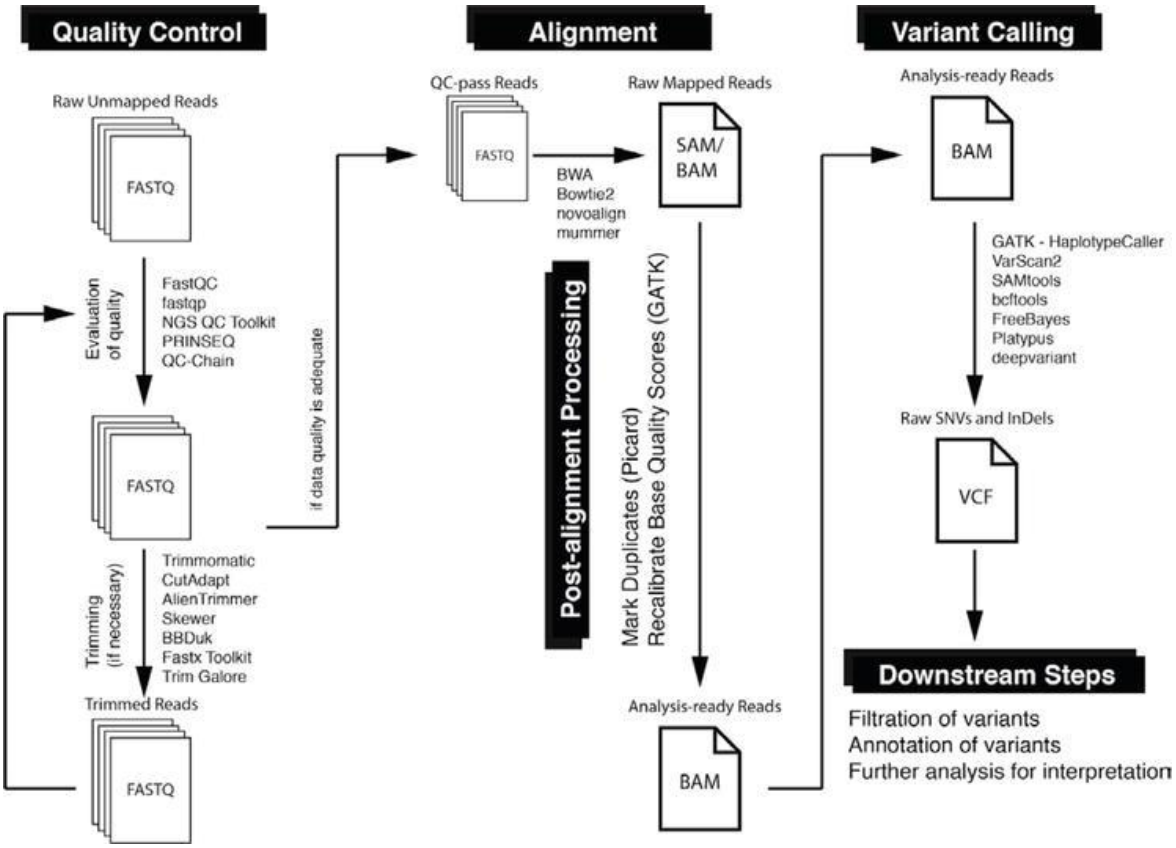


Figure 4: Worflow "basique" de variant calling

2.2.1.1. Alignement ou Assemblage des Séquences

Lors de l'analyse des données de séquençage, deux approches principales sont utilisées pour interpréter les séquences d'ADN : l'alignement sur un génome de référence et l'assemblage de novo. Chacune de ces méthodes a des applications spécifiques et est choisie en fonction des objectifs de l'étude et de la disponibilité des données de référence.

Alignement sur un génome de référence :

L'alignement consiste à placer les séquences d'ADN, appelées "reads", obtenues par séquençage, sur un génome de référence préexistant pour déterminer leur position précise. Cette technique est essentielle pour identifier les variations génétiques, en particulier dans les régions du génome qui présentent des complexités telles que des séquences répétitives ou des motifs génétiques similaires. Les outils utilisés pour l'alignement, tels que BWA, Bowtie et HISAT2, sont conçus pour optimiser à la fois la précision et l'efficacité du processus, même dans les zones à haute complexité génomique. Grâce à ces outils, il est possible de cartographier rapidement et précisément les reads sur un génome de référence, facilitant ainsi l'identification des variations génétiques importantes.

Assemblage de novo :

L'assemblage de novo est utilisé lorsque le génome de l'organisme étudié n'est pas disponible ou est incomplet. Plutôt que de s'appuyer sur une référence préexistante, cette méthode reconstruit le génome à partir des fragments séquencés en les assemblant comme les pièces d'un puzzle. Cette approche est particulièrement utile pour explorer des organismes peu caractérisés ou pour découvrir de nouvelles séquences génomiques. Les outils d'assemblage de novo, comme SPAdes, permettent de reconstituer des génomes complets avec une précision élevée, même en l'absence d'une référence. Ces outils sont capables de gérer les complexités inhérentes à l'assemblage, telles que la couverture inégale et les régions répétitives, fournissant ainsi une base solide pour la détection de variantes génétiques et l'étude de la diversité génomique.

2.2.1.2. Pré-traitement et Vérification de la Qualité des Lectures

Évaluation et correction des erreurs de séquençage :

Avant de procéder au variant calling, il est essentiel de recalibrer les scores de qualité des lectures de séquençage pour minimiser les erreurs potentielles. Ce processus de recalibration utilise des modèles statistiques pour ajuster les scores de qualité en fonction de la probabilité d'erreur, ce qui permet de corriger les erreurs systématiques introduites par les technologies de séquençage. Une correction précise des scores de qualité est cruciale pour garantir l'intégrité des données de séquençage et réduire les erreurs lors de l'identification des variations génétiques. Des outils spécialisés comme GATKBaseRecalibrator sont couramment utilisés pour effectuer cette étape de pré-traitement, optimisant ainsi la qualité globale des données avant l'analyse des variants.

Filtrage des lectures et élimination des duplicats :

Une fois les erreurs de séquençage recalibrées, il est nécessaire d'appliquer des filtres supplémentaires pour améliorer la qualité des données. Ces filtres éliminent les lectures de faible qualité et les duplicats de séquençage, qui peuvent fausser les résultats et introduire des biais lors de la détection des variants. En nettoyant les données de ces éléments indésirables, on améliore la précision de l'appel des variants et on réduit le risque de faux positifs, garantissant ainsi des résultats plus fiables. Ce processus de filtrage et d'élimination des duplicats est une étape clé pour assurer la robustesse des analyses de séquençage et la validité des conclusions tirées des études génomiques.

2.2.1.3. Détection des variants

Identification des variants :

Après l'alignement des séquences et la vérification de la qualité des données, l'étape suivante consiste à analyser les différences entre les lectures alignées et le génome de référence, ou un assemblage de novo, pour identifier les variants potentiels. Cette analyse se base sur plusieurs facteurs, notamment la profondeur de couverture des séquences, la cohérence des

variations détectées à travers les lectures multiples, et la fiabilité des régions génomiques où ces variations sont observées. En tenant compte de ces éléments, il est possible de repérer avec précision les différences génétiques significatives, qui peuvent correspondre à des mutations importantes pour l'étude.

Filtrage et validation des variants

Une fois les variants potentiels identifiés, il est crucial de les filtrer et de les valider à l'aide de critères statistiques rigoureux. Cette étape de filtrage utilise des mesures telles que le score de qualité (qual score), la profondeur de lecture (qui indique combien de fois une région donnée a été séquencée), et la fréquence allélique (la proportion d'un certain variant parmi toutes les séquences étudiées). Ces critères permettent de distinguer les vrais variants des artefacts de séquençage ou des erreurs de données. En appliquant ces filtres, on réduit le nombre de faux positifs et de faux négatifs, garantissant ainsi la fiabilité et la reproductibilité des résultats obtenus lors de l'analyse génomique.

2.2.1.4. Annotation et interprétation des variants

Annotation des variants :

Une fois les variants détectés, il faut les annoter pour déterminer leur localisation précise dans le génome et évaluer leurs effets potentiels sur les structures géniques. Cette annotation permet d'identifier si les variants se trouvent dans des exons, des introns, des régions promotrices, ou d'autres éléments fonctionnels du génome. Elle aide également à comprendre les implications possibles des variants pour les phénotypes et les pathologies. Pour réaliser cette tâche, les chercheurs utilisent fréquemment des bases de données comme dbSNP, ClinVar, et OMIM, ainsi que des outils d'annotation tels qu'ANNOVAR et SnpEff, qui fournissent des informations détaillées sur les effets connus ou présumés des variants.

Interprétation biologique et clinique :

Après l'annotation, les variants doivent être interprétés pour évaluer leur importance biologique et clinique. Cette étape comprend l'analyse de l'impact fonctionnel des variants sur les protéines codées, ainsi que la prédiction de leurs effets possibles sur divers processus biologiques ou sur le développement de conditions pathologiques. Cette interprétation est essentielle pour orienter les recherches futures et informer les applications cliniques, notamment dans le cadre du développement de thérapies personnalisées. En comprenant mieux la signification des variants identifiés, les scientifiques peuvent concevoir des stratégies de traitement plus ciblées et adaptées aux besoins individuels des patients.

2.2.1.5. Validation Expérimentale (si nécessaire)

Validation par des méthodes orthogonales : Dans certains cas, en particulier pour les variants rares ou de grande importance clinique, une validation expérimentale supplémentaire peut être nécessaire. Des techniques telles que le séquençage Sanger ou le PCR spécifique d'allèle

sont utilisées pour confirmer les résultats du variant calling, assurant une haute confiance dans les résultats obtenus.

2.3. Les outils et logiciels du « variant calling »

Dans le domaine du séquençage de nouvelle génération (NGS), le choix des outils bioinformatiques pour le variant calling est crucial pour l'analyse précise des données génomiques. Les recherches sur PubMed et Google Scholar montrent une utilisation variée de plusieurs logiciels de variant calling, chacun ayant ses propres avantages et applications spécifiques en fonction des types de données et des contextes d'analyse.

2.3.1. Analyse des tendances de publication

Dans le domaine du séquençage de nouvelle génération (NGS), divers outils bioinformatiques sont utilisés pour le variant calling, chacun ayant des avantages spécifiques en fonction du type de données et des besoins de l'analyse. Les tendances de publication sur PubMed révèlent une utilisation variée de ces logiciels, ce qui reflète leur pertinence et leur adoption dans différentes applications de recherche génomique.

GATK (Genome Analysis Toolkit) est le logiciel de variant calling le plus utilisé, avec 317 publications sur PubMed. Depuis sa sortie en 2010, GATK a été régulièrement mis à jour et est devenu incontournable pour la détection des SNPs et des indels grâce à son algorithme HaplotypeCaller. Sa popularité et sa robustesse sont soulignées par les 172 publications depuis 2019 sur PubMed, dont 75 depuis 2022, ce qui démontre son adaptabilité aux nouvelles technologies et méthodes de séquençage.

SAMtools et BCFtools sont également largement utilisés, principalement pour la manipulation et le filtrage des fichiers de séquençage tels que BAM/CRAM et VCF/BCF. SAMtools est mentionné dans 119 publications sur PubMed, avec 39 publications depuis 2019, indiquant son importance continue dans les workflows de génomique. BCFtools, avec 31 publications sur PubMed, reste essentiel pour les tâches de préparation des données avant le variant calling.

DeepVariant, introduit par Google en 2017, utilise l'apprentissage profond pour améliorer la précision du variant calling. Il a été cité dans 36 publications sur PubMed, dont 33 depuis 2019, ce qui montre son adoption croissante pour les études nécessitant une détection de variants très précise.

FreeBayes est un outil apprécié pour l'analyse de populations génétiques complexes, avec 51 publications sur PubMed. Bien qu'il soit moins utilisé que GATK ou SAMtools/BCFtools, FreeBayes reste précieux pour les études nécessitant une gestion des génotypes complexes et des variants multi-alleliques, avec 24 publications depuis 2019.

iVar, principalement utilisé pour le séquençage viral, a vu une utilisation accrue pendant la pandémie de COVID-19. Avec 16 publications sur PubMed, dont 7 depuis 2019, iVar s'est avéré crucial pour le séquençage du SARS-CoV-2 et les études sur les virus émergents.

VarScan et LoFreq sont des outils spécialisés pour la détection de variants à faible fréquence, respectivement dans les études oncologiques et la recherche sur la résistance aux

médicaments antiviraux. VarScan a été mentionné dans 39 publications sur PubMed, tandis que LoFreq en compte 26, ce qui démontre leur pertinence dans des niches spécifiques.

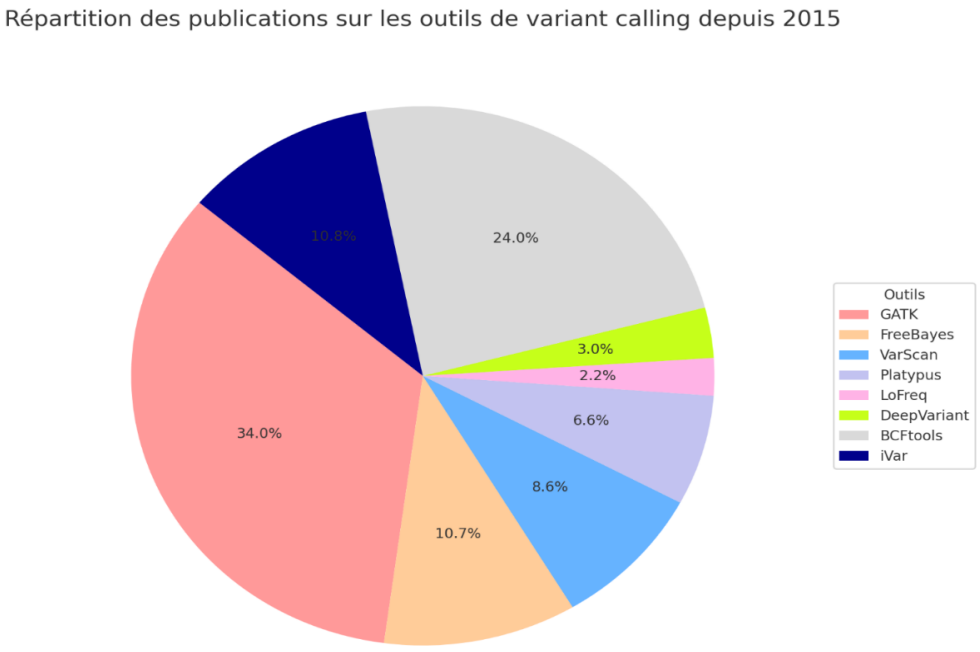


Figure 5: Graphique de la répartition des différents outils de variant calling

Enfin, bien que certains outils de variant calling soient moins cités dans la littérature, ils répondent souvent à des besoins spécifiques dans des contextes de recherche de niche ou pour des types de données particuliers, prouvant ainsi leur valeur dans des applications ciblées.

2.3.2. Résumé des différents programmes de variant calling

Une recherche succincte sur les logiciels de variant calling les plus utilisés a permis de créer ce tableau récapitulatif, mettant en lumière les caractéristiques et les spécificités de chaque outil. Ces logiciels, développés entre 2009 et 2018, reflètent l'évolution rapide des technologies de séquençage de nouvelle génération (NGS) et des besoins croissants en bioinformatique. Les dates de sortie varient, avec des outils comme SAMtools/BCFtools et VarScan, publiés dès 2009, représentant les premières générations de logiciels de variant calling, tandis que DeepVariant et iVar, plus récents, montrent l'adoption de méthodes avancées comme l'apprentissage profond pour améliorer la précision du variant calling. L'analyse des dates de sortie et des versions actuelles démontre que le développement et la mise à jour continue de ces outils sont essentiels pour répondre aux exigences changeantes de la recherche génomique.

Tableau 2: Tableau descriptif des différents programmes de variant calling

Outil	Description	Date de sortie	Version actuelle	Systèmes d'exploitation	Site officiel
GATK	Outil polyvalent pour le variant calling avec des algorithmes avancés comme HaplotypeCaller.	2010	4.5.0	Unix/Linux, macOS	GATK Broad Institute
FreeBayes	Logiciel adapté pour l'analyse de populations complexes, gérant efficacement les poly-haploïdes et les variants multi-alléliques.	2011	1.3.8	Unix/Linux, macOS	FreeBayes GitHub
SAM-tools/BCF-tools	Outils fondamentaux pour la manipulation de fichiers de séquençage, indispensables pour la préparation des données.	2009	1.20	Unix/Linux, macOS	HTSlib
VarScan	Spécialisé dans la détection de mutations somatiques et germinales à faible fréquence, idéal pour les études de cancer.	2009	2.4.6	Unix/Linux, macOS	VarScan SourceForge
LoFreq	Outil statistique pour la détection de variants à faible fréquence, adapté aux études virales et de résistance aux médicaments.	2012	2.1.5	Unix/Linux, macOS	LoFreq GitHub
DeepVariant	Utilise l'apprentissage profond pour modéliser les erreurs de séquençage, offrant une précision élevée pour la détection des variants.	2017	1.6.1	Unix/Linux, macOS	DeepVariant GitHub
iVar	Spécialisé dans l'analyse de séquençage viral, offrant des outils pour la détection de variants et la génération de consensus.	2018	1.4.2	Unix/Linux, macOS	iVar GitHub
Platypus	Utilisé pour des analyses rapides et la gestion de variants complexes dans des ensembles de données volumineux.	2014	0.8.1	Unix/Linux, macOS	Platypus GitHub

2.3.3. Choix des logiciels de variant calling selon le contexte d'étude

Dans l'analyse génomique, le choix des logiciels de variant calling est essentiel pour garantir une détection précise et fiable des variations génétiques. Chaque logiciel de variant calling présente des performances différentes selon plusieurs facteurs clés, tels que le type de données utilisées (comme l'ADN, l'ARN, ou des données virales, bactériennes et métagénomiques), la disponibilité d'un génome de référence, le type de variants recherchés (SNPs, indels, CNVs, variations structurelles), ainsi que le niveau de connaissance préalable des variations existantes. Ce sous-chapitre explore ces paramètres afin de faciliter le choix des outils de variant calling en fonction des besoins spécifiques de chaque étude.

2.3.3.1. Les types de données

Le choix du logiciel de variant calling dépend fortement du type de données de séquençage disponibles, qu'il s'agisse de données d'ADN, d'ARN, virales, bactériennes ou métagénomiques. Chaque type de données présente des défis uniques, et le logiciel sélectionné doit être capable de traiter ces spécificités pour garantir une analyse précise.

Pour les données d'ADN, les outils tels que GATK et DeepVariant sont souvent privilégiés. GATK utilise l'algorithme HaplotypeCaller, reconnu pour sa grande précision dans la détection des SNPs et des indels, grâce à ses modèles statistiques robustes et à ses nombreuses options d'optimisation des données d'entrée. DeepVariant, quant à lui, repose sur des techniques d'apprentissage profond pour améliorer la précision du variant calling, particulièrement utile pour détecter des variants rares ou situés dans des régions génomiques complexes.

En ce qui concerne les données d'ARN (RNA-seq), qui nécessitent une prise en compte des introns et des exons en raison de l'épissage, FreeBayes et SAMtools/BCFtools sont couramment utilisés. FreeBayes est efficace pour identifier des variants dans des échantillons polyploïdes et hétérogènes, offrant une flexibilité nécessaire pour traiter la complexité des données d'ARN. SAMtools/BCFtools est très apprécié pour ses capacités de filtrage et de manipulation des fichiers BAM/CRAM, essentielles pour préparer les données avant le variant calling.

Pour les données virales, où la couverture peut être inégale et où la détection de variants à faible fréquence est cruciale, des outils comme LoFreq et iVar sont particulièrement adaptés. LoFreq utilise des modèles statistiques qui prennent en compte la qualité des séquences pour détecter des variants à très faible fréquence, ce qui est vital pour les études sur la résistance aux médicaments antiviraux. iVar se distingue par ses fonctionnalités spécialisées, telles que la gestion des amplicons, ce qui le rend idéal pour les études sur les virus émergents, y compris celles menées durant la pandémie de COVID-19.

Pour les données bactériennes, qui se caractérisent par une grande diversité génétique au sein des populations, FreeBayes et SAMtools/BCFtools sont également recommandés. FreeBayes, avec son approche probabiliste, est particulièrement efficace pour analyser des populations bactériennes complexes. De leur côté, SAMtools/BCFtools sont essentiels pour la manipulation des données de séquençage, permettant de gérer efficacement les grands volumes de données souvent générés dans les études bactériennes.

Enfin, pour les données métagénomiques, impliquant l'analyse de communautés microbiennes complexes, des outils capables de gérer une diversité génétique considérable et des échantillons mixtes sont indispensables. FreeBayes est bien adapté à cette tâche grâce à sa capacité à analyser des génotypes complexes et des variants multi-alleliques. SAMtools/BCFtools jouent également un rôle clé dans le filtrage et la préparation des données, garantissant une manipulation efficace des grands ensembles de données typiques des études métagénomiques.

2.3.3.2. Présence ou non du génome de référence

Le choix du logiciel de variant calling est largement influencé par la disponibilité d'un génome de référence. Lorsque ce dernier est disponible, des outils tels que GATK, DeepVariant et SAMtools/BCFtools sont particulièrement efficaces. GATK et DeepVariant utilisent le

génome de référence pour aligner les séquences avec précision et identifier les variants en s'appuyant sur des modèles fondés sur une connaissance approfondie du génome. Cette approche permet une détection très précise des variations génétiques, en optimisant l'utilisation des données préexistantes.

En revanche, lorsqu'aucun génome de référence n'est disponible, l'analyse devient plus complexe, nécessitant un assemblage de novo pour reconstruire le génome de l'organisme étudié. Dans ce contexte, FreeBayes est souvent privilégié pour sa flexibilité dans l'analyse des populations génétiques complexes, sans nécessiter de génome de référence. SAMtools/BCFtools restent également essentiels, car ils permettent de gérer efficacement les données issues de l'assemblage de novo, en offrant des fonctionnalités de tri, de filtrage et de conversion des données qui sont cruciales pour le variant calling dans ces situations.

2.3.3.3. Types de variants recherchés

Le type de variants génétiques recherchés est un autre facteur crucial qui influence le choix du logiciel de variant calling. Pour la détection des SNPs et des indels, GATK et DeepVariant sont largement reconnus comme les meilleurs outils, grâce à leurs algorithmes sophistiqués qui optimisent la précision et la sensibilité. DeepVariant, en particulier, utilise des techniques d'apprentissage profond pour améliorer la détection des SNPs, même dans des régions génomiques complexes.

Pour la détection des insertions et délétions (indels), VarScan et FreeBayes sont des choix solides. VarScan est particulièrement adapté pour identifier des mutations somatiques dans des environnements à faible fréquence allélique, ce qui est essentiel pour les études oncologiques. FreeBayes, avec son approche bayésienne, est efficace pour détecter des indels dans des contextes où les variants multi-alleliques sont fréquents.

La détection des variations structurelles, telles que les duplications, inversions et translocations, requiert des outils capables de gérer des réarrangements génomiques complexes. Bien que les outils mentionnés précédemment, comme GATK et DeepVariant, soient principalement conçus pour les SNPs et les indels, DeepVariant a démontré une certaine capacité à identifier des variations structurelles dans des études récentes, grâce à ses algorithmes d'apprentissage profond. Cependant, pour une détection plus spécialisée des variations structurelles, d'autres outils comme Lumpy ou Manta peuvent être plus adaptés.

2.3.3.4. Tableau récapitulatif

Tableau 3: Tableau récapitulatif des logiciels de varaint calling en fonction du contexte

Critère	Contexte spécifique	Logiciel recommandé
Type de données	ADN	GATK, DeepVariant
	ARN	FreeBayes, SAMtools/BCF-tools
	Viral	LoFreq, iVar
	Bactérien	FreeBayes, SAMtools/BCF-tools
	Métagénomique	FreeBayes, SAMtools/BCF-tools
Présence ou absence de génome de référence	Présence d'un génome de référence	GATK, DeepVariant, SAM-tools/BCFtools
	Absence de génome de référence	FreeBayes, SAMtools/BCF-tools
Types de variants recherchés	SNPs et Indels	GATK, DeepVariant
	Insertions et Délétions (Indels)	VarScan, FreeBayes
	Variations structurelles	DeepVariant

Le tableau met en lumière l'importance de choisir le logiciel de variant calling en fonction du contexte spécifique de l'analyse génomique. Selon le type de données, la présence d'un génome de référence, et les types de variants recherchés, certains outils se révèlent plus adaptés que d'autres. Pour les données complexes ou hétérogènes, des logiciels flexibles capables de gérer différentes structures génomiques sont indispensables. En revanche, lorsqu'un génome de référence est disponible, les outils qui exploitent cette information offrent généralement une meilleure précision dans la détection des variants. Enfin, le choix du logiciel doit également tenir compte des types de variations génétiques à identifier, qu'il s'agisse de mutations ponctuelles, d'insertion/délétions ou de réarrangements structurels plus complexes. Cette diversité d'outils et de capacités souligne la nécessité d'une sélection attentive pour répondre aux objectifs spécifiques de chaque étude génomique.

3. Matériels et méthodes

3.1. Configuration matérielle

L'étude a été réalisée sur une station de travail HP ZBook Power G7 Mobile Workstation, équipée de 16 GB de mémoire RAM et d'un processeur Intel i7-10850H à 2,7 GHz (jusqu'à 5,1 GHz en mode Turbo Boost), disposant de 6 cœurs physiques et 12 cœurs logiques. Un disque SSD NVMe de 256 GB a été utilisé pour le stockage, offrant des vitesses de lecture séquentielle jusqu'à 2,338 MBytes/sec et d'écriture séquentielle jusqu'à 1,716 MBytes/sec. Le système d'exploitation Ubuntu Linux 24.04 a été utilisé ainsi que les dernières versions stables disponibles de chaque logiciel.

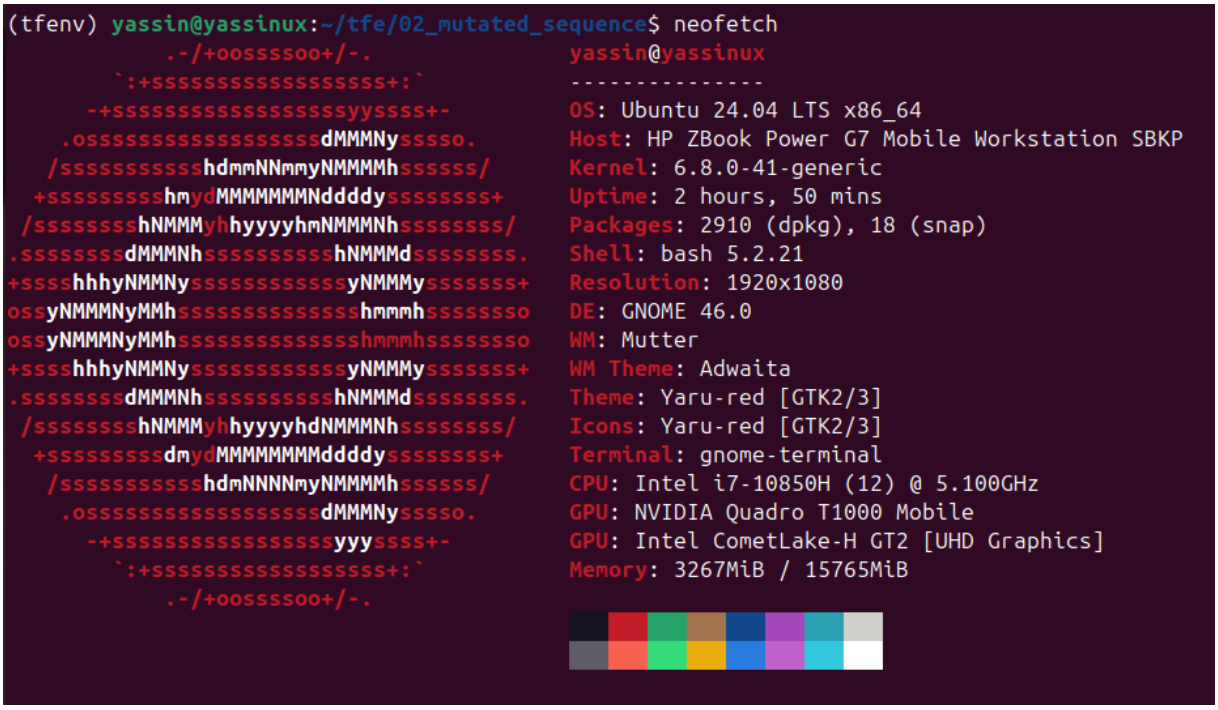


Figure 6: Capture d'écran "neofetch" sur les caractéristiques de l'ordinateur

3.2. Expérience 1

L'objectif principal de cette étude est de vérifier si les résultats théoriques et les attentes issues de la littérature scientifique concordent avec les résultats obtenus dans une expérience pratique. Pour cela, une séquence complète du génome du poliovirus de type 1, souche Mahoney, a été choisie comme base de référence.

3.2.1. Téléchargement de la séquence originale

La séquence utilisée pour cette étude est le génome complet du poliovirus de type 1, souche Mahoney, référencée sous l'accession V01149.1. Ce génome est une séquence d'ARN linéaire de 7 440 bases, codant pour un polyprotéine qui est ensuite clivée en plusieurs protéines fonctionnelles, essentielles pour le cycle de vie du virus. La séquence inclut des régions codantes pour les protéines de la capside, le génome viral, ainsi que des

éléments régulateurs nécessaires à la réplication du virus. Classé sous les Picornaviridae dans l'ordre des Picornavirales, ce virus est un membre du genre Enterovirus, spécifiquement Enterovirus C. Ce génome de référence a été largement étudié et sert de base pour comprendre la structure, l'organisation génique, et l'expression des polypeptides du poliovirus, ainsi que pour les études de mutation et de variant calling.

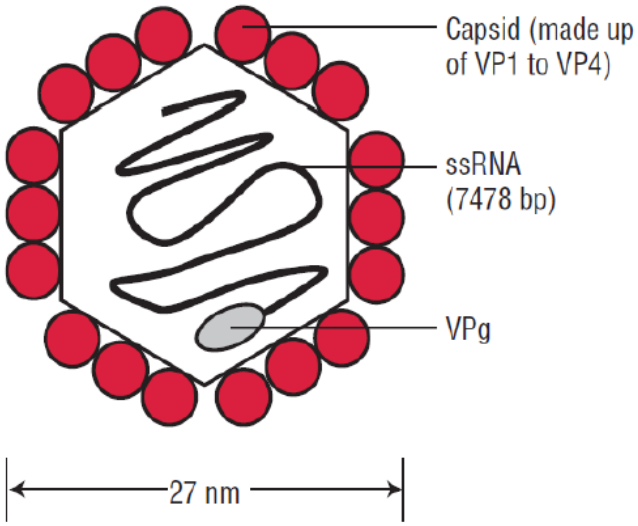


Figure 7: Schéma illustratif de la structure du poliovirus

3.2.2. Introduction des mutations

Pour simuler les mutations observées dans des contextes de variabilité génétique naturelle, un script Python a été développé pour introduire aléatoirement des substitutions, des insertions, et des délétions dans la séquence de référence du poliovirus. Un taux de mutation de 0,3 % a été appliqué, représentant la variabilité génétique typique rencontrée dans des populations virales. Des taux d'insertion et de délétion de 0,01 % ont également été introduits, alignés sur les observations courantes dans les études de virologie moléculaire.

```
(tfenv) yassin@yassinux:~/tfe/02_mutated_sequence$ python3 difference.py
Différences detectées: 20 différences trouvées.
Position 22: A -> C
Position 1198: T -> G
Position 1297: A -> T
Position 2069: G -> C
Position 2678: G -> C
Position 3038: T -> C
Position 3573: A -> C
Position 3626: C -> T
Position 3708: A -> T
Position 4038: C -> G
Position 4179: A -> G
Position 4346: A -> G
Position 4348: T -> G
Position 4385: A -> C
Position 4471: C -> A
Position 4622: T -> C
Position 5085: G -> A
Position 5233: T -> A
Position 5580: T -> A
Position 6776: T -> C
```

Figure 8: Capture d'écran d'un script python qui affiche les variations entre la séquence originale et mutée

Le script python a créé au total 20 changement de bases. Cette nouvelle séquence mutée sera donc utilisée pour la suite de l'expérience.

3.1.4. Simulation des reads de séquençage

Des reads de séquençage Illumina ont été simulés à partir de la séquence mutée en utilisant l'outil ART, un simulateur de séquençage Illumina. Pour cette simulation, le modèle de séquenceur HiSeq 2500 (HS25) a été sélectionné avec des paramètres spécifiques : une longueur de lecture de 150 nucléotides, une couverture de 5000x, une taille moyenne d'insertion de 200 bases, et une déviation standard de 10 bases. Ces paramètres ont été choisis pour imiter les caractéristiques typiques des données de séquençage Illumina à haute couverture, produisant ainsi un nombre de reads conséquents avec des qualités variables plus proches de la réalité.

```
art_illumina -ss HS25 -i
/home/yassin/tfe/polio/02_mutated_sequence/mutated_sequence.fasta -p -l 150 -f
5000 -m 200 -s 10 -o /home/yassin/tfe/polio/03_reads/mutated_reads
```

Tableau 4: Tableau récapitulatif sur les caractéristiques des fichiers FASTQ

Nom du fichier	Taille du fichier (Mo)	Nombre de reads	Pourcentage GC	Taille moyenne des reads
mutated_reads 1.fq	38.62	122500	46	150.0
mutated_reads 2.fq	38.62	122500	46	150.0

3.1.5. Nettoyage des reads

Après l'analyse FastQC des reads simulés à partir de la séquence mutée, les résultats ont montré une qualité globale modérée mais acceptable pour les objectifs de cette étude. Cependant, un problème notable identifié est le niveau de duplication des séquences. Comme le montre le graphique de l'analyse de duplication, une proportion significative des reads est dupliquée, avec seulement 18,06 % des séquences restant uniques après déduplication. Ce niveau élevé de duplication est récurrent lors d'analyses de séquençage viral à haute couverture, où la redondance des séquences est souvent observée en raison de la taille réduite des génomes viraux et de la nécessité d'une couverture approfondie pour une détection précise des variants. Malgré ce problème de duplication, il a été décidé de ne pas effectuer de trimming des reads. L'objectif est de tester la robustesse des logiciels de variant calling dans des conditions réalistes, incluant les défis typiques rencontrés dans les analyses de séquençage réelles, notamment avec des données à haute couverture et des niveaux de duplication élevés.

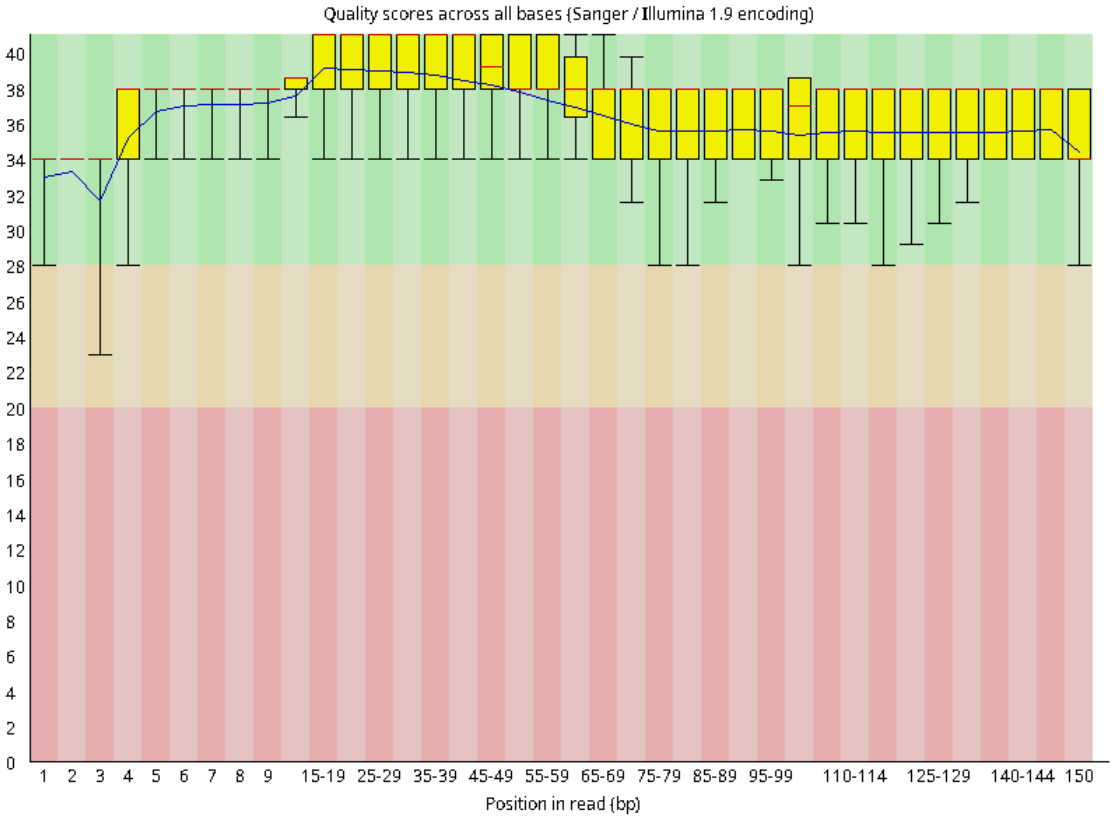


Figure 9: Graphique de qualité FASTQC de la séquence 1

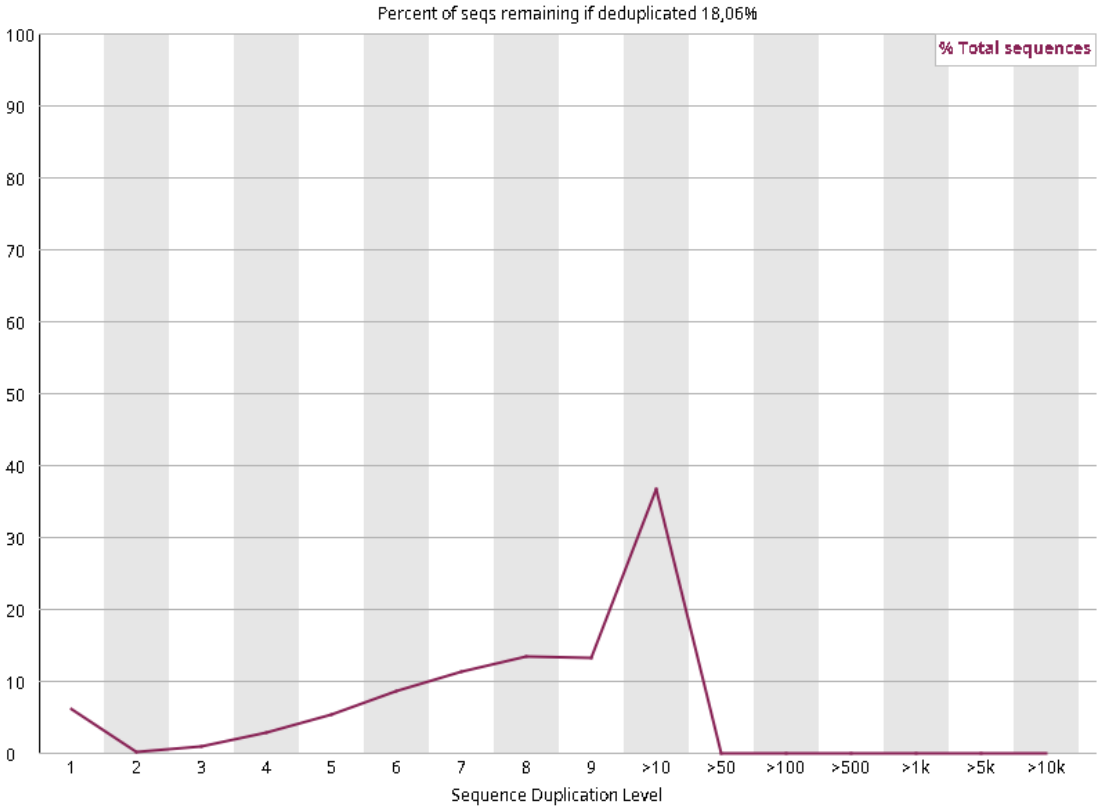


Figure 10: Graphique FASTQC du pourcentage de déduplication de la séquence 1

3.1.6. Alignement des reads

Une fois nettoyées, les lectures ont été alignées sur la séquence de référence à l'aide de BWA (Burrows-Wheeler Aligner), un outil efficace pour aligner les séquences de nucléotides contre une large séquence de référence. La commande suivante a été exécutée pour indexer la séquence de référence, préparant le terrain pour un alignement rapide :

```
bwa index /home/yassin/tfe/polio/01_reference/V01149.1.fasta
```

L'alignement proprement dit a été réalisé avec la commande bwa mem, qui est optimisée pour les lectures de haute qualité et de longueur modérée :

```
bwa mem /home/yassin/tfe/polio/01_reference/V01149.1.fasta
/home/yassin/tfe/polio/03_reads/mutated_reads1.fq
/home/yassin/tfe/polio/03_reads/mutated_reads2.fq >
/home/yassin/tfe/polio/05_SAM_files/mutated_reads.sam
```

Cette étape produit un fichier SAM contenant les lectures alignées, essentiel pour les analyses de variant calling subséquentes.

3.1.7. Conversion et tri des fichiers SAM

Le fichier SAM généré nécessite plusieurs étapes de traitement pour optimiser les performances des analyses ultérieures. Ces étapes ont été réalisées à l'aide de Samtools, un ensemble d'utilitaires pour manipuler les alignements au format SAM/BAM.

```
samtools view -S -b /home/yassin/tfe/polio/05_SAM_files/mutated_reads.sam >
/home/yassin/tfe/polio/06_BAM_files/mutated_reads.bam
```

Le fichier BAM a ensuite été trié pour préparer l'indexation, une étape cruciale pour les recherches et l'accès efficace aux données :

```
samtools sort /home/yassin/tfe/polio/06_BAM_files/mutated_reads.bam -o
/home/yassin/tfe/polio/06_BAM_files/mutated_reads.sorted.bam
```

Ensuite, le fichier BAM trié a été indexé pour permettre un accès rapide aux lectures lors des étapes d'analyse des variants :

```
samtools index /home/yassin/tfe/polio/06_BAM_files/mutated_reads.s.bam
```

Ces étapes d'alignement et de traitement des fichiers SAM/BAM sont cruciales pour assurer la qualité et l'efficacité des analyses de variant calling. Elles garantissent que les données sont préparées et optimisées pour des analyses ultérieures, fournissant une base solide pour des résultats précis et reproductibles.

Après cela, une vérification du taux d’alignement a été nécessaire grâce à cette commande :

```
samtools flagstat /home/yassin/tfe/polio/06_BAM_files/mutated_reads.bam
```

```
yassin@yassinux:~/tfe/polio/08_consensus_sequence$ samtools flagstat /home/yassin/tfe/polio
/06_BAM_files/mutated_reads.bam
245000 + 0 in total (QC-passed reads + QC-failed reads)
245000 + 0 primary
0 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
245000 + 0 mapped (100.00% : N/A)
245000 + 0 primary mapped (100.00% : N/A)
245000 + 0 paired in sequencing
122500 + 0 read1
122500 + 0 read2
244998 + 0 properly paired (100.00% : N/A)
245000 + 0 with itself and mate mapped
0 + 0 singletons (0.00% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

Figure 11: Capture d'écran de la commande "flagstat"

Enfin, les duplicats ont été supprimés et le nouveau fichier BAM réindéxer.

java -jar /home/yassin/bin/picard.jar MarkDuplicates
I=/home/yassin/tfe/polio/06_BAM_files/mutated_reads.s.bam
O=/home/yassin/tfe/polio/06_BAM_files/mutated_reads.dedup.bam
M=/home/yassin/tfe/polio/06_BAM_files/mutated_reads.metrics.txt
REMOVE_DUPLICATES=true

Après la suppression des duplicats avec Picard, les résultats montrent une réduction du nombre total de reads, passant de 245,000 à 195,828. Cela signifie que 49,172 reads ont été identifiés comme duplicats et supprimés, soit environ 20% des reads initiaux. Étant donné que les données utilisées pour cette analyse sont simulées, ce taux de duplication relativement faible est attendu, car les séquences générées par simulation sont souvent plus uniformes et contiennent moins d'artefacts que les données expérimentales réelles. Tous les reads restants sont alignés (100% mappés) et correctement appariés (100% properly paired), ce qui indique un alignement de haute qualité et reflète la précision des simulations.

3.1.8. Variant calling

Dans cette étude, nous avons sélectionné trois outils de variant calling, GATK, FreeBayes et BCFtools, car ils sont considérés comme les plus adaptés pour les données génomiques étudiées, selon les résultats de la littérature. GATK est reconnu pour sa précision dans la détection des SNPs et indels, FreeBayes pour son efficacité dans l'analyse de populations complexes, et BCFtools pour ses capacités de manipulation et d'analyse de données génomiques. À titre comparatif, nous avons également inclus iVar, un outil moins souvent recommandé pour le séquençage de génomes complets, afin d'évaluer sa performance et de vérifier si ses résultats sont cohérents avec les attentes établies par les recherches antérieures dans un contexte moins optimisé pour cet outil.

3.1.8.1. lofreq

lofreq call --min-bq 6 --min-cov 3 -f /home/yassin/tfe/polio/01_reference/V01149.1.fasta -o /home/yassin/tfe/polio/07_variant_calling/03_lofreq/var.vcf /home/yassin/tfe/polio/06_BAM_files/mutated_reads.RG.dedup.recal.bam

3.1.8.2. GATK

Pour réaliser le variant calling avec GATK, un ensemble de commandes spécifiques a été exécuté pour maximiser la précision de la détection des variants à partir des données de séquençage simulées. GATK est largement reconnu pour sa robustesse et sa capacité à traiter des données de haute qualité, ce qui en fait un choix privilégié pour les études génomiques complexes. Cependant, son utilisation requiert une préparation minutieuse des fichiers d'entrée afin d'éviter des erreurs pouvant compromettre les résultats.

La commande principale utilisée pour cette analyse est `HaplotypeCaller`, un outil de GATK conçu pour identifier les SNPs et les indels en comparant les lectures alignées à une séquence de référence. Pour le génome du poliovirus de sérotype 1 (V01149.1), les paramètres suivants ont été utilisés :

gatk --java-options "-Xmx12g -XX:ParallelGCThreads=12" HaplotypeCaller \

-R /home/yassin/tfe/polio/01_reference/V01149.1.fasta \

-I /home/yassin/tfe/polio/06_BAM_files/mutated_reads.RG.dedup.bam \

-O /home/yassin/tfe/polio/07_variant_calling/04_gatk/var.vcf \

--ploidy 1 \

-ERC GVCF \

--native-pair-hmm-threads 12

Dans cette commande, `-R` spécifie le fichier de référence FASTA, qui a été indexé au préalable, et `-I` indique le fichier BAM d'entrée trié et nettoyé des duplicats. Le paramètre `--ploidy 1` est utilisé pour indiquer que l'échantillon est haploïde, une condition nécessaire pour les analyses virales.

Au cours de la mise en place de ce pipeline, deux problèmes majeurs ont été rencontrés. Le premier problème concernait l'absence des groupes de lecture (Read Groups) dans le fichier BAM, qui sont indispensables pour GATK car ils contiennent des métadonnées cruciales telles que l'identifiant de l'échantillon et les détails de la plateforme de séquençage. Pour résoudre ce problème, le fichier BAM a été modifié à l'aide de l'outil Picard pour ajouter les groupes de lecture requis. La commande utilisée était :

java -jar /home/yassin/bin/picard.jar AddOrReplaceReadGroups \

I=/home/yassin/tfe/polio/06_BAM_files/mutated_reads.dedup.bam \

O=/home/yassin/tfe/polio/06_BAM_files/mutated_reads.RG.dedup.bam \

RGID=1 RGLB=lib1 RGPL=illumina RGPU=unit1 RGSM=sample1

Deuxièmement, une erreur due à l'absence d'un fichier de dictionnaire de séquence a été rencontrée. GATK exige un fichier de dictionnaire (`.dict`) pour le génome de référence afin de fonctionner correctement. Ce fichier est essentiel pour l'interprétation correcte de la séquence de référence et pour effectuer les appels de variants. Pour générer ce fichier de dictionnaire, la commande suivante de GATK a été utilisée :

gatk CreateSequenceDictionary \

-R /home/yassin/tfe/polio/01_reference/V01149.1.fasta \

-O /home/yassin/tfe/polio/01_reference/V01149.1.dict

3.1.8.3. freebayes

Une commande est utilisée pour identifier les variants en comparant les lectures alignées à une séquence de référence. FreeBayes est bien adapté aux analyses de populations complexes grâce à son approche probabiliste qui prend en compte la qualité des bases, la couverture et la fréquence des variants.

freebayes -f /home/yassin/tfe/polio/01_reference/V01149.1.fasta \

--ploidy 1 --min-base-quality 20 --min-coverage 10 --min-alternate-fraction 0.2 \

/home/yassin/tfe/polio/06_BAM_files/mutated_reads.dedup.bam > \

/home/yassin/tfe/polio/07_variant_calling/01_freebayes/var.vcf

Dans cette commande, l'option -f spécifie la séquence de référence au format FASTA. FreeBayes est configuré avec --ploidy 1 pour indiquer un génome haploïde, ce qui est pertinent pour l'analyse des virus comme le poliovirus. L'option --min-base-quality 20 filtre les bases de qualité inférieure à 20, améliorant ainsi la précision des appels de variants. --min-coverage 10 exige une couverture minimale de 10 reads, garantissant que les variants sont détectés dans des régions suffisamment couvertes, réduisant ainsi les faux positifs.

L'option --min-alternate-fraction 0.2 permet de détecter des variants présents dans au moins 20% des reads à une position donnée, ce qui est crucial pour identifier des variants de faible fréquence dans des populations virales hétérogènes. Les résultats du variant calling sont ensuite enregistrés dans un fichier VCF, prêt pour des analyses comparatives avec la séquence mutée.

3.1.8.4. iVar

Pour réaliser le variant calling avec iVar, la commande utilise un pipeline qui combine samtools mpileup et iVar variants. Cette approche est particulièrement adaptée pour analyser des données de séquençage viral, comme celles du poliovirus, où l'identification de variants à faible fréquence est cruciale.

```
samtools mpileup -aa -A -d 0 -B -Q 0 -f /home/yassin/tfe/polio/01_reference/V01149.1.fasta \
/home/yassin/tfe/polio/06_BAM_files/mutated_reads.dedup.bam | \
ivar variants -p /home/yassin/tfe/polio/07_variant_calling/02_iVar/var -q 20 -t 0.2 -m 10
```

La commande commence par samtools mpileup, qui génère un fichier d'entrée pour iVar. Les options -aa et -A sont utilisées pour inclure toutes les positions, même celles sans alignements ou celles avec un alignement ambigu, garantissant une analyse exhaustive des variants potentiels. -d 0 désactive la limite par défaut de profondeur de couverture, permettant de traiter des régions avec une couverture très élevée, ce qui est souvent nécessaire pour les séquences virales. L'option -B désactive la correction de biais d'alignement de base, et -Q 0 inclut toutes les bases indépendamment de leur score de qualité. Ces paramètres permettent une analyse complète, y compris dans des zones où la qualité pourrait être variable.

Ensuite, iVar variants est utilisé pour identifier les variants. Le préfixe de sortie est spécifié par -p, ce qui permet d'organiser les résultats de manière structurée. L'option -q 20 impose un seuil de qualité minimale de 20 pour qu'une base soit considérée dans le processus d'appel de variants, réduisant le risque d'inclure des erreurs de séquençage. Le paramètre -t 0.2 fixe un seuil de fréquence minimale de 20% pour appeler un variant, assurant que seuls les variants présents à des fréquences significatives sont rapportés, ce qui est crucial pour détecter des mutations pertinentes dans un contexte viral. Enfin, -m 10 assure que seules les positions avec une couverture minimale de 10 reads sont prises en compte, augmentant ainsi la fiabilité des variants détectés.

L'utilisation de iVar dans ce contexte est particulièrement pertinente pour des études sur les virus, car il est optimisé pour détecter des variants de faible fréquence et gérer des données de séquençage amplicon.

3.1.9. Génération de la séquence consensus

La génération des séquences consensus est une étape cruciale pour évaluer la précision des différents outils de variant calling en reconstituant la séquence génomique à partir des données de séquençage après l'identification des variants. Cette étape permet de comparer directement les séquences reconstruites avec la séquence de référence modifiée, afin de déterminer la fidélité des logiciels utilisés dans l'étude.

Pour chaque outil de variant calling testé (GATK, FreeBayes, lofreq, et iVar), la séquence consensus a été générée à partir des fichiers VCF (fichier TSV pour iVar) produits. Ces fichiers contiennent toutes les informations sur les variations détectées par rapport à la séquence de référence. Voici les étapes suivies pour générer les séquences consensus pour chaque logiciel.

Voici le cheminement pour un fichier VCF classique de variant calling:

```
bgzip -c /home/yassin/tfe/polio/07_variant_calling/03_lofreq/var.vcf > /home/yassin/tfe/polio/07_variant_calling/03_lofreq/var.vcf.gz
```

```
tabix -p vcf /home/yassin/tfe/polio/07_variant_calling/03_lofreq/var.vcf.gz

bcftools consensus -f /home/yassin/tfe/polio/01_reference/V01149.1.fasta
/home/yassin/tfe/polio/07_variant_calling/03_lofreq/var.vcf.gz >
/home/yassin/tfe/polio/08_consensus_sequence/consensus_lofreq.fasta
```

Et voici le cheminement pour créer une séquence consensus à partir d'un fichier TSV (iVar) car le programme permet de créer la séquence consensus directement à l'aide d'un pipe d'un fichier mpileup dans iVar consensus :

```
samtools mpileup -aa -A -d 0 -Q 0 -f /home/yassin/tfe/polio/01_reference/V01149.1.fasta
/home/yassin/tfe/polio/06_BAM_files/mutated_reads.RG.dedup.recal.bam | ivar consensus -p
/home/yassin/tfe/polio/08_consensus_sequence/consensus_ivar -q 20 -t 0.2 -m 10 -n N
```

Ces étapes de génération de séquence consensus ont permis d'évaluer la performance de chaque outil de variant calling en comparant les séquences reconstruites avec la séquence de référence modifiée. Les séquences consensus obtenues sont ensuite alignées et comparées pour déterminer la précision des appels de variants de chaque logiciel, fournissant ainsi une base pour évaluer leur efficacité dans des conditions de simulation réalistes. Cette approche assure que les résultats des différents outils peuvent être directement comparés sur une base équitable, facilitant ainsi l'interprétation des données.

3.3. Expérience 2

L'objectif de cette étude est d'approfondir l'analyse des variants génétiques du virus Monkeypox (mpox) en utilisant des données séquencées réelles, téléchargées depuis la base de données du NCBI. Cette analyse inclut une étape de variant calling pour identifier les différences par rapport au génome de référence et une étape d'annotation des variants pour évaluer leurs effets potentiels. L'annotation permet de caractériser les variants en termes d'impact biologique et fonctionnel, en identifiant notamment les mutations pouvant influencer la structure et la fonction des protéines codées. Cette approche offre une meilleure compréhension des mécanismes évolutifs du virus et de ses interactions avec l'hôte.

3.3.1. Téléchargement des données

Pour cette analyse, les données de séquençage brut ont été téléchargées depuis l'archive SRA sous l'accès [SRX25776527](https://www.ncbi.nlm.nih.gov/sra/SRX25776527). Ces données contiennent des lectures de séquences provenant d'échantillons biologiques pour le virus Monkeypox.

Le génome de référence utilisé pour l'alignement et l'annotation des variants est [GCF_014621545.1](https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_014621545.1/) du NCBI. Ce génome fournit une séquence complète et annotée du virus Monkeypox, nécessaire pour l'identification et l'interprétation des variants détectés.

3.3.2. Génération des fichiers FASTQ

Les données de séquençage brutes en format SRA ont été converties en fichiers FASTQ pour faciliter le traitement et l'analyse. Cette conversion a été réalisée en utilisant l'outil fasterq-dump avec l'option --split-3 sur l'identifiant d'accès SRA SRR30316267. Cette commande permet de séparer les lectures en paires, nécessaires pour les analyses de variant calling et d'alignement ultérieures.

3.3.3. Nettoyage des données de séquençage

Le nettoyage des données de séquençage a été effectué en deux étapes afin d'améliorer la qualité des lectures avant l'alignement et l'analyse de variants.

3.3.3.1. Trimmomatic

```
TrimmomaticPE -threads 12 -phred33
/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_1.fastq
/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_2.fastq
/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_1_paired_trimmed.fastq
/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_1_unpaired_trimmed.fastq
/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_2_paired_trimmed.fastq
/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_2_unpaired_trimmed.fastq
ILLUMINACLIP:/home/yassin/bin/Trimmomatic-0.39/adapters/TruSeq3-PE.fa:2:30:10
LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
```

TrimmomaticPE : Utilise le mode de lecture paire pour les données de séquençage.

-threads 12 : Utilise 12 threads pour accélérer le processus.

-phred33 : Spécifie le format de notation de qualité Phred+33 pour les fichiers FASTQ.

ILLUMINACLIP : Enlève les séquences d'adaptateurs basées sur le fichier d'adaptateurs fourni. Les paramètres 2:30:10 contrôlent la correspondance de l'adaptateur, la moyenne et la tolérance.

LEADING:3 et TRAILING:3 : Supprime les bases de faible qualité (score de qualité inférieur à 3) au début et à la fin des lectures.

SLIDINGWINDOW:4:15 : Applique une fenêtre glissante de 4 bases pour couper quand la qualité moyenne descend en dessous de 15.

MINLEN:36 : Élimine toutes les lectures de longueur inférieure à 36 bases.

Les résultats montrent que 94,69% des paires de lectures ont survécu à cette étape, tandis que 4,41% ont été rejetées en raison d'une mauvaise qualité.

3.3.3.2. BBDuk

```
bbduk.sh
in1=/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_1_paired_trimmed.fastq \
in2=/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_2_paired_trimmed.fastq \
out1=/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_1_clean.fastq \
out2=/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_2_clean.fastq \
ref=/home/yassin/bin/Trimmomatic-0.39/adapters/TruSeq3-PE.fa \
ktrim=r k=23 mink=11 hdist=1 \
qtrim=rl trimq=10 maq=10
```

in1 et in2 : Spécifient les fichiers d'entrée appariés après le traitement avec Trimmomatic.

out1 et out2 : Spécifient les fichiers de sortie après nettoyage avec BBDuk.

ref : Utilise le fichier des adaptateurs TruSeq3 pour identifier et supprimer les séquences d'adaptateurs.

ktrim=r : Coupe à la droite des adaptateurs détectés.

k=23 : Définit la taille des k-mers à 23 pour la détection d'adaptateurs.

mink=11 : Définit la taille minimale des k-mers pour la détection d'adaptateurs.

hdist=1 : Autorise un mismatch dans les k-mers pour la détection d'adaptateurs.

qtrim=rl : Coupe des deux côtés des lectures basées sur la qualité.

trimq=10 : Supprime les bases de qualité inférieure à 10.

maq=10 : Assure que toutes les lectures ont une qualité moyenne d'au moins 10.

Les résultats de BBDuk montrent que l'opération de nettoyage des séquences de lecture a été réalisée avec succès, garantissant une qualité optimale des données pour les analyses ultérieures. Le nombre total de lectures d'entrée était de 19,704,148, avec un total de 2,414,917,627 bases. Après le nettoyage, 164,358 lectures (0,83%) ont été affectées par la coupe des bases de faible qualité (QTrimmed), représentant 735,277 bases (0,03%). De plus, la coupe des adaptateurs (KTrimmed) a touché 4,115 lectures (0,02%), soit 46,480 bases (0,00%). Aucune lecture ou base de faible qualité n'a été complètement supprimée (0,00%). Au total, 781,757 bases ont été supprimées, ce qui représente 0,03% des données initiales. Finalement, le nombre de lectures restantes après le nettoyage est de 19,704,148 (100,00%), avec 2,414,135,870 bases restantes, soit 99,97% des bases initiales. Ces résultats confirment que les séquences de lecture ont été efficacement nettoyées et sont prêtes pour les étapes suivantes de l'analyse.

3.3.4. Alignement des séquences

3.2.3.1. Indexation du génome de référence

```
bwa index /home/yassin/tfe/mpox/00_raw_data/reference.fna
```

L'indexation du génome de référence est une étape pour préparer le fichier de référence (reference.fna) pour l'alignement. La commande bwa index crée plusieurs fichiers d'index qui permettent de rechercher rapidement les séquences lors de l'alignement. Cette étape est nécessaire avant d'exécuter l'alignement proprement dit avec BWA.

3.2.3.2. Alignement avec BWA-MEM

```
bwa mem /home/yassin/tfe/mpox/00_raw_data/reference.fna
/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_1_clean.fastq
/home/yassin/tfe/mpox/01_fastq_data/SRR30316267_2_clean.fastq >
/home/yassin/tfe/mpox/02_sam_files/SRR30316267_aligned.sam
```

L'alignement des séquences de lecture sur le génome de référence est réalisé avec l'outil BWA-MEM, qui est particulièrement efficace pour les lectures de longueur variable et les génomes de grande taille. Les arguments de la commande incluent le fichier de référence et les fichiers FASTQ contenant les séquences nettoyées. Le résultat est un fichier SAM qui contient les alignements bruts des séquences sur le génome de référence, avec des informations sur la qualité de l'alignement et la position des lectures.

3.2.3.3. Conversion du fichier SAM en BAM

```
samtools view -bS /home/yassin/tfe/mpox/02_sam_files/SRR30316267_aligned.sam >
/home/yassin/tfe/mpox/03_bam_files/SRR30316267_aligned.bam
```

Le fichier SAM généré par BWA est converti en format BAM à l'aide de samtools view. Le format BAM est une version binaire compressée du format SAM, ce qui le rend plus efficace pour le stockage et la manipulation des données d'alignement. Les options -bS indiquent que l'entrée est un fichier SAM et que la sortie doit être un fichier BAM. Le fichier BAM résultant est plus compact et rapide à manipuler pour les étapes suivantes de l'analyse.

3.2.3.4. Tri et indexation des fichiers BAM

```
samtools sort /home/yassin/tfe/mpox/03_bam_files/SRR30316267_aligned.bam -o
/home/yassin/tfe/mpox/03_bam_files/SRR30316267_sorted.bam

samtools index /home/yassin/tfe/mpox/03_bam_files/SRR30316267_sorted.bam
```

Une fois le fichier BAM généré, il est trié par position sur le génome de référence en utilisant samtools sort. Cette étape organise les lectures alignées de manière à faciliter leur accès et leur visualisation. Le fichier BAM trié (SRR30316267_sorted.bam) est ensuite indexé avec samtools index, ce qui permet un accès rapide aux données d'alignement. L'indexation est indispensable pour effectuer des recherches rapides et des analyses subséquentes, telles que la détection de variants.

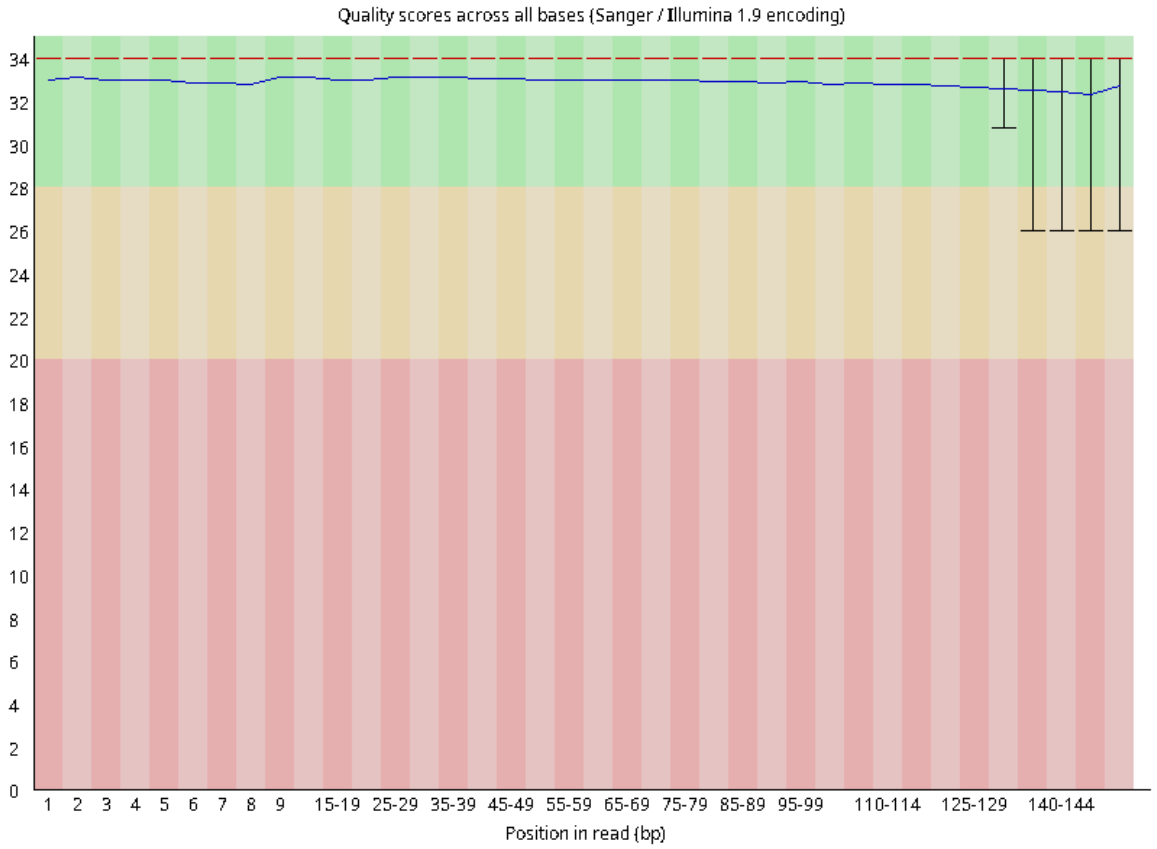


Figure 13: Graphique de qualité sur le fichier S.BAM

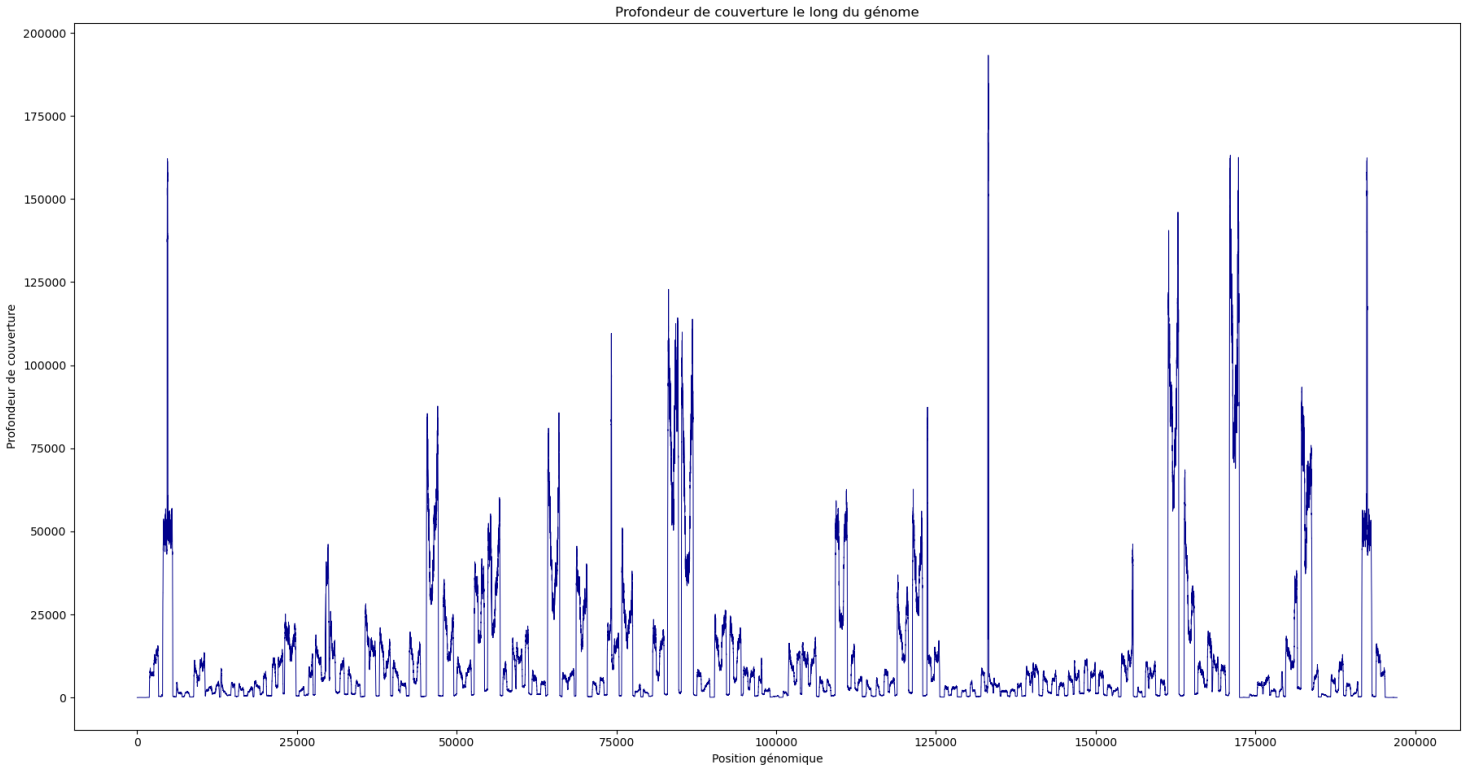


Figure 12: Profondeur de couverture le long du génome

3.2.4. La détection des variants

freebayes -f /home/yassin/tfe/mpox/00_raw_data/reference.fna -p 1 -b /home/yassin/tfe/mpox/03_bam_files/SRR30316267_sorted.bam -v /home/yassin/tfe/mpox/04_variant_calling/SRR30316267_variants.vcf --min-coverage 10 --min-base-quality 20 --min-mapping-quality 20

L'appel de variants avec FreeBayes permet de détecter les variations génétiques par rapport au génome de référence. Les paramètres utilisés dans cette commande sont les suivants :

-f /home/yassin/tfe/mpox/00_raw_data/reference.fna : Fichier FASTA du génome de référence.

-p 1 : Définit la ploidie de l'organisme analysé. Ici, la ploidie est définie à 1.

-b /home/yassin/tfe/mpox/03_bam_files/SRR30316267_sorted.bam : Fichier BAM contenant les alignements des lectures.

-v /home/yassin/tfe/mpox/04_variant_calling/SRR30316267_variants.vcf : Fichier de sortie en format VCF pour stocker les variants détectés.

--min-coverage 10 : Seuls les sites avec une couverture d'au moins 10 lectures sont considérés pour assurer une fiabilité minimale des appels de variants.

--min-base-quality 20 : Exclut les bases dont la qualité est inférieure à 20, garantissant que seules les bases de haute qualité sont utilisées pour l'appel de variants.

--min-mapping-quality 20 : Exclut les lectures dont la qualité d'alignement est inférieure à 20, assurant que seules les lectures correctement alignées sont utilisées.

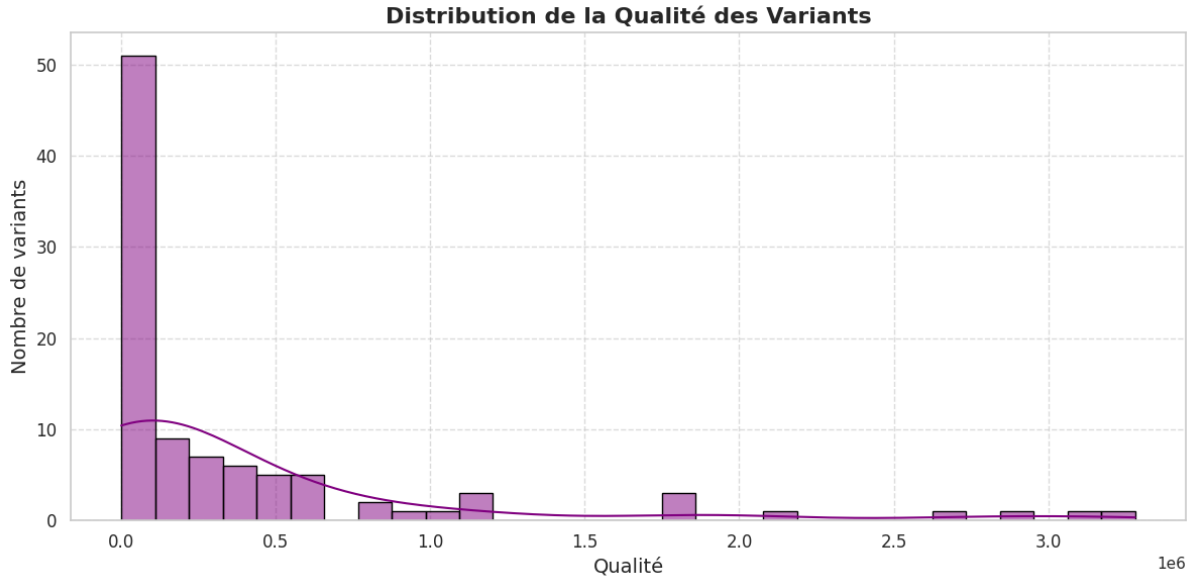


Figure 14: Graphique de la distribution de la qualité des variants

3.2.5. L’annotation des variants

3.2.5.1. Création de la base de données snpEff

```
mkdir -p /home/yassin/bin/snpEff/data/monkeypox/
cp /home/yassin/tfe/mpox/00_raw_data/genomic.gtf
/home/yassin/bin/snpEff/data/monkeypox/genes.gtf
cp /home/yassin/tfe/mpox/00_raw_data/protein.faa
/home/yassin/bin/snpEff/data/monkeypox/protein.fa
cp /home/yassin/tfe/mpox/00_raw_data/cds_from_genomic.fna
/home/yassin/bin/snpEff/data/monkeypox/cds.fa
```

Ces commandes créent un répertoire dédié pour le génome de Monkeypox (monkeypox) dans le répertoire de données de SnpEff et copient les fichiers nécessaires :

genomic.gtf : Fichier GTF contenant les annotations des gènes.

protein.faa : Fichier FASTA des séquences protéiques.

cds_from_genomic.fna : Fichier FASTA des séquences codantes (CDS).

3.2.5.2. Modification du fichier de configuration

Cette ligne configure SnpEff pour reconnaître le génome "monkeypox" comme une entrée valide pour l'annotation. Il est nécessaire de définir correctement cette entrée dans le fichier de configuration pour que SnpEff puisse accéder aux données correspondantes.

```
nano /home/yassin/bin/snpEff/snpEff.config
```

Et ajouter cette ligne:

```
monkeypox.genome : monkeypox
```

Et finalement créer la base de doonées grâce à cette commande :

```
java -Xmx12g -jar /home/yassin/bin/snpEff/snpEff.jar build -gtf22 -v monkeypox
```

Cette commande construit la base de données pour le génome de Monkeypox en utilisant le fichier GTF (-gtf22 spécifie le format GTF version 2.2). Les options utilisées sont :

-Xmx12g : Alloue jusqu'à 12 Go de RAM pour Java afin d'assurer un processus de construction efficace.

-jar : Indique à Java d'exécuter le fichier JAR de SnpEff.

build : Commande SnpEff pour construire la base de données.

-v : Mode verbeux pour afficher des informations supplémentaires durant le processus.

3.2.5.3. Annotation des variants

Cette commande utilise SnpEff pour annoter les variants détectés par rapport au génome de référence "monkeypox" et la nouvelle base de données créée.

```
java -Xmx12g -jar /home/yassin/bin/snpEff/snpEff.jar -v monkeypox
/home/yassin/tfe/mpox/04_variant_calling/SRR30316267_variants.vcf -s
```

/home/yassin/tfe/mpox/04_variant_calling/snpEff_summary.html >
/home/yassin/tfe/mpox/04_variant_calling/SRR30316267_variants_annotated.vcf

-Xmx12g : Alloue jusqu'à 12 Go de RAM pour Java.

-jar : Exécute le fichier JAR de SnpEff.

-v : Mode verbeux pour afficher des détails supplémentaires durant l'annotation.

monkeypox : Nom du génome à utiliser pour l'annotation.

/home/yassin/tfe/mpox/04_variant_calling/SRR30316267_variants.vcf : Fichier VCF d'entrée contenant les variants à annoter.

-s /home/yassin/tfe/mpox/04_variant_calling/snpEff_summary.html : Génère un rapport de résumé de l'annotation dans un fichier HTML.

> : Redirige la sortie annotée vers le fichier SRR30316267_variants_annotated.vcf.

Cette étape permet d'ajouter des informations fonctionnelles sur les variants, telles que les impacts sur les gènes et les protéines, facilitant ainsi l'interprétation biologique des données de variants.

3.2.6. Génération de la séquence consensus

3.2.6.1. Compression du fichier VCF

bgzip /home/yassin/tfe/mpox/04_variant_calling/SRR30316267_variants.vcf

3.2.6.2. Indexation du fichier VCF compressé

tabix -p vcf /home/yassin/tfe/mpox/04_variant_calling/SRR30316267_variants.vcf.gz

3.2.6.3. Génération de la séquence consensus

bcftools consensus -f /home/yassin/tfe/mpox/00_raw_data/reference.fna
/home/yassin/tfe/mpox/04_variant_calling/SRR30316267_variants.vcf.gz >
/home/yassin/tfe/mpox/05_consensus_sequence/SRR30316267_consensus.fasta

bcftools consensus : Cette sous-commande génère une séquence consensus, qui intègre les variants identifiés dans les séquences de lecture par rapport à un génome de référence.

-f /home/yassin/tfe/mpox/00_raw_data/reference.fna : Spécifie le fichier FASTA du génome de référence à utiliser pour la génération du consensus.

/home/yassin/tfe/mpox/04_variant_calling/SRR30316267_variants.vcf.gz : Fichier VCF compressé et indexé contenant les variants identifiés.

> : Redirige la sortie vers le fichier SRR30316267_consensus.fasta qui contiendra la séquence consensus générée.

4. Résultats

4.1. Expérience 1

L'analyse des variants a été réalisée en utilisant quatre outils de variant calling : GATK, FreeBayes, LoFreq, et iVar. Les résultats des fichiers de séquence consensus générés à partir des fichiers VCF montrent que GATK et FreeBayes ont chacun détecté 20 variants. iVar a détecté 19 variants, tandis que LoFreq n'a détecté aucun variant.

Pour évaluer la précision des appels de variants par chaque programme, une heatmap a été générée pour comparer les séquences consensus obtenues avec la séquence mutée de référence. Les résultats de cette analyse révèlent que les séquences consensus produites par GATK et FreeBayes sont identiques à la séquence mutée, suggérant une détection complète et précise des mutations introduites. La séquence obtenue avec iVar montre une correspondance avec 19 variants, indiquant qu'une mutation n'a pas été capturée, ce qui pourrait être dû à des différences dans les algorithmes ou les seuils de qualité utilisés par iVar. LoFreq, en revanche, n'a pas détecté de variants, produisant une séquence consensus identique à la séquence de référence non mutée. Ces résultats mettent en évidence des différences significatives dans la sensibilité et la spécificité des outils de variant calling, avec GATK et FreeBayes montrant une performance supérieure dans ce contexte spécifique, tandis que LoFreq n'a pas réussi à détecter les mutations attendues.

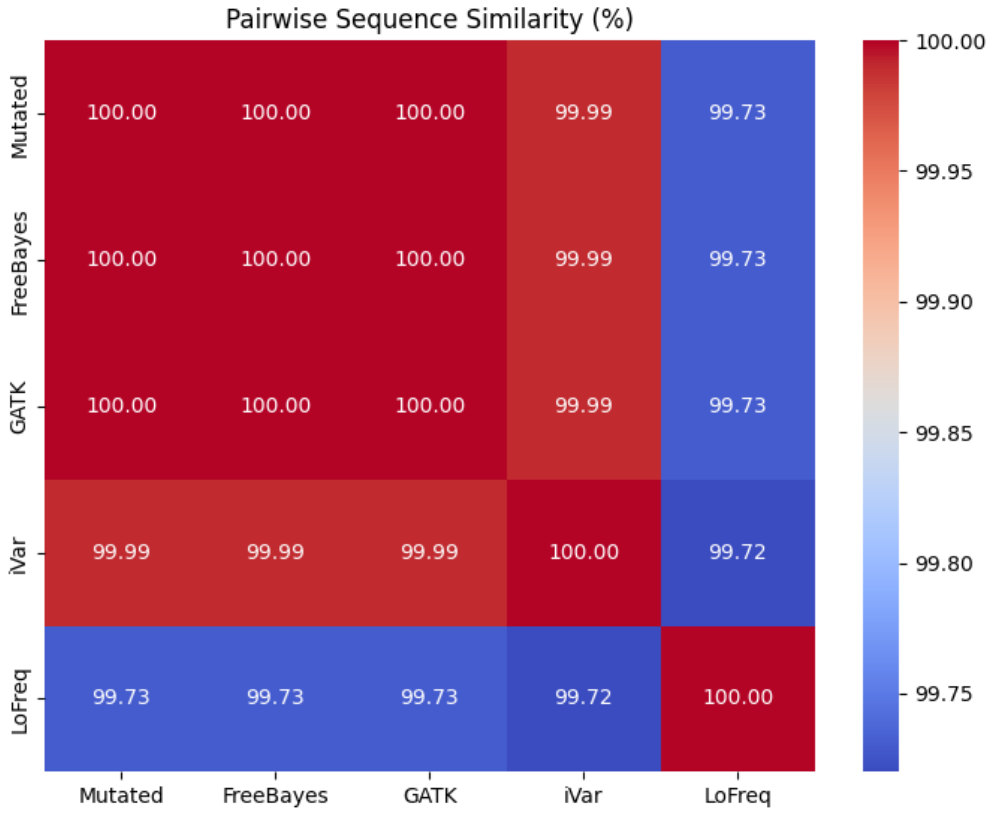


Figure 15: Heat matrix entre les séquences consesnus et la séquence mutée

4.2. Expérience 2

L'analyse des variants du génome du virus monkeypox (MPXV) a permis d'identifier plusieurs mutations significatives à travers le génome viral. Les résultats sont basés sur un alignement précis des séquences de lecture nettoyées contre le génome de référence, suivi d'un appel de variants et d'une annotation approfondie pour comprendre les impacts fonctionnels potentiels de ces mutations.

4.2.1. Identification et classification des variants

Le fichier VCF résultant de l'appel de variants a révélé un total de 98 variants, répartis en 88 SNPs, 4 MNPs, et 6 indels. Une analyse plus détaillée a montré que ces variants incluent principalement des substitutions, avec une prédominance de transitions (80) par rapport aux transversions (7), ce qui est typique dans les séquences virales en raison de la nature des mécanismes de réparation des erreurs de réplication. Le ratio transitions/transversions (Ts/Tv) calculé est de 11,43, indiquant un biais typique des transitions dans l'évolution des génomes viraux.

Summary

Genome	monkeypox
Date	2024-09-01 15:57
SnpEff version	`SnpEff 5.2c (build 2024-04-09 12:24), by Pablo Cingolani`
Command line arguments	`SnpEff monkeypox /home/yassin/tfe/mpox/04_variant_calling/SRR30316267_variants.vcf -s /home/yassin/tfe/mpox/04_variant_calling/snpEff_summary.html`
Warnings	0
Errors	0
Number of lines (input file)	98
Number of variants (before filter)	98
Number of non-variants (i.e. reference equals alternative)	0
Number of variants processed (i.e. after filter and non-variants)	98
Number of known variants (i.e. non-empty ID)	0 (0%)
Number of multi-allelic VCF entries (i.e. more than two alleles)	0
Number of annotations	1,009
Genome total length	197,209
Genome effective length	197,209
Variant rate	1 variant every 2,012 bases

Figure 16: Capture d'écran du summary de snpEff

4.2.2. Distribution des variants par position génomique

L'analyse de la répartition des variants par position génomique a mis en évidence des régions spécifiques du génome présentant une densité plus élevée de mutations. Les régions codantes, en particulier celles codant pour des protéines impliquées dans la réplication virale et la modulation de la réponse immunitaire de l'hôte, ont montré une concentration accrue de variants, suggérant une possible pression sélective. Par exemple, le gène NBT03_gp140 a été identifié avec un variant à impact élevé, ce qui pourrait affecter la protéine C4L/C10L-like, impliquée dans les interactions hôte-pathogène.

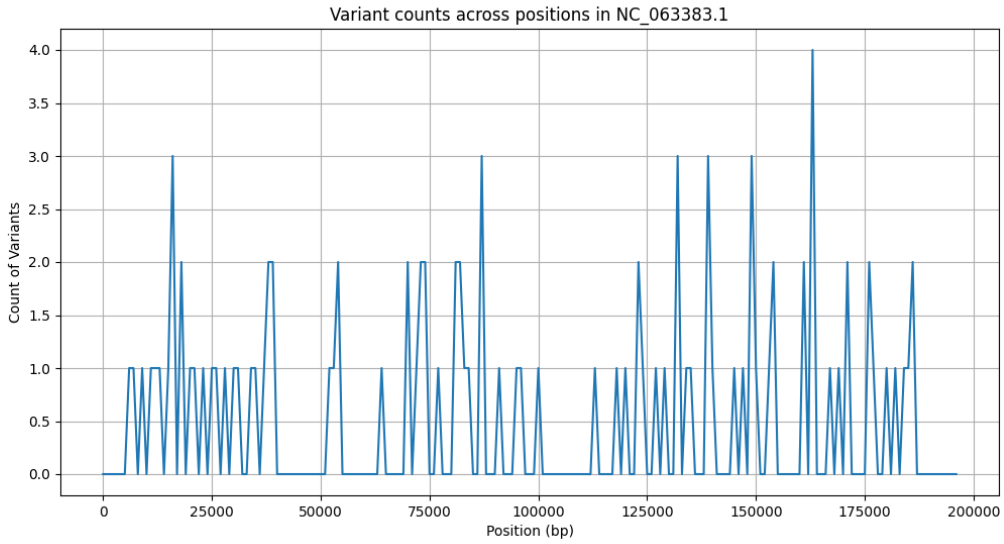


Figure 17: Graphique du nombre de variants en fonction de la position dans le génome

4.2.3. Effets des variants sur les protéines codées

L'annotation fonctionnelle des variants à l'aide de SnpEff a révélé plusieurs impacts potentiels sur les protéines codées par le génome viral. Un variant à impact élevé a été détecté dans le gène NBT03_gp140, tandis que plusieurs variants à impact modéré ont été identifiés dans des gènes codant pour des protéines essentielles du cycle de vie viral. Ces mutations modérées sont principalement des substitutions non synonymes (missense), qui peuvent altérer la fonction des protéines codées. Les variantes synonymes ont également été observées, mais elles n'ont pas d'impact direct sur les séquences protéiques.

4.2.4. Analyse de la couverture et de la qualité des données

L'évaluation de la couverture de séquençage a montré que la majorité des régions du génome avaient une profondeur de lecture suffisante, garantissant une détection robuste des variants. La qualité moyenne des lectures était élevée, avec une majorité des bases ayant un score de qualité supérieur à 30, ce qui minimise le risque d'erreurs lors de l'appel de variants. Toutefois, quelques régions de faible couverture ont été notées, ce qui pourrait limiter la détection complète des mutations dans ces segments.

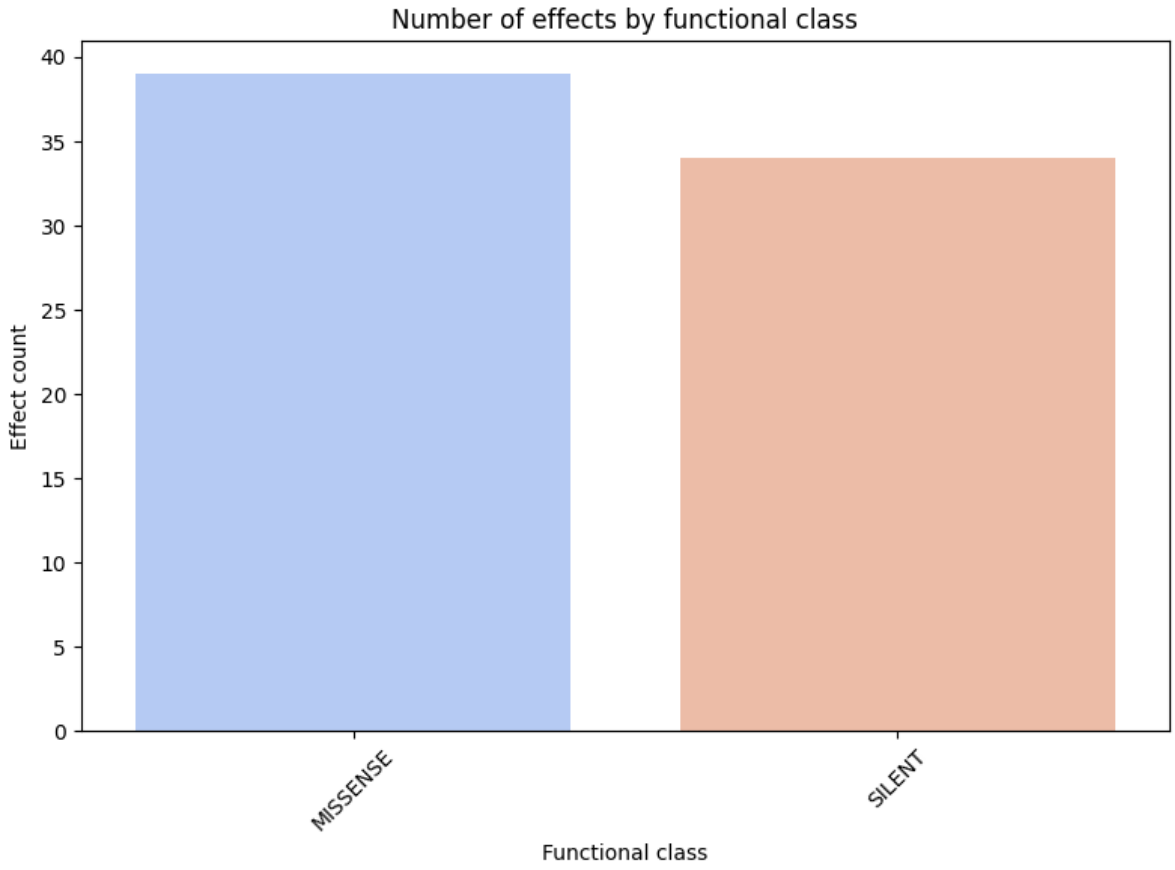


Figure 18: Graphique du nombre d'effets par classe fonctionelle

4.2.5. Annotation et impact fonctionnel des variants

L'annotation par SnpEff a permis de classifier les variants selon leur impact potentiel sur la fonction génétique et leur position relative sur le génome. En plus des impacts élevés et modérés mentionnés précédemment, une large proportion de variants a été classée comme ayant un impact faible ou modificateur, suggérant des effets moins significatifs sur les fonctions protéiques ou régulatrices. Les variantes non codantes, telles que les variants en amont et en aval des gènes, ont également été identifiées, mais leur rôle biologique reste à explorer.

Number of effects by impact

Type (alphabetical order)	Count	Percent
HIGH	1	0.099%
LOW	34	3.37%
MODERATE	44	4.361%
MODIFIER	930	92.17%

Figure 19: Capture d'écran de snpEff sur le nombre d'effets et leur impacts

Ces résultats offrent une vue d'ensemble des mutations présentes dans le génome du virus monkeypox et fournissent des informations pour comprendre les mécanismes évolutifs du virus ainsi que ses interactions potentielles avec l'hôte.

5. Discussions

5.1. Expérience 1

L'évaluation des logiciels de variant calling, GATK, FreeBayes, iVar, et LoFreq sur la séquence du poliovirus de type 1 a révélé des différences notables dans leur performance. GATK et FreeBayes ont détecté tous les 20 SNPs introduits, confirmant leur fiabilité pour identifier des variants dans des génomes courts avec des mutations claires et bien représentées. Ces outils sont particulièrement adaptés pour des analyses de séquençage de haute qualité où la précision est essentielle.

iVar a identifié 19 variants, manquant un SNP. Bien que légèrement moins sensible que GATK et FreeBayes, iVar reste efficace dans des contextes de séquençage de type amplicon, où il est crucial de détecter des variants à faible fréquence. Son échec à détecter un SNP peut être attribué à ses seuils rigoureux de qualité et de fréquence, conçus pour minimiser les faux positifs.

LoFreq, cependant, n'a détecté aucun variant. Ce résultat est surprenant compte tenu de sa conception pour identifier des variants à très faible fréquence dans des données de séquençage à très haute couverture. LoFreq est généralement plus efficace pour des taux de mutation bien inférieurs (moins de 0,1 %) et des génomes plus complexes ou plus grands que ceux utilisés dans cette étude. Le taux de mutation relativement élevé (0,3 %) et la petite taille du génome du poliovirus peuvent expliquer pourquoi LoFreq n'a pas détecté de variants.

En conclusion, le choix du logiciel de variant calling doit être adapté à la taille du génome, à la fréquence des mutations et à la profondeur de couverture des données de séquençage. GATK et FreeBayes sont recommandés pour les analyses de génomes viraux courts et bien couverts, tandis que LoFreq pourrait être plus pertinent pour les analyses de variants à très faible fréquence dans des contextes de séquençage à très haute couverture.

5.2. Expérience 2

Les résultats de cette étude apportent un éclairage nouveau sur les mécanismes évolutifs et d’adaptation du virus monkeypox. Les mutations identifiées, notamment celles ayant un impact élevé et modéré, sont susceptibles de jouer un rôle clé dans la pathogenèse et la propagation du virus. En particulier, les mutations dans des gènes tels que NBT03_gp140, qui sont impliqués dans la modulation de la réponse immunitaire de l'hôte, suggèrent que le virus pourrait avoir développé des stratégies pour échapper aux défenses immunitaires, augmentant ainsi sa capacité à causer des infections sévères.

De plus, les mutations observées dans les gènes impliqués dans la réplication virale, telles que celles présentes dans NBT03_gp008 et NBT03_gp012, pourraient indiquer une adaptation du virus pour améliorer son efficacité de réplication. Ces mutations pourraient permettre au virus d’accroître sa virulence en augmentant sa charge virale, sa transmissibilité ou sa capacité à persister dans l’hôte. Cette adaptation pourrait également avoir des implications pour la gestion de la maladie, en influençant la réponse aux traitements antiviraux.

Il est cependant important de souligner que l’impact biologique exact de ces mutations reste incertain sans des études fonctionnelles complémentaires. Les résultats obtenus via des approches bioinformatiques fournissent des indications précieuses sur les mutations qui

pourraient être d'intérêt, mais ces hypothèses nécessitent une validation expérimentale. Par exemple, des expériences in vitro et in vivo pourraient être menées pour déterminer comment ces mutations affectent la fonction des protéines virales et l’interaction avec l’hôte.

Enfin, ces résultats renforcent l'importance de la surveillance génomique continue dans le contexte de la gestion des maladies virales émergentes. Le suivi des variants du virus monkeypox pourrait fournir des informations cruciales pour anticiper les futures évolutions du virus, adapter les stratégies vaccinales et thérapeutiques, et mieux comprendre les déterminants moléculaires de la virulence et de la transmission. Les études futures devraient également intégrer des données cliniques et épidémiologiques pour établir des corrélations plus directes entre les mutations génomiques et les manifestations cliniques de l'infection. Cela permettrait d'élaborer des stratégies plus ciblées pour prévenir et contrôler les épidémies de monkeypox.

6. Conclusions

Ce travail a été essentiel pour acquérir une compréhension globale du processus de variant calling et de l'utilisation efficace des outils d'analyse génomique. L'étude a mis en lumière l'importance de nombreux paramètres à prendre en compte pour obtenir des résultats fiables, tels que la ploïdie, le type d'allèles, la longueur des séquences, la taille du génome, et la couverture de séquençage.

Bien que la plupart des logiciels de variant calling puissent être adaptés à différentes séquences génomiques, leur performance dépend fortement des paramètres choisis et de la configuration précise des analyses. GATK a démontré sa robustesse grâce à ses options avancées, permettant une personnalisation détaillée qui optimise la détection des variants en fonction des caractéristiques spécifiques de chaque séquence étudiée.

DeepVariant s'est également distingué, notamment grâce à ses sorties de fichiers claires, y compris un fichier HTML qui offre une analyse visuelle partielle des résultats de variant calling, facilitant ainsi l'interprétation des données. Ce projet a permis de tester et d'évaluer un large éventail de logiciels à chaque étape du pipeline, depuis la préparation des données jusqu'à la génération des séquences consensus, tout en intégrant des outils d'annotation qui relient les variations génétiques à leur impact biologique potentiel.

En somme, ce travail a non seulement offert une formation pratique sur l'utilisation de divers outils de variant calling, mais a également souligné l'importance de choisir et de configurer soigneusement ces outils en fonction des spécificités de chaque analyse pour maximiser la précision et la pertinence des résultats obtenus. Par ailleurs, ce rapport peut servir de guide utile pour un novice souhaitant se lancer dans le variant calling, en fournissant des exemples concrets et des conseils pratiques sur l'ensemble du processus, de la préparation des données à l'analyse des résultats.

7. Perspectives

Les expériences menées sur le variant calling des virus monkeypox et poliovirus offrent plusieurs perspectives pour les recherches futures :

- Optimisation des outils de variant calling: Tester les performances des outils de variant calling sur des génomes plus longs et avec des taux de mutation variés pourrait permettre une meilleure compréhension de leurs forces et limites, en particulier pour les séquences virales complexes.

- Approfondissement des analyses d'annotation: Utiliser des outils d'annotation avancés pour prédire les impacts fonctionnels des variants peut fournir des informations importantes sur les effets biologiques des mutations, notamment en les associant à des données structurelles de protéines.

- Études comparatives: Comparer les variants détectés avec d'autres souches virales ou espèces pourrait révéler des modèles d'évolution et d'adaptation virale, en termes de virulence et de résistance aux traitements.

- Développement de pipelines automatisés: Créer des pipelines automatisés spécifiques aux séquences virales, incluant le variant calling, l'annotation et la génération de séquences consensus, améliorerait la reproductibilité des analyses génomiques et rendrait ces techniques plus accessibles, surtout pour les chercheurs novices.

Ces perspectives mettent en lumière l'importance d'une amélioration continue des outils et méthodes pour approfondir la compréhension des mutations virales et de leurs impacts biologiques, tout en facilitant leur application dans des contextes de recherche diversifiés.

8. Bibliographie

1. *Ardui, S., Ameur, A., Vermeesch, J. R., & Hestand, M. S. (2018). Single molecule real-time (SMRT) sequencing comes of age: applications and utilities for medical diagnostics. Nucleic Acids Research, 46(5), 2159–2168. https://doi.org/10.1093/nar/gky066*
2. *Brookes, A. J. (1999). The essence of SNPs. Gene, 234(2), 177–186. https://doi.org/10.1016/s0378-1119(99)00219-x*
3. *Koboldt, D. C. (2020). Best practices for variant calling in clinical sequencing. Genome Medicine, 12(1). https://doi.org/10.1186/s13073-020-00791-w*
4. *Mills, R. E., Walter, K., Stewart, C., Handsaker, R. E., Chen, K., Alkan, C., Abyzov, A., Yoon, S. C., Ye, K., Cheetham, R. K., Chinwalla, A., Conrad, D. F., Fu, Y., Grubert, F., Hajirasouliha, I., Hormozdiari, F., Iakoucheva, L. M., Iqbal, Z., Kang, S., . . . Korbel, J. O. (2011). Mapping copy number variation by population-scale genome sequencing. Nature, 470(7332), 59–65. https://doi.org/10.1038/nature09708*
5. *Mirchandani, C. D., Shultz, A. J., Thomas, G. W. C., Smith, S. J., Baylis, M., Arnold, B., Corbett-Detig, R., Enbody, E., & Sackton, T. B. (2023). A fast, reproducible, high-throughput variant calling Workflow for population genomics. Molecular Biology and Evolution, 41(1). https://doi.org/10.1093/molbev/msad270*
6. *Modi, A., Vai, S., Caramelli, D., & Lari, M. (2021a). The Illumina Sequencing Protocol and the NovaSeq 6000 System. Methods in Molecular Biology, 15–42. https://doi.org/10.1007/978-1-0716-1099-2_2*
7. *Modi, A., Vai, S., Caramelli, D., & Lari, M. (2021b). The Illumina Sequencing protocol and the NovaSeq 6000 system. Methods in Molecular Biology, 15–42. https://doi.org/10.1007/978-1-0716-1099-2_2*
8. *Sezerman, O. U., Ulgen, E., Seymen, N., & Durasi, I. M. (2019). Bioinformatics workflows for genomic variant discovery, interpretation and prioritization. In IntechOpen eBooks. https://doi.org/10.5772/intechopen.85524*
9. *Stankiewicz, P., & Lupski, J. R. (2010). Structural Variation in the Human Genome and its Role in Disease. Annual Review of Medicine, 61(1), 437–455. https://doi.org/10.1146/annurev-med-100708-204735*

9. Lexique et abréviations

Ploïdie : Nombre de jeux de chromosomes dans une cellule. Une cellule haploïde a un jeu, une diploïde en a deux.

Qualité Phred : Score mesurant la confiance dans la précision d'une base séquencée. Plus le score est élevé, plus la précision est grande.

SAM : Format texte pour stocker l'alignement des séquences contre un génome de référence.

BAM : Version binaire compressée du format SAM, utilisée pour un stockage plus efficace des alignements.

VCF : Format texte pour enregistrer les variations génétiques détectées par rapport à un génome de référence.

Workflow : Série d'étapes structurées pour effectuer une analyse bioinformatique, de la préparation des données à l'interprétation des résultats.

BAM	Binary Alignment/Map
BWA	Burrows-Wheeler Aligner
CDS	Coding DNA Sequence
CNV	Copy Number Variation
FASTA	Format de texte pour les séquences d'ADN ou d'ARN
FASTQ	Format de fichier texte qui contient les séquences de reads et leurs qualités associées
GATK	Genome Analysis Toolkit
NCBI	National Center for Biotechnology Information
NGS	Next-Generation Sequencing
PCR	Polymerase Chain Reaction
RNA-seq	RNA sequencing
SAM	Sequence Alignment/Map
SNP	Single Nucleotide Polymorphism
SRA	Sequence Read Archive
TSV	Tab-Separated Values
VCF	Variant Call Format

10. Annexes

1. Script mutation de la séquence V01149.1

```
import random

def introduce_mutations(sequence, mutation_rate=0.003, insertion_rate=0.0001, deletion_rate=0.0001):
    sequence = list(sequence)
    bases = ['A', 'T', 'C', 'G']

    num_substitutions = int(len(sequence) * mutation_rate)
    num_insertions = int(len(sequence) * insertion_rate)
    num_deletions = int(len(sequence) * deletion_rate)

    for _ in range(num_substitutions):
        i = random.randint(0, len(sequence) - 1)
        sequence[i] = random.choice(bases)

    for _ in range(num_insertions):
        i = random.randint(0, len(sequence) - 1)
        sequence.insert(i, random.choice(bases))

    for _ in range(num_deletions):
        if len(sequence) > 1:  # Ensure there is something to delete
            i = random.randint(0, len(sequence) - 1)
            sequence.pop(i)

    return ''.join(sequence)

def write_fasta(sequence, output_file, line_length=70):
    with open(output_file, 'w') as f:
        f.write('>mutated_sequence\n')
        for i in range(0, len(sequence), line_length):
            f.write(sequence[i:i+line_length] + '\n')

# Définir les chemins des fichiers
input_file = '/home/yassin/tfe2/01_raw_data/01_reference/V01149.1.fasta'
output_file = '/home/yassin/tfe2/01_raw_data/01_reference/mutated_sequence.fasta'

# Chargement de la séquence de référence
with open(input_file) as f:
    lines = f.readlines()
    reference_sequence = ''.join([line.strip() for line in lines if not line.startswith('>')])

# Introduction des mutations
mutated_sequence = introduce_mutations(reference_sequence)

# Sauvegarde de la séquence mutée avec formatage FASTA
write_fasta(mutated_sequence, output_file)
```

2. Script bash du workflow complet de l'expérience 1

```
#!/bin/bash

# Définir le chemin de base pour l'expérience
BASE_DIR="/home/yassin/tfe/polio"
REF_GENOME="$BASE_DIR/01_reference/V01149.1.fasta"
MUTATED_SEQ="$BASE_DIR/02_mutated_sequence/mutated_sequence.fasta"
READS_DIR="$BASE_DIR/03_reads"
SAM_DIR="$BASE_DIR/05_SAM_files"
BAM_DIR="$BASE_DIR/06_BAM_files"
VARIANT_DIR="$BASE_DIR/07_variant_calling"
CONSENSUS_DIR="$BASE_DIR/08_consensus_sequence"
LOG_DIR="$BASE_DIR/logs"

# Créer les dossiers nécessaires
mkdir -p $READS_DIR $SAM_DIR $BAM_DIR $VARIANT_DIR $CONSENSUS_DIR $LOG_DIR

# Étape 1: Simulation des reads de séquençage Illumina
echo "Simulation des reads de séquençage Illumina..."
art_illumina -ss HS25 -i $MUTATED_SEQ -p -l 150 -f 5000 -m 200 -s 10 -o $READS_DIR/mutated_reads >
$LOG_DIR/art_illumina.log 2>&1

# Étape 2: Indexation du génome de référence
echo "Indexation du génome de référence..."
bwa index $REF_GENOME > $LOG_DIR/bwa_index.log 2>&1

# Étape 3: Alignement des reads
echo "Alignement des reads..."
bwa mem $REF_GENOME $READS_DIR/mutated_reads1.fq $READS_DIR/mutated_reads2.fq > $SAM_DIR/mu-
tated_reads.sam 2> $LOG_DIR/bwa_mem.log

# Étape 4: Conversion du fichier SAM en BAM, tri et indexation
echo "Conversion et tri du fichier BAM..."
samtools view -S -b $SAM_DIR/mutated_reads.sam > $BAM_DIR/mutated_reads.bam 2> $LOG_DIR/sam-
tools_view.log
samtools sort $BAM_DIR/mutated_reads.bam -o $BAM_DIR/mutated_reads.sorted.bam 2> $LOG_DIR/sam-
tools_sort.log
samtools index $BAM_DIR/mutated_reads.sorted.bam > $LOG_DIR/samtools_index.log 2>&1

# Étape 5: Marquage des duplicats avec Picard
echo "Marquage des duplicats..."
java -jar /home/yassin/bin/picard.jar MarkDuplicates \
  I=$BAM_DIR/mutated_reads.sorted.bam \
  O=$BAM_DIR/mutated_reads.dedup.bam \
  M=$BAM_DIR/mutated_reads.metrics.txt \
  REMOVE_DUPLICATES=true > $LOG_DIR/picard_markduplicates.log 2>&1

# Étape 6: Variant calling avec LoFreq
echo "Variant calling avec LoFreq..."
lofreq call --min-bq 6 --min-cov 3 -f $REF_GENOME -o $VARIANT_DIR/03_lofreq/var.vcf $BAM_DIR/mu-
tated_reads.dedup.bam > $LOG_DIR/lofreq_call.log 2>&1

# Étape 7: Variant calling avec GATK HaplotypeCaller
echo "Variant calling avec GATK HaplotypeCaller..."
gatk --java-options "-Xmx12g -XX:ParallelGCThreads=12" HaplotypeCaller \
  -R $REF_GENOME \
  -I $BAM_DIR/mutated_reads.dedup.bam \
```

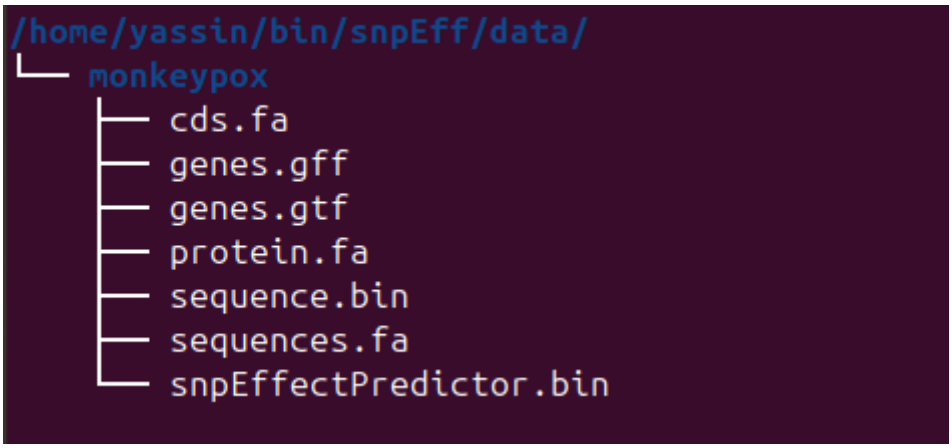
```
  -O $VARIANT_DIR/04_gatk/var.vcf \
  --ploidy 1 \
  -ERC GVCF \
  --native-pair-hmm-threads 12 > $LOG_DIR/gatk_haplotypecaller.log 2>&1

# Étape 8: Création du dictionnaire de séquence pour GATK
echo "Création du dictionnaire de séquence..."
gatk CreateSequenceDictionary \
  -R $REF_GENOME \
  -O $BASE_DIR/01_reference/V01149.1.dict > $LOG_DIR/gatk_createdict.log 2>&1

# Étape 9: Variant calling avec FreeBayes
echo "Variant calling avec FreeBayes..."
freebayes -f $REF_GENOME --ploidy 1 --min-base-quality 20 --min-coverage 10 --min-alternate-fraction 0.2 \
  $BAM_DIR/mutated_reads.dedup.bam > $VARIANT_DIR/01_freebayes/var.vcf 2> $LOG_DIR/freebayes.log

# Étape 10: Variant calling avec iVar
echo "Variant calling avec iVar..."
samtools mpileup -aa -A -d 0 -B -Q 0 -f $REF_GENOME $BAM_DIR/mutated_reads.dedup.bam | \
  ivar variants -p $VARIANT_DIR/02_iVar/var -q 20 -t 0.2 -m 10 > $LOG_DIR/ivar_variants.log 2>&1

# Étape 11: Génération des séquences consensus
echo "Génération des séquences consensus..."
# Pour lofreq, freebayes et GATK
for TOOL in lofreq freebayes gatk; do
  bgzip -c $VARIANT_DIR/03_$TOOL/var.vcf > $VARIANT_DIR/03_$TOOL/var.vcf.gz
  tabix -p vcf $VARIANT_DIR/03_$TOOL/var.vcf.gz
  bcftools consensus -f $REF_GENOME $VARIANT_DIR/03_$TOOL/var.vcf.gz > $CONSENSUS_DIR/consen-
sus_$TOOL.fasta
done

# Pour iVar
samtools mpileup -aa -A -d 0 -Q 0 -f $REF_GENOME $BAM_DIR/mutated_reads.RG.dedup.recal.bam | \
ivar consensus -p $CONSENSUS_DIR/consensus_ivar -q 20 -t 0.2 -m 10 -n N > $LOG_DIR/ivar_consensus.log 2>&1

echo "Pipeline terminé avec succès!"
```

3. Script bash complet du workflow de l’expérience 2

```bash
#!/bin/bash

# Définir les chemins de base pour l'expérience
BASE_DIR="/home/yassin/tfe/mpox"
RAW_DATA_DIR="$BASE_DIR/00_raw_data"
FASTQ_DIR="$BASE_DIR/01_fastq_data"
SAM_DIR="$BASE_DIR/02_sam_files"
BAM_DIR="$BASE_DIR/03_bam_files"
VARIANT_DIR="$BASE_DIR/04_variant_calling"
CONSENSUS_DIR="$BASE_DIR/05_consensus_sequence"
LOG_DIR="$BASE_DIR/logs"
SRA_ID="SRR30316267"
REF_GENOME="$RAW_DATA_DIR/reference.fna"

# Créer les dossiers nécessaires
mkdir -p $FASTQ_DIR $SAM_DIR $BAM_DIR $VARIANT_DIR $CONSENSUS_DIR $LOG_DIR

echo "=== Début du pipeline de traitement pour l'analyse des variants du virus Monkeypox ==="

# Étape 1: Téléchargement et conversion des données SRA en fichiers FASTQ
echo "Téléchargement et conversion des données SRA en fichiers FASTQ..."
fasterq-dump --split-3 $SRA_ID -O $FASTQ_DIR > $LOG_DIR/fasterq-dump.log 2>&1

# Étape 2: Nettoyage des données de séquençage avec Trimmomatic
echo "Nettoyage des données de séquençage avec Trimmomatic..."
TrimmomaticPE -threads 12 -phred33 \
  $FASTQ_DIR/${SRA_ID}_1.fastq $FASTQ_DIR/${SRA_ID}_2.fastq \
  $FASTQ_DIR/${SRA_ID}_1_paired_trimmed.fastq $FASTQ_DIR/${SRA_ID}_1_unpaired_trimmed.fastq \
  $FASTQ_DIR/${SRA_ID}_2_paired_trimmed.fastq $FASTQ_DIR/${SRA_ID}_2_unpaired_trimmed.fastq \
  ILLUMINACLIP:/home/yassin/bin/Trimmomatic-0.39/adapters/TruSeq3-PE.fa:2:30:10 \
  LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36 > $LOG_DIR/trimmomatic.log 2>&1

# Étape 3: Nettoyage supplémentaire des données avec BBDuk
echo "Nettoyage supplémentaire des données avec BBDuk..."
bbduk.sh in1=$FASTQ_DIR/${SRA_ID}_1_paired_trimmed.fastq \
  in2=$FASTQ_DIR/${SRA_ID}_2_paired_trimmed.fastq \
  out1=$FASTQ_DIR/${SRA_ID}_1_clean.fastq \
  out2=$FASTQ_DIR/${SRA_ID}_2_clean.fastq \
  ref=/home/yassin/bin/Trimmomatic-0.39/adapters/TruSeq3-PE.fa \
  ktrim=r k=23 mink=11 hdist=1 qtrim=rl trimq=10 maq=10 > $LOG_DIR/bbduk.log 2>&1

# Étape 4: Indexation du génome de référence
echo "Indexation du génome de référence..."
bwa index $REF_GENOME > $LOG_DIR/bwa_index.log 2>&1

# Étape 5: Alignement avec BWA-MEM
echo "Alignement des séquences avec BWA-MEM..."
bwa mem $REF_GENOME $FASTQ_DIR/${SRA_ID}_1_clean.fastq $FASTQ_DIR/${SRA_ID}_2_clean.fastq >
$SAM_DIR/${SRA_ID}_aligned.sam 2> $LOG_DIR/bwa_mem.log

# Étape 6: Conversion du fichier SAM en BAM et tri
echo "Conversion et tri du fichier BAM..."
samtools view -bS $SAM_DIR/${SRA_ID}_aligned.sam > $BAM_DIR/${SRA_ID}_aligned.bam 2> $LOG_DIR/sam-
tools_view.log
samtools sort $BAM_DIR/${SRA_ID}_aligned.bam -o $BAM_DIR/${SRA_ID}_sorted.bam 2> $LOG_DIR/sam-
tools_sort.log

# Étape 7: Indexation du fichier BAM trié
```

```
echo "Indexation du fichier BAM trié..."
samtools index $BAM_DIR/${SRA_ID}_sorted.bam > $LOG_DIR/samtools_index.log 2>&1

# Étape 8: Appel de variants avec FreeBayes
echo "Appel de variants avec FreeBayes..."
freebayes -f $REF_GENOME -p 1 -b $BAM_DIR/${SRA_ID}_sorted.bam \
  -v $VARIANT_DIR/${SRA_ID}_variants.vcf \
  --min-coverage 10 --min-base-quality 20 --min-mapping-quality 20 > $LOG_DIR/freebayes.log 2>&1

# Étape 9: Création de la base de données snpEff pour le génome Monkeypox
echo "Création de la base de données snpEff pour Monkeypox..."
mkdir -p /home/yassin/bin/snpEff/data/monkeypox/
cp $RAW_DATA_DIR/genomic.gtf /home/yassin/bin/snpEff/data/monkeypox/genes.gtf
cp $RAW_DATA_DIR/protein.faa /home/yassin/bin/snpEff/data/monkeypox/protein.fa
cp $RAW_DATA_DIR/cds_from_genomic.fna /home/yassin/bin/snpEff/data/monkeypox/cds.fa

echo "Modification du fichier de configuration snpEff..."
echo "monkeypox.genome : monkeypox" >> /home/yassin/bin/snpEff/snpEff.config

echo "Construction de la base de données snpEff..."
java -Xmx12g -jar /home/yassin/bin/snpEff/snpEff.jar build -gtf22 -v monkeypox > $LOG_DIR/snpEff_build.log
2>&1

# Étape 10: Annotation des variants avec snpEff
echo "Annotation des variants avec snpEff..."
java -Xmx12g -jar /home/yassin/bin/snpEff/snpEff.jar -v monkeypox $VARIANT_DIR/${SRA_ID}_variants.vcf \
  -s $VARIANT_DIR/snpEff_summary.html > $VARIANT_DIR/${SRA_ID}_variants_annotated.vcf 2>
$LOG_DIR/snpEff_annotation.log

# Étape 11: Génération de la séquence consensus
echo "Génération de la séquence consensus..."
bgzip $VARIANT_DIR/${SRA_ID}_variants.vcf
tabix -p vcf $VARIANT_DIR/${SRA_ID}_variants.vcf.gz
bcftools consensus -f $REF_GENOME $VARIANT_DIR/${SRA_ID}_variants.vcf.gz > $CONSEN-
SUS_DIR/${SRA_ID}_consensus.fasta

echo "=== Fin du pipeline de traitement ==="
```

4. Structure du dossier /snpEff/data



5. Modification du fichier de configuration de snpEff

```
ftp.ensemblgenomes.org/pub/release-46
_pseudomonas_geniculata_atcc_19374_jcm_13324.retrieval_date : 2020-01-26

_pseudomonas_syringae_pv_tomato_str_dc3000.genome :
_pseudomonas_syringae_pv_tomato_str_dc3000
_pseudomonas_syringae_pv_tomato_str_dc3000.reference :
ftp.ensemblgenomes.org/pub/release-46
_pseudomonas_syringae_pv_tomato_str_dc3000.retrieval_date : 2020-01-26

# Ebola virus
ebola_zaire.genome: Ebola Zaire Virus KJ660346.1

# Ursidibacter maritimus
lekn01.genome: Ursidibacter maritimus
lekn01.reference: https://www.ncbi.nlm.nih.gov/Traces/wgs/LEKN01

monkeypox.genome : monkeypox
monkeypox.reference : gtf
```

6. Script python pour générer la « heat matrix »

```
from Bio import SeqIO
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Function to calculate pairwise similarity percentage
def calculate_similarity(seq1, seq2):
    matches = sum(1 for a, b in zip(seq1, seq2) if a == b)
    similarity = (matches / len(seq1)) * 100
    return round(similarity, 2)

# Load sequences using absolute paths
sequences = {
    "Mutated": str(SeqIO.read("/home/yassin/tfe/polio/02_mutated_sequence/mutated_sequence.fasta",
"fasta").seq),
    "FreeBayes": str(SeqIO.read("/home/yassin/tfe/polio/08_consensus_sequence/consensus_freebayes.fasta",
"fasta").seq),
    "GATK": str(SeqIO.read("/home/yassin/tfe/polio/08_consensus_sequence/consensus_gatk.fasta", "fasta").seq),
    "iVar": str(SeqIO.read("/home/yassin/tfe/polio/08_consensus_sequence/consensus_ivar.fa", "fasta").seq),
    "LoFreq": str(SeqIO.read("/home/yassin/tfe/polio/08_consensus_sequence/consensus_lofreq.fasta",
"fasta").seq),
}

sequence_names = ["Mutated", "FreeBayes", "GATK", "iVar", "LoFreq"]

# Initialize similarity matrix
similarity_matrix = np.zeros((len(sequences), len(sequences)))

# Calculate pairwise similarities in the specified order
for i, name1 in enumerate(sequence_names):
    for j, name2 in enumerate(sequence_names):
        similarity_matrix[i, j] = calculate_similarity(sequences[name1], sequences[name2])

# Plot heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(similarity_matrix, annot=True, fmt=".2f", xticklabels=sequence_names, yticklabels=sequence_names,
cmap="coolwarm")
plt.title("Pairwise Sequence Similarity (%)")
plt.show()
```

7. Capture d’écran du fichier html de snpEff

Summary

Genome	monkeypox
Date	2024-09-01 15:57
SnpEff version	`SnpEff 5.2c (build 2024-04-09 12:24), by Pablo Cingolani`
Command line arguments	`SnpEff monkeypox /home/yassin/tfe/mpox/04_variant_calling/SRR30316267_variants.vcf -s /home/yassin/tfe/mpox/04_variant_calling/snpEff_summary.html`
Warnings	0
Errors	0
Number of lines (input file)	98
Number of variants (before filter)	98
Number of non-variants (i.e. reference equals alternative)	0
Number of variants processed (i.e. after filter and non-variants)	98
Number of known variants (i.e. non-empty ID)	0 (0%)
Number of multi-allelic VCF entries (i.e. more than two alleles)	0
Number of annotations	1,009
Genome total length	197,209
Genome effective length	197,209
Variant rate	1 variant every 2,012 bases

Variants rate details

Chromosome	Length	Variants	Variants rate
NC_063383.1	197,209	98	2,012
Total	**197,209**	**98**	**2,012**

Number variants by type

Type	Total
SNP	88
MNP	4
INS	2
DEL	4
MIXED	0
INV	0
DUP	0
BND	0
INTERVAL	0
Total	98