TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ADN	Acide désoxyribonucléique
API	Application Programming Interface
ARN	Acide ribonucléique
AWS	Amazon Web Services
BAM	Binary Alignment Mapping
CADD	Combined Annotation Dependent Depletion
CGP	Cancer Genome Project
ClinGen	Clinical Genome Ressource
CNV	Copy number variations
CPIQ	The Clinical Pharmacogenetics Implementation
CSS	Cascading Style Sheet
dbSNP	Short Genetic Variations Data Base
EBS	Elastic Block Storage
EC2	Elastic Compute Cloud
EMBL	The European Bioinformatics Institute
EMR	Elastic Map Reduce
ENSEMBL	Joint Project with EMBL
ETL	Extract Transform and Load
ExAc	The Exome Aggregation Consortium
FTP	File Transfer Protocol
gnomAD	The Genome Aggregation Database

GOAT	Genetic Output Analysis Tool
GWAS	Genome Wide Association Study
HDFS	Hadoop File System
HTTP	Hypertext Transfer Protocol
ICGC	International Cancer Genome Consortium
IGV	Integrative Genomics Viewer
INDEL	Insertion and deletion
JSON	JavaScript Object Notation
JSX	Java Serialization to XML
MVC	Model View Controller
NGS	Séquençage nouvelle Génération
ORM	Object relational mapping
PharmGKB	Pharmacogenetics Knowledge Implementation
RSID	Biomarqueur génétique
SAM	Sequence Alignment Mapping
SNP	Single nucleotide polymorphism
SPA	Single Page Application
SQL	Structured Query Language
S3	Simple Cloud Storage
TCGA	The Cancer Genome Atlas
UCSC	University of California at Santa Cruz
VCF	Variant Calling Format
WBS	Work BreakDown Structure

INTRODUCTION

Avant les années trente, les généticiens reconnaissaient que des molécules spécifiques étaient porteuses d'information génétique, et ce bien avant les chimistes. Il était établi que les chromosomes contenaient de l'acide désoxyribonucléique (ADN) ; mais la preuve que ce dernier était porteur de l'information génétique restait à venir. (Watson J, 2012)

C'est ainsi qu'entre 1930 et 1963, plusieurs scientifiques tels que Oswald T. Avery (1944), Erwin Chargaff (1949), Rosalind Franklin (1950), Alfred D. Hershey (1950), Martha Chase (1952), William W. Stahl (1958), Francis Crick et James Watson (1963) ont réalisé différentes expériences qui ont permis de démontrer non seulement que l'ADN est la fondation universelle de l'information génétique, mais aussi qu'il possède une structure en double hélice, et qu'il fonctionne telle une matrice pour les molécules d'ARN qui, à leur tour, permettent de déterminer l'ordre des acides aminés dans la suite de protéines. (Watson J, 2012)

Par la suite, dans les années 1960, quand la molécule d'ADN a été élucidée, on a aussi commencé à comprendre le fonctionnement de la séquence d'ADN, et par quels moyens on pouvait déterminer des phénotypes liés à des maladies. (Watson J, 2012)

De nos jours, les techniques de séquençage de nouvelle génération (NGS) ont permis de séquencer le génome complet de l'être humain, nous orientant ainsi vers une nouvelle ère ; celle de la médecine personnalisée.

Cette approche vise à proposer le traitement le mieux adapté à chaque patient en se basant sur les caractéristiques d'un ensemble de profils moléculaires établis comme des techniques de cartographie de nouvelle génération, dont le séquençage du génome. Il ne s'agit donc plus d'un recadrage d'une pratique médicale qui conduirait à une relation de plus grande proximité entre un médecin et son patient, mais plutôt d'une médecine techno scientifique qui associe l'acquisition et le stockage d'une grande quantité d'informations, d'analyses statistiques et des traitements bio-informatiques de ces mégas données. (Billaud, 2015)

Un des domaines le plus prometteurs de cette nouvelle pratique est en cancérologie. Elle consiste à traiter chaque patient selon les caractéristiques génétiques et biologiques de sa tumeur, tout en tenant compte de l'environnement du patient, de son mode de vie, etc. (Paci, 2013)

Le cancer englobe plusieurs maladies qui dérivent de mutations somatiques, telles que des substitutions, des indels (insertions ou délétions dans la séquence d'ADN), des gènes d'amplification focaux, des délétions homozygotes, et des gènes de fusion, mais encore de modifications épigénétiques, transcriptomiques, et protéomiques qui se sont accumulés au sein du génome des cellules cancéreuses. Ces altérations impliquent plusieurs processus cellulaires qui sont caractérisés, entre autres, par une signalisation proliférante, une résistance à l'apoptose, une induction de l'invasion et des métastases, et de la néoangiogenèse. La perte ou le gain de fonction somatique sont surreprésentés dans une région génomique spécifique, ce qui peut indiquer le potentiel rôle suppressif ou oncogénique. (Klonowska K, 2016)

Les récentes avancées technologiques, particulièrement le NGS, ont permis de faire de grandes avancées dans ce domaine. Celles-ci ont permis de mettre en place plusieurs projets centrés sur le cancer tels que *Cancer Genome Project (CGP*), le *Cancer Genome Atlas (TCGA)* et l'*International Cancer Genome Consortium (ICGC*). Ces projets ont été initiés afin de pouvoir analyser des données génétiques, épigénétiques et protéomiques à travers des milliers d'échantillons de cellules cancéreuses. Le but étant de rendre disponibles publiquement des bases de données contenant des données oncogénétiques à des fins d'analyse, de comparaison et de compréhension des mécanismes menant au développement d'un cancer. Il est donc capital que tous les chercheurs puissent extraire, analyser, interpréter et visualiser ces données. (Klonowska K, 2016)

CHAPITRE 1

REVUE LITTÉRAIRE

1.1 Le besoin de visualisation

Tel que présenté à la section précédente, les techniques de nouvelle génération permettent aux chercheurs l'exploration de nouvelles avenues de traitement à l'aide des informations génétiques. Cependant celles-ci génèrent une très grande quantité de données qu'il faut être en mesure d'extraire, d'analyser, de visualiser, d'interpréter et possiblement ré analyser.

Les chercheurs souhaitent analyser leurs données en temps réel afin de pouvoir en tirer des conclusions dans un délai rapide pour soigner leurs patients. Néanmoins, de nos jours, bien que plusieurs tâches puissent être automatisées, il reste encore plusieurs étapes exigeant une intervention humaine, ce qui limite l'accès et la rapidité des décisions quant à d'éventuelles directions à adopter lors de la recherche d'un traitement. (Nielsen CB, 2013)

À titre d'exemple, plusieurs chercheurs utilisent le visualisateur IGV (*Integrative Genomics Viewer*) offert gratuitement par le Broad Institute (Robinson JT, 2011) qui permet de visualiser des parties du génome. Ce logiciel permet de rechercher des mutations génétiques telles que des substitutions de base, des indels, des gènes d'amplification focaux, des délétions homozygotes et des gènes de fusions dans des échantillons cliniques.

Les données représentées à la figure 1.1 proviennent du « *Cancer Genome Atlas (TCGA) »* (Weinstein JN, 2013), et cette figure est reproduite directement du manuel d'utilisation du logiciel IGV. La région du génome représenté est délimitée par l'étiquette 1 et mesure environ 40 mégas bases (c.-à-d. 40 000 000 de bases).

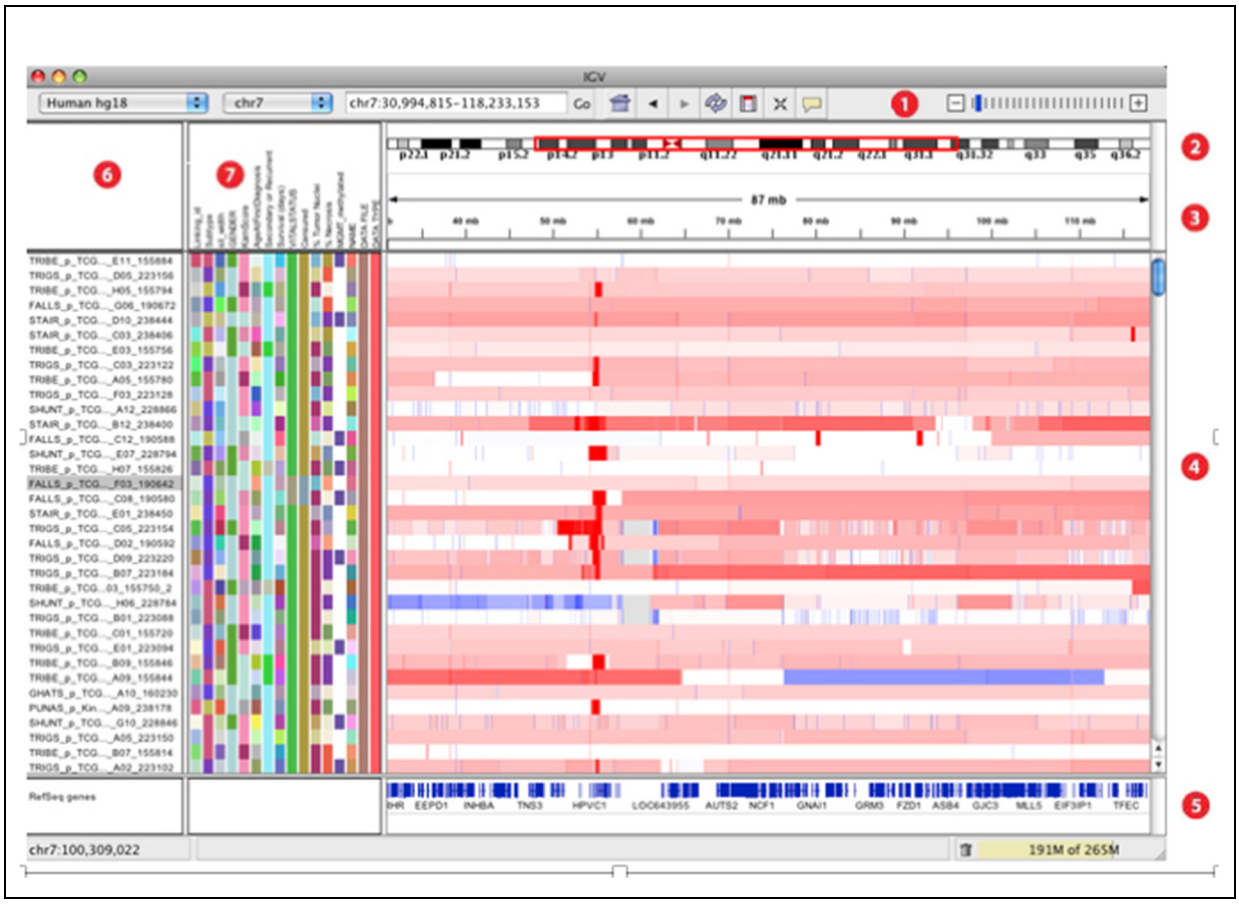

Figure 1.1 Image de l'interface de l'outil IGV. Tirée de Li J, 2014

Bien que ce logiciel soit très utilisé par la communauté scientifique, la navigation dans son interface utilisateur est difficile lors de l'exploration et de la localisation d'environ 50 000 mutations par patients. Beaucoup d'informations sont affichées, mais elles ne concernent qu'une seule mutation. De plus, avant de pouvoir visualiser ces données, elles doivent être converties du format original (c.-à-d. celui du chercheur) au format imposé par l'outil et décrit dans le guide d'utilisateur ; soit un fichier de type BAM (*Binary Alignment Map*) ou SAM (*Sequence Alignment Map*). Les fichiers de format BAM et SAM sont typiquement générés à l'aide des outils de la suite SAMTools (Li H, 2009) ou GATK (do Valle, 2016) du Broad Institute. En général, le chercheur lui-même ne peut les générer et ne peut les examiner puisque leur utilisation requiert une main-d'œuvre spécialisée (c.-à-d. l'implication de bio-

informaticiens) pour transformer ces formats en un fichier VCF (figure 1.3) qui lui, peut être lisible à l'œil nu. Les étapes menant à la génération de ce fichier sont indiquées à la figure 1.2.

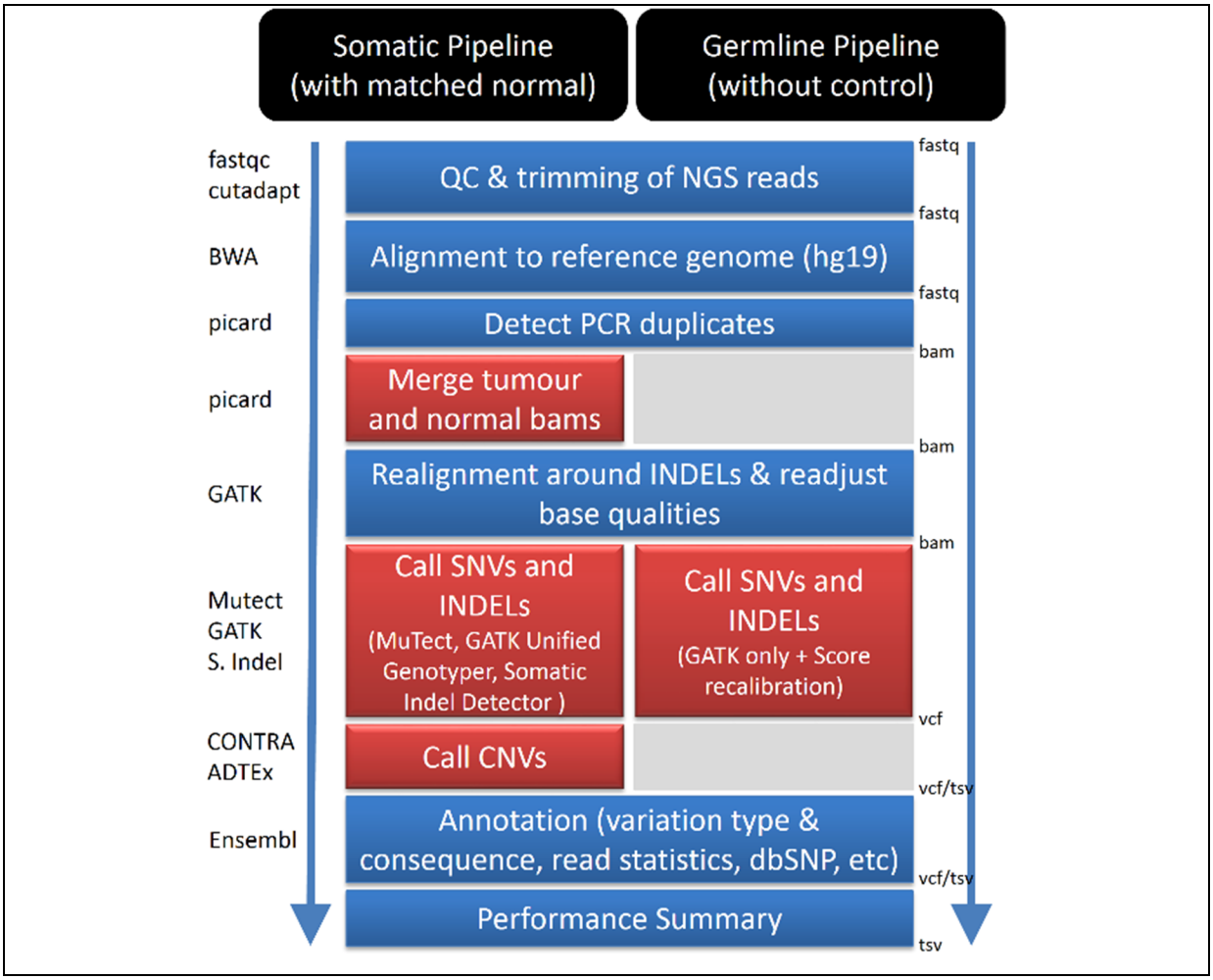

Figure 1.2 Schéma montrant les étapes du séquençage de l'exome. Tirée de Li J, 2014

La comparaison du génome d'un patient avec un génome de référence est toujours nécessaire et permet de déduire quels transcrits sont exprimés. L'alignement de l'ADN relativement au génome de référence se fait avec l'aide d'un logiciel tel que BWA (O'Rawe, 2013) lors de l'étape de prétraitement des données. Le fichier de résultat, qui est lisible par l'œil humain, est représenté à l'avant-dernière étape de la figure 1.2, c'est-à-dire l'étape « *Annotation (variation type & consequence, read statistics, dbSNP, etc))* », qui utilise un format VCF (*Variant Calling Format*).

Plusieurs chercheurs disposent du fichier des données génétiques de leur patient au format VCF pour la visualisation des données. Les logiciels libres actuellement disponibles permettant la visualisation de ce fichier sont : IGV (Robinson JT, 2011), LocusZoom (Pruim, 2010), UCSC Genome Browser (Kent, 2002) et TCGA Genome Browser (Weinstein JN, 2013).

Ces logiciels permettent une exploration de la région génomique, mais autour d'une seule mutation, alors que le VCF contenant les mutations détectées au niveau cellulaire en contient environ 50 000, ce qui impose au chercheur un effort additionnel d'exploration pour arriver à trouver la mutation ou les patrons d'expression génique responsable d'une maladie spécifique.

Une comparaison de la performance de l'utilisation de ces logiciels avait déjà été effectuée pour évaluer la performance lors de la recherche d'une seule mutation. Dans cette étude, LocusZoom affichait la performance la plus faible, avec 20 secondes pour afficher une seule mutation, alors que GOAT et IGV arrivaient à une seconde (Kanzki BS, 2016). Même avec ce résultat d'une seconde, un chercheur ou un médecin en oncogénétique devra investir entre 13 heures et 11.6 jours pour explorer tout le profil génétique de son patient, si ce dernier contient 50 000 mutations. L'effort requis ainsi que les délais afin d'obtenir un résultat est donc assez décevant.

Un autre inconvénient observé, lié à l'utilisation de ces logiciels, est la difficulté à parcourir et naviguer dans le flot d'informations affiché sur leurs interfaces utilisateurs. Bien que les équipes de chercheurs sachent quoi rechercher à l'échelle génomique, l'information affichée est tellement exhaustive, qu'il faut souvent faire des recherches supplémentaires sur internet pour s'assurer qu'on visualise la bonne information. En l'occurrence, considérons le *UCSC Genome Browser,* (*notez que TCGA Genome Browser affiche ces mêmes informations*), ce logiciel affiche de l'information exhaustive sur les régions génomiques.

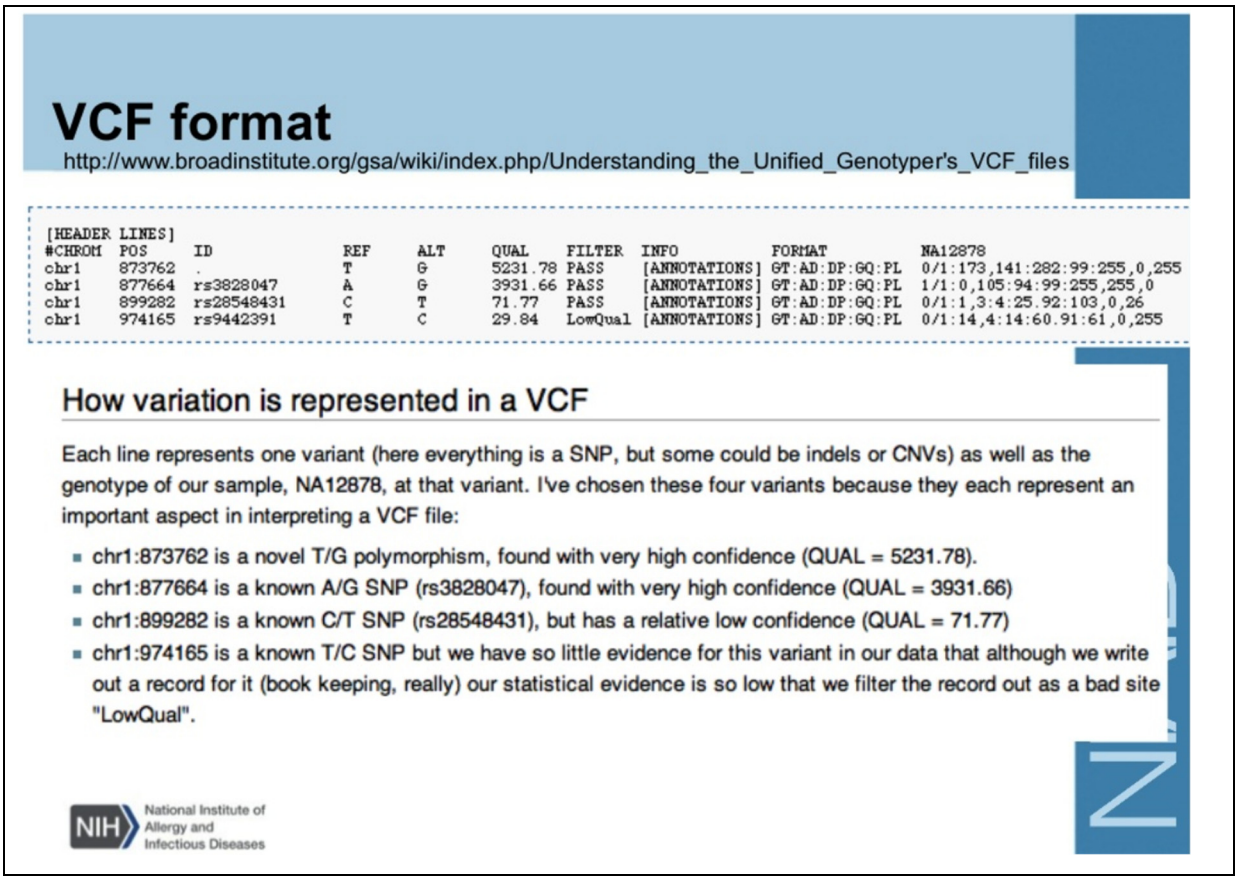

Figure 1.3 Image et explication du format VCF d'après le Broad Institute. Tirée de Quinones, 2015

Il y a plusieurs tutoriels sur YouTube, et d'autres sites Web, qui tentent d'expliquer comment apprendre à naviguer sur son interface utilisateur pour retrouver rapidement l'information que l'on recherche. La grande quantité de ces sites de support sont une indication que l'interface utilisateur et même la documentation existante ne suffisent pas à expliquer simplement son utilisation pour un chercheur typique. De plus il existe un temps de latence (c.-à-d. un problème de performance du rendu de l'interface utilisateur) lors de la navigation et de l'exploration des différentes mutations à travers l'outil. Un aperçu de l'interface utilisateur de ce logiciel est présenté à la figure 1.4.

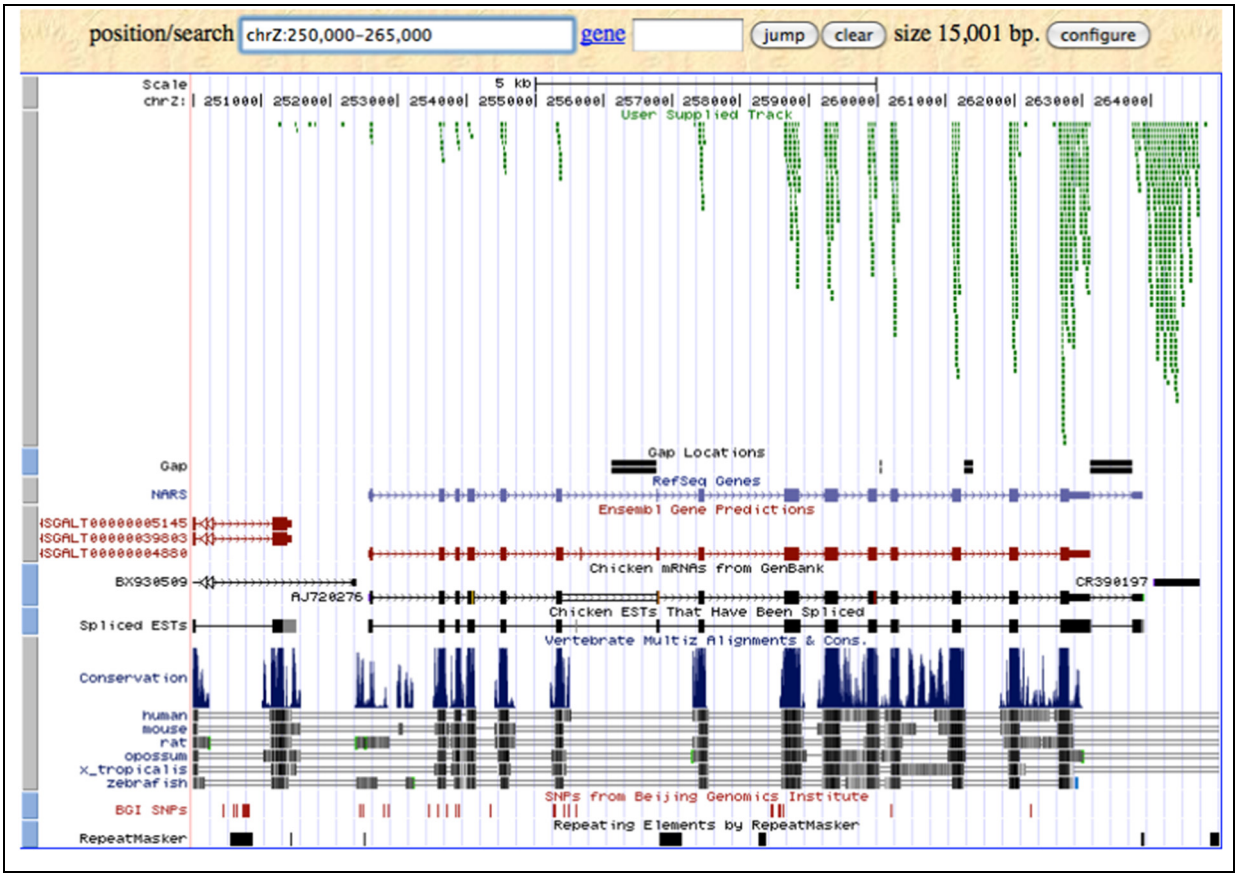

Figure 1.4 Image de résultat de UCSC Genome Browser. Tirée de THE REGENTS OF THE UNIVERSITY OF CALIFORNIA., s.d.

Après avoir essayé ces logiciels qui sont disponibles librement dans le domaine de la visualisation de la génomique, c'est-à-dire l'*IGV*, le *UCSC Genome Browser*, et le *TCGA Genome Browse,* on conclut que bien qu'ils soient couramment utilisés, ces logiciels ne répondent pas complètement aux besoins des chercheurs. Les chercheurs sont constamment à l'affut de nouveaux logiciels de visualisation du génome, qui seraient plus efficaces, performant, et leur permettraient d'avoir une vision globale des mutations trouvées sans avoir recours à du personnel spécialisé pour les opérer. Cela leur permettrait d'établir plus rapidement des profils génétiques liés à des maladies et de traiter de manière plus rapide leurs patients.

Finalement, en plus des observations déjà faites, il y a une autre situation qui rend difficile leur utilisation, c'est-à-dire la taille toujours grandissante des jeux de données qui requiert de plus en plus de ressources et de compétences en informatique pour leur traitement.

1.2 Taille des données générées en génétique et oncogénétiques

À la suite à l'essai des logiciels les plus populaires utilisés en visualisation génomique (que nous avons décrits à la section précédente), il est important de traiter du problème de la taille des jeux de données communément utilisés dans ce domaine par les chercheurs. À titre d'exemple, la base de données européenne de séquence de nucléotides, c'est-à-dire la base de données EMBL-Bank, qui regroupe, organise et distribue toutes les séquences génétiques publiées connait une grande croissance. Il a été reporté qu'elle double de taille tous les 10 à 12 mois. Cette croissance, en décembre 2001, a établi sa taille, à 15 milliards de bases (Stoesser (G.), 2002). Or si en 2001, elle atteignait déjà 15 milliards de base, et qu'elle double de taille à chaque année environ, il y aurait près de 225 milliards de bases répertoriées dans cette base de données en 2017.

Il a été aussi observé que les techniques récentes de séquençage NGS, ainsi que les bases de données, telles qu'EMBL-Bank, ne sont pas les seules à générer des quantités importantes de données qui doivent être prises en compte par les chercheurs. Il y a aussi d'autres domaines de recherche tels que la transcriptomique (c.-à-d. l'étude de l'ADN transcrit en ARN), la protéomique (c.-à-d. la recherche de la caractérisation des protéines), et la métabolomique (c.-à-d. les études des métabolites) qui génèrent aussi des quantités importantes de données de leurs côtés. Ainsi, en recherche génomique et génétique, rechercher uniquement dans les fichiers de résultats des technologies NGS ne suffit pas. Pour profiter des connaissances des autres chercheurs, il est essentiel de pouvoir consulter et comparer ses résultats aux données rendues disponibles dans toutes les bases de données publiques contenant de l'information de référence.

Conséquemment, les algorithmes permettant de traiter cette quantité de données ont souvent une complexité quadratique (O^2) et évoluent selon la loi de Moore (c.-à-d. la puissance de calcul, qui double tous les 18 mois). Les technologies de l'information actuelles, utilisées par les laboratoires de recherche typiques, ne suffisent plus pour répondre à ces volumes croissants. Conséquemment, il devient donc essentiel de disposer d'architecture logicielle et technologique spécialisée pour traiter facilement et efficacement toutes ces données. (Stoesser (G.), 2002)

Comme nous venons de le voir, la grande taille des données à traiter ne limite pas seulement la vitesse des découvertes scientifiques ; mais aussi le type d'analyses qui peuvent être effectuées sur une interface utilisateur. Bien sûr, la taille des données a une conséquence directe sur la quantité d'analyses qu'il est possible d'effectuer dans une journée. Par exemple pour créer des modèles de prédiction, par exemple pour faire du « *Deep Learning* », des outils d'analyses statistiques, tels que R (Robert., 2008), ont actuellement de la difficulté à traiter ces grandes quantités de données efficacement. D'autres technologies doivent être explorées. Il existe d'autres langages de programmation, tels que le langage Python (Zelle John M, 2004) qui est adapté pour effectuer ce genre d'analyse. Ainsi, malgré la popularité grandissante de la librairie Python pour la bio-informatique BioPython, elle est reconnue pour ses limites de traitement causées par son « *single thread* » qui peut difficilement distribuer ou effectuer le travail de manière efficace sur des grappes d'ordinateurs modernes.

Il devient donc incontournable de s'intéresser autant aux logiciels utilisés en génomique, ainsi qu'aux bases de données publiques disponibles couramment et utilisées dans ce champ de recherche pour comprendre comment rendre mieux disponible l'information aux chercheurs et traiter efficacement ces mégas donnés.

1.3 Outils et bases de données publiques disponibles

En oncogénétique, plusieurs bases de données publiques ont été mises à la disposition de la communauté scientifique afin de permettre l'identification du catalogue complet des

altérations somatiques propres au génome, l'épigénome et le transcriptome des échantillons tumoraux. C'est une étape cruciale pour l'extraction des altérations et de leurs relations entre elles par les experts afin de mieux comprendre les mécanismes menant à une néogenèse.
De plus, certains logiciels permettent une visualisation intuitive des altérations somatiques et de leurs significations cliniques. Le tableau A-1, en annexe, résume tous les logiciels de visualisation disponibles publiquement, en spécifiant le type de visualisation, la plateforme sur laquelle ils fonctionnent et le genre de données qu'il est possible d'utiliser avec ces derniers. (Klonowska K, 2016)

On pourrait penser, en parcourant rapidement la liste énumérée au tableau A-1, que ces logiciels libres, mis à la disposition des chercheurs, puissent satisfaire entièrement leurs besoins d'extraction, d'analyse et de visualisation des données. Pourtant les publications récentes ainsi que le développement de logiciels de visualisation n'ont cessé d'augmenter depuis quelques années. Pour ne citer que quelques-uns, non répertoriés au tableau A-1 et qui ont été lancés en 2016 seulement, on retrouve : ***UCSC Xena*** un portail Web qui permet l'extraction et l'analyse d'échantillons tumoraux (Goldman M, 2016), ***caOmics*** une librairie en langage R pour la visualisation de donnée tumorale sous forme de « *heatmap* » ou de réseau (Zhang H, 2016), ***ProteinPlant*** qui permet l'exploration d'altérations génomiques dans les cas de cancers pédiatriques (Zhou X, 2016), et aussi ***GlioVis***, un portail Web disponible pour la visualisation et l'analyse d'échantillons tumoraux provenant du cerveau (Bowman RL, 2016), et finalement ***Cascade*** qui permet la visualisation dans le domaine de l'oncogénomique (Shifman AR, 2016).

Bien que tous ces logiciels libres soient disponibles, le besoin d'un outil puissant et rapide de visualisation qui peut traiter des mégadonnées et qui comporte une interface utilisateur simple demeure réel, surtout pour éliminer l'intervention de la main-d'œuvre spécialisée et du préformatage des données avant de pouvoir utiliser le visualisateur.

En plus des logiciels présentés au tableau A-1 à l'annexe I, il faut aussi prendre en compte les bases de données publiques qui permettent d'extraire des jeux de données précises afin de faire

des analyses plus poussées. Les tableaux A-2 et A-3 nous donnent un résumé de ces outils, le type de données qu'on retrouve sur ces dernières, et dans certains cas, le type d'accès requit pour chacun afin de procéder aux extractions.

À la suite de l'évaluation des logiciels répertoriés dans ces deux tableaux, on note assez rapidement que les données les plus faciles à extraire sont celles situées au niveau III et IV. C'est-à-dire qu'il s'agit de données déjà interprétées, et finales. En général, les données de niveau I et II peuvent être obtenues seulement à l'aide d'une permission spéciale, car elles sont privées. L'information détaillée concernant les mutations somatiques se retrouve dans les données de niveau III et IV. Une référence aux jeux de données utilisées est fournie à titre informatif. Il existe aussi des bases de données publiques, dont l'accès est sans restriction et d'autres logiciels répertoriés, non cités précédemment, qui ont été placés dans le tableau A-2 à l'annexe I à titre indicatif.

Tableau 1.1 Base de données publique

gnomAD	Contient 126 216 séquences exomiques, 15 136 (Whole genome sequencing) WGS provenant d'individus non liés entre eux. (Lek M, 2016)
ExAc	Contient 60 706 génomes d'individus non liés entre eux. (Lek M, 2016)

Tableau 1.2 Outils utiles aux cliniciens

CADD	Combined Annotation Dependent Depletion. Logiciel qui permet d'avoir un score pour des snps (*single nucleotide polymorphism*) délétères, ainsi que des indels chez l'humain. (Kircher M, 2014)
CPIQ	The Clinical Pharmacogenetics Implementation Consortium. L'information de ce site est révisée constamment par les paires du domaine. Le but de ce consortium est d'adresser les obstacles à l'implantation de la pharmacogénétique dans la pratique clinique (PharmGKB and PGRN, 2018).
ClinGen	Financé par le National Institute of Health (NIH). Le but de ce consortium est de rassembler les gènes identifiés dans des maladies à partir de tests cliniques. (Hunter JE, 2016)
ClinGen Pathogenecity Calculator	Score de pathogénicité basé sur les données de ClinGen. (Hunter JE, 2016)
PharmGKB	Base de données de pharmaco génomique qui inclut le dosage, les médicaments, des gènes qui peuvent être ciblés lors de traitements. (Gammal RS, 2015)
Hétérogénéité tumorale	Score pour l'hétérogénéité de la tumeur. (Bedard PL, 2013)

La nécessité d'intégrer ces logiciels, et/ou ces données (c.-à-d. d'utiliser plus d'un de ces logiciels) lors d'une analyse dépend principalement de ce que l'on cherche à démontrer, et de la quantité de ressources dont le chercheur dispose. Il faut aussi noter que toutes ces bases de données ont en commun le fait qu'elles rendent disponibles des données à jour qui sont nécessaires aux analyses et conséquemment elles sont typiquement très utilisées en ce moment.

Toutefois aucune de ces bases de données ne permet aux chercheurs de charger leurs propres données pour une analyse plus poussée par rapport à ce qui a déjà été publié. Ils doivent le faire eux-mêmes dans leur laboratoire avec leurs propres outils et moyens.

1.4 Problématique

Le but de la revue de littérature a été d'inventorier les logiciels libres de visualisation génétique et d'identifier leurs limitations actuelles. Ensuite, les besoins actuels des chercheurs ont été identifiés dans le but de concevoir un prototype expérimental d'un nouveau type de visualisateur qui pourra être utilisé par une équipe de recherche sans avoir recours à des bio-informaticiens, ou informaticiens pour effectuer ses analyses. Ceci nous oriente donc vers deux questions de recherche :

1) À quoi ressemblerait l'interface utilisateur qui répondrait aux besoins de visualisation des chercheurs en médecine personnalisée pour visualiser des profils génétiques de patients, et quels types de fonctionnalités devrait-elle avoir ?
2) Serait-il possible d'afficher le profil génétique complet d'un patient en un temps raisonnable par rapport aux autres outils disponibles qui n'affichent qu'une seule mutation à la fois ?

En ce qui a trait à la première question nous allons faire une recherche pour trouver les requis nécessaires pour l'affichage du profil génétique complet d'un patient. Nous nous concentrerons sur le type d'analyses et sur les résultats pertinents généralement pris en considération par les chercheurs afin de les trouver. Parmi les attributs de qualité visés pour le prototype logiciel du nouveau visualisateur et son interface future, nous avons établi que la performance est la caractéristique la plus importante à démontrer.

Pour répondre à la deuxième question, nous expérimenterons avec le format génétique ADAM et avec la technologie Big Data Spark pour l'extraction rapide et efficace d'une très grande quantité de données génétiques.

Les étapes d'expérimentation proposées sont présentées à la figure 1.5 suivante :

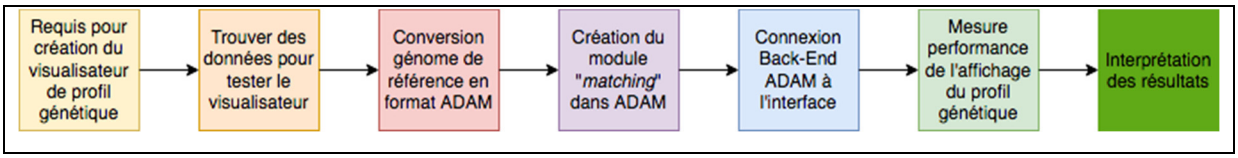

Figure 1.5 Étapes d'expérimentation pour vérifier si GNOMEVIEWER répond aux besoins de visualisation et de performance des chercheurs en oncogénétiques

À la suite de la conception et de l'expérimentation proposées, nous pourrons constater si notre proposition d'un nouveau type de visualisateur, nommé GNOMEViewer répond tant au besoin de visualisation que de performance des chercheurs en génétiques.

1.5 Conclusion

Les chercheurs en génétique ont un grand besoin de visualiser et d'extraire des données et ils souhaitent pouvoir le faire de manière autonome, interactive, efficace et dans un temps raisonnable. Une limitation inhérente aux différents laboratoires de recherche se retrouve au niveau de l'équipe de recherche elle-même ; en effet' elle n'a souvent pas la formation pour utiliser les logiciels disponibles, et pour effectuer leurs propres analyses et visualiser leurs données sans l'aide de spécialistes en bio-informatique.

Par ailleurs, les différentes bases de données publiques disponibles contiennent des jeux de données exhaustives qui sont d'une grande utilité aux chercheurs en oncogénétique, et plusieurs logiciels libres ont été développés afin de pouvoir les visualiser, extraire leurs données et les analyser. Mais les technologies de l'information disponible aux chercheurs, l'interface utilisateur inadéquate des logiciels libres disponibles (c.-à-d. la visualisation d'une seule mutation à la fois), la latence au niveau de la performance lors de la visualisation ainsi que leur difficulté à traiter de très grandes quantités de données causent des maux de tête aux chercheurs du domaine.

Le but de cette recherche est de concevoir un prototype de visualisation interactif et facile d'utilisation, ou les informations des bases de données publiques seront disponibles pour un examen rapide afin d'accélérer le processus de découverte de marqueurs génétiques.

CHAPITRE 2

CRÉATION DE L'INTERFACE POUR VISUALISATION DU PROFIL GÉNÉTIQUE DE PATIENTS

2.1 Prérequis pour la création du module de visualisation de profil génétique

Tel que mentionné à la section 1.2, les logiciels libres de visualisation génétique disponibles aujourd'hui ne permettent de visualiser uniquement une mutation à la fois. Donc lorsqu'un médecin (ou un chercheur) fait la demande de séquençage de l'exome d'un patient, il se concentrera uniquement sur les mutations communes déjà détectées, car il n'a ni le temps ni les ressources pour parcourir le fichier de résultat au complet. Le fichier contenant le profil génétique d'un patient contient entre 30 000 et 50 000 mutations.

Conséquemment, le visualisateur génomique idéal qui permet de voir le profil complet d'un patient devrait permettre de téléverser et afficher le contenu des résultats du séquençage exomique afin que le chercheur puisse naviguer interactivement à l'échelle chromosomique.

Mais étant donné que certaines mutations seront d'intérêt pour classifier le patient avec son profil génétique, les fonctionnalités contenues dans les autres outils doivent être maintenues. C'est-à-dire la fonction existante de visualisation d'une seule mutation à la fois.

Ces deux fonctionnalités devraient être disponibles sur la même interface utilisateur. Il est donc nécessaire de concevoir un nouveau type d'affichage qui puisse satisfaire ces deux conditions.

La conception envisagée, étudie la possibilité de faire l'affichage d'un caryotype humain pour afficher le profil génétique d'un patient. En général, cette représentation est utilisée afin de mettre en évidence des anomalies chromosomiques, ou pour définir certaines caractéristiques d'un individu. (Schrock, 1996).
C'est ainsi que l'affichage apparaissant à la figure 2.4 a été créé.

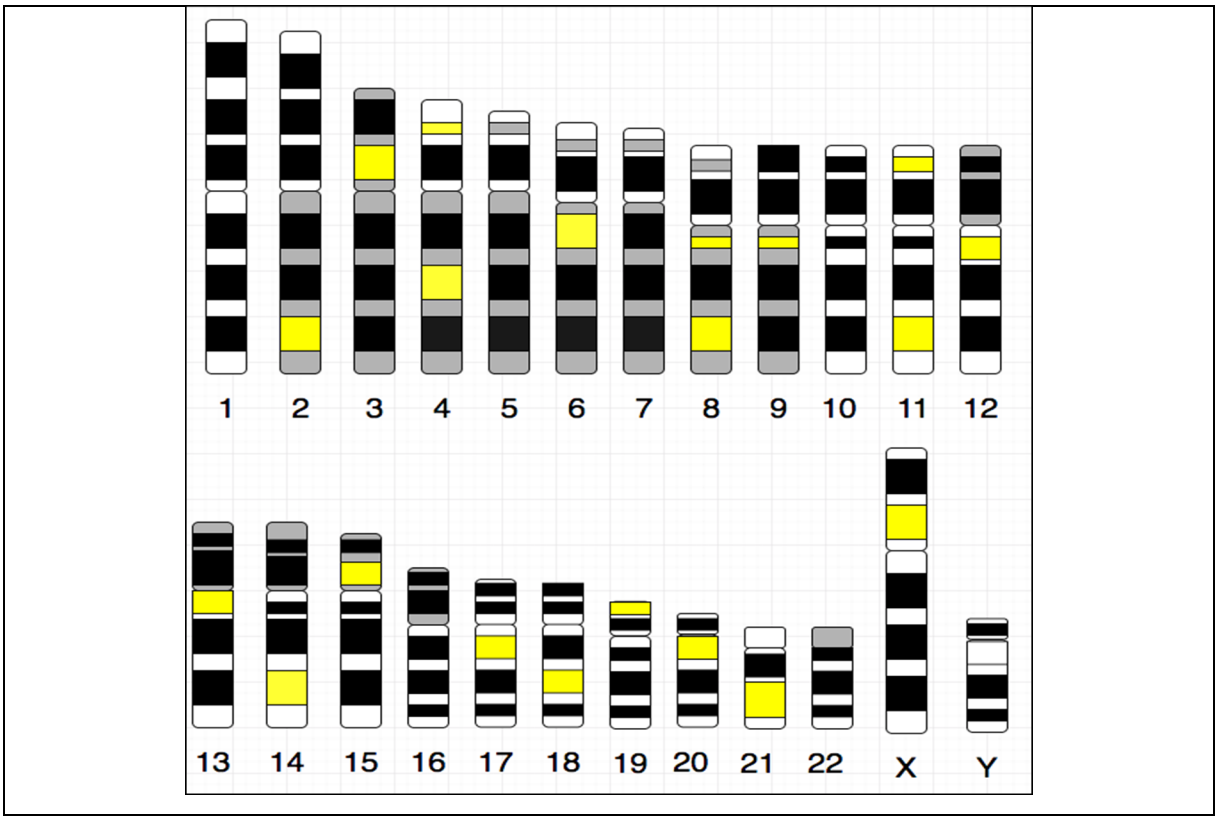

Figure 2.1 Maquette du graphique souhaité pour l'affichage immédiat des variants correspondants du génome de référence avec GNOMEViewer

Sur cet affichage, on peut facilement voir la localisation des différentes mutations et de plus on peut voir si ces mutations se trouvent dans les parties codantes de l'ADN. En effet, les zones noires de la figure 2.1 représentent les exons (c.-à-d. les parties codantes de l'ADN) et les zones jaunes représentent l'emplacement des biomarqueurs, détectés dans le fichier VCF, contenant le profil génétique d'un patient donné. Bien que cet affichage ne présente pas beaucoup de détails, il permet la localisation de toutes les mutations. Il permet donc au médecin ou au chercheur d'avoir le profil mutagénique de son patient en un seul coup d'œil.

Le cancer étant une maladie liée à des traits génétiques complexes, il a été observé que plusieurs régions de l'ADN interagissent entre elles pour entretenir la tumeur. De ce fait, en ayant accès à ce genre de représentation graphique, il sera plus facile de détecter leurs localisations afin de les explorer.

Les zones jaunes de la figure 2.3 sont des grappes de mutations qui correspondent à des positions sur le génome humain. Ces grappes peuvent être liées à des maladies ou à des mutations connues, ou encore non explorées. À partir de cette interface utilisateur, il serait donc possible, d'une part, de filtrer les mutations pour voir lesquelles sont inconnues et requièrent une investigation ou une recherche en profondeur, et d'autre part, lesquelles sont communes en les filtrant directement à l'aide d'une fonctionnalité sur cette interface.

À l'aide de cette interface utilisateur, le chercheur voudra peut-être explorer certaines d'entre elles pour avoir plus d'informations en sélectionnant la grappe d'intérêt qui affichera un nouveau type de graphe présenté à la figure 2.4. Le chercheur pourra ainsi vérifier le taux de recombinaison des allèles, le lien entre les gènes, et en même temps voir interactivement, comme pour l'approche précédente, les types de mutations et les maladies associées à chacune en les filtrant de manière interactive. Ces informations sont de la plus haute importance pour les chercheurs pour les raisons suivantes :

- Celles-ci permettent de connaitre les régions du génome qui sont stables. (Julie Hussin, 2013) ;
- La recombinaison étant un mécanisme important pour la réparation de l'ADN, donc un défaut dans le taux de recombinaison rend vulnérable à certains cancers. (Walsh CS, 2015) ;
- Cela permet l'identification de la localisation de mutation somatique de novo (nouveaux types de mutations) ;
- Ainsi que de voir les mutations communes à un ou plusieurs cancers sur une seule et même page.

Donc l'affichage d'une mutation unique sera aussi possible en naviguant sur le chromosome sélectionné, et ce interactivement. Le chercheur pourra ainsi faire une exploration du chromosome dans son ensemble et ainsi réduire considérablement l'effort de recherche actuel qui se situe entre 13 heures et 11.6 jours.

2.2 Réutilisation d'architecture et des données d'un projet précédent

Un projet de recherche antérieure, effectuée aussi à l'ÉTS avec le professeur April et le Dr Pavel Hamet, avait produit un premièr prototype d'application de visualisation qui avait pour but d'afficher des données génétiques de type GWAS (*Genome Wide Analysis Study*). Il s'agit du prototype d'application GOAT (*Genetic Output Analysis Tool*), un prototype expérimental Web permettant au chercheur de visualiser, de partager et d'annoter les données GWAS générées par leur laboratoire. Ce logiciel expérimental possédait une architecture logicielle en couche telle que présentée à la figure suivante :

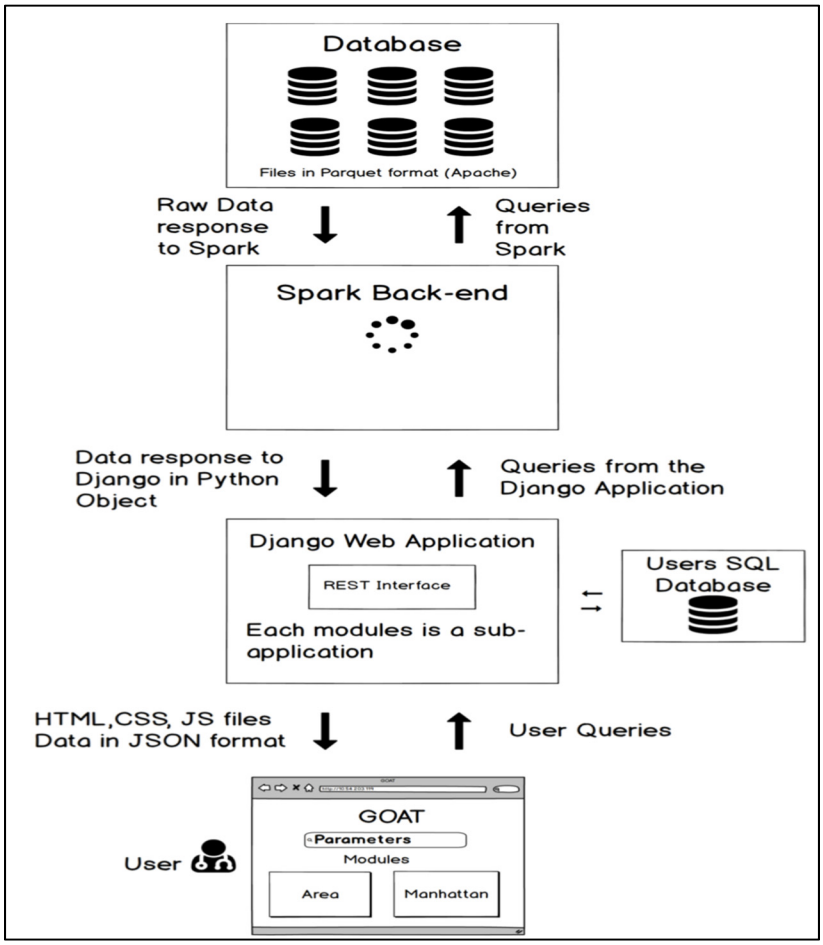

Figure 2.2 Architecture en couche de GOAT. Tirée de Lauzon D, 2016

GOAT a une interface pour la visualisation interactive qui permet d'identifier des associations significatives, à travers plusieurs variants ou SNP (c.-à-d. Single Nucleotide Polymorphism), appartenant à un phénotype donné. À partir du génome de référence, il offre la possibilité de visualiser les données génétiques à l'aide de trois modules logiciels (Kanzki BS, 2016) :

1) **Un module de visualisation de GWAS (GeneQuery)**, où un graphe de type Manhattan interactif prend forme, en temps réel, lorsque l'utilisateur recherche un SNP ou un gène, tout en affichant le phénotype le plus significatif de la base de données. Le graphe résultant est accompagné d'un tableau contenant les données détaillées présentées sur le graphe ; le tout accompagné d'un menu déroulant qui permet de rechercher d'autres phénotypes. Il est aussi possible de filtrer, de sélectionner, d'afficher et de sauvegarder les données les plus significatives, incluant les graphes et le tableau d'information détaillée ;

2) **Un module de visualisation de région génomique (AreaSelection),** où un graphe interactif représentant une région génomique sur un intervalle de trois-millions de paires de bases nucléotidiques représente la région associée aux biomarqueurs identifiés dans le module « GeneQuery ». Les maladies complexes, telles que le cancer et le diabète de type II, possèdent couramment des biomarqueurs qui ont des interactions avec d'autres régions génomiques. Le code source de ce module permet de présenter, directement sur le graphe, tous les gènes environnants sur un intervalle à +/— 1 500 000 paires de bases afin de faire une analyse plus poussée. Il est donc possible de changer de position et de chromosome afin de visualiser d'autres régions. Les graphes et tableaux produits par ce module sont non seulement interactifs, mais aussi permettent la filtration, la sélection, un affichage et la sauvegarde de données d'intérêt ;

3) **SNPS Gènes :** Ce dernier module de GOAT se spécialise dans la fonctionnalité qui permet de sélectionner tous les variants génétiques propres à un gène spécifique.

Après l'examen des codes sources de GOAT, nous avons décidé d'étudier l'affichage du profil génétique des patients en utilisant le génome de référence HG19 contenu dans sa base de données MySQL (A.B, 2001). Au début du projet de recherche GNOMEViewer nous disposions donc de la version 3 de GOAT pour nous inspirer de fonctionnalités antérieures disponibles.

Il a donc été décidé que le prototype logiciel GNOMEViewer aurait lui aussi une architecture en couche, mais avec une conception impliquant différents composants, tel que présenté à la figure suivante :

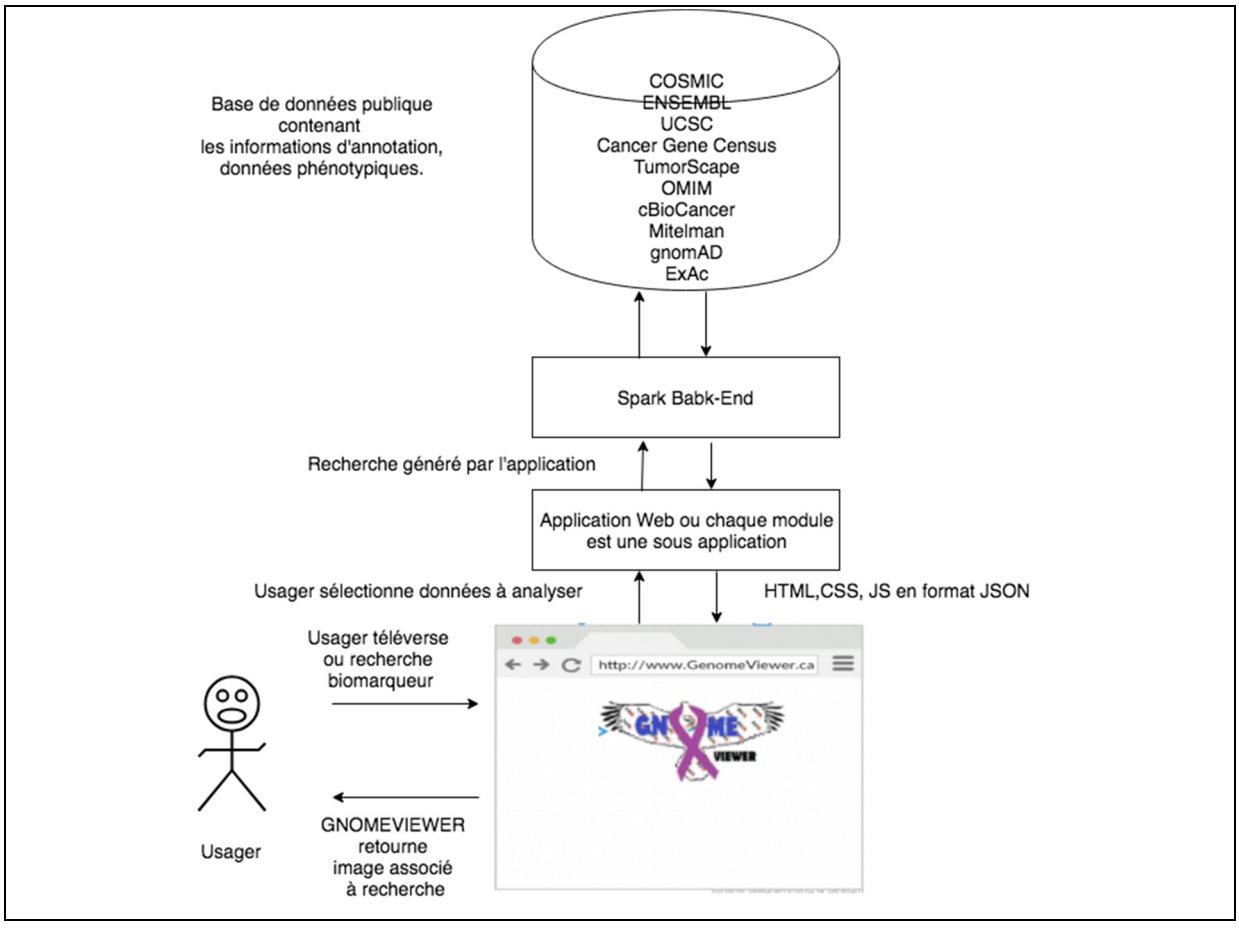

Figure 2.3 Architecture en couche de GNOMEViewer. Tirée de Kanzki B, 2017

À la différence de GOAT, la technologie Big Data Spark a été sélectionnée pour concevoir le « back-end » de GNOMEViewer, car il sera nécessaire d'afficher environ 50 000 mutations

génétiques sur caryotype humain en prenant en référence le génome humain. Et cette fois, plus de bases de données publiques seront intégrées au « *back-end* ».

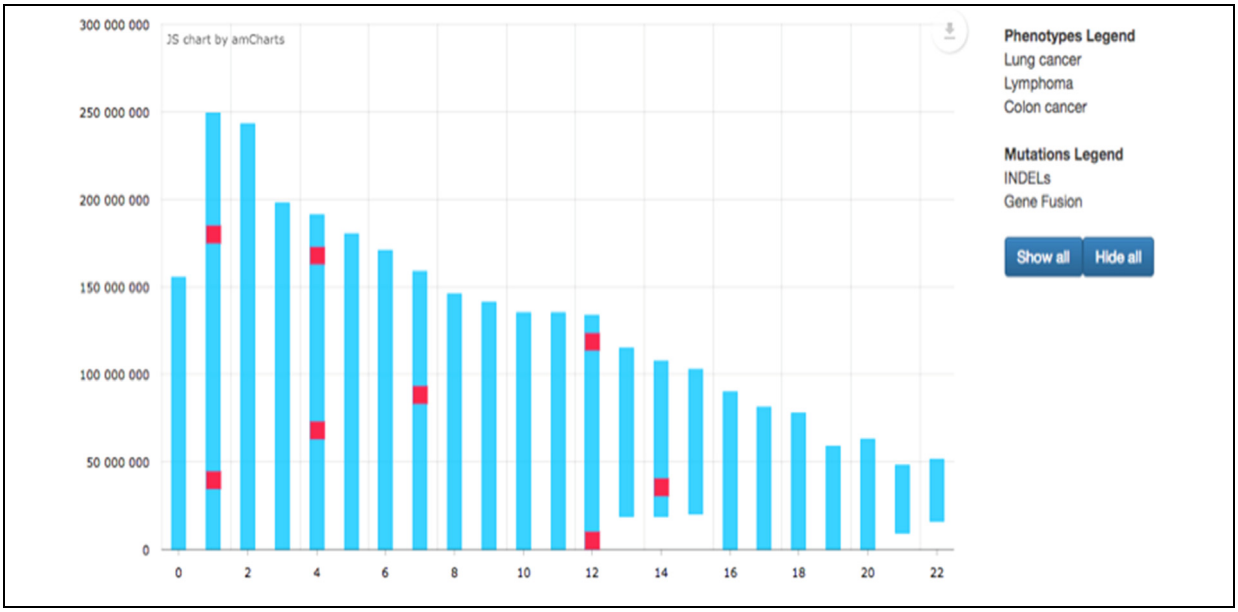

Figure 2.4 Affichage du profil mutagénique d'un patient

Le résultat d'un premier affichage, présenté à la figure 2.4, représente fidèlement la taille des chromosomes et la position des mutations détectées selon le génome de référence HG19 tel que contenu dans la base de données MySQL. À cause de l'utilisation d'une technologie de bases de données relationnelles, pour cette première preuve de concept, l'affichage du graphe a pris un peu plus de temps. Mais cela ne nuit en rien la convivialité du graphe qui offre des fonctionnalités interactives et qui démontre la fonctionnalité de visualisation requise pour GNOMEViewer.

Cette représentation graphique proposée est entièrement interactive, et réagit lors de la sélection des maladies en faisant apparaître ou disparaître les mutations détectées et les maladies auxquelles elles sont généralement associées dans le menu de droite.

Cette nouvelle fonctionnalité permettra aux chercheurs de détecter rapidement les patrons mutagéniques à l'échelle du génome d'un patient pour une dite maladie. Cela représentera une

amélioration importante dans ce domaine, car les autres logiciels libres disponibles permettent seulement de voir une position unique sur le génome pour une mutation donnée. Avec l'innovation de l'interface utilisateur de GNOMEViewer, le chercheur sera capable de visualiser à l'échelle du génome complet sur un seul graphe en sélectionnant et désélectionnant le menu de droite.

On peut donc constater que ce caryotype humain permet fidèlement l'affichage du profil génétique d'un patient avec environ 50 000 mutations, en plus de permettre une comparaison rapide avec d'autres profils trouvés dans des bases de données publiques.

Par contre, pour répondre aux besoins de performance d'un affichage rapide de ce profil génétique, nous discuterons du remplacement de la base de données MySQL, de la première preuve de concept vers une technologie Big Data beaucoup plus performante telle que Spark et le nouveau format ADAM proposé par l'Université Berkeley de Californie.

2.3 Conclusion

Étant donné que le profil génétique complet du patient doit être affiché, il est incontournable de trouver une manière rapide (c.-à-d. efficace) et conviviale qui permettrait de montrer la localisation des mutations génétiques des patients. L'affichage choisi est une représentation graphique du caryotype humain sous forme d'histogramme où il est possible de voir le profil mutagénique d'un patient.

En utilisant les données du génome de référence, il a été possible de concevoir et de tester cet affichage à l'aide d'une première preuve de concept. Pour s'assurer d'une performance adéquate, la technologie de base de données relationnelle ne suffira pas et une nouvelle technologie de base de données du domaine du Big Data sera expérimentée.

CHAPITRE 3

ADAM COMME MOTEUR DE RECHERCHE

3.1 Conversion vers ADAM

La performance des bases de données relationnelles est inadéquate pour ce projet de recherche, car elles comportent des limitations importantes pour être utilisables dans le contexte de mégadonnées. De plus, il a été mentionné, lors de l'introduction, que les formats et les activités d'analyse de données en génétique et bio-informatique requièrent un niveau de performance qui dépasse leurs capacités. Conséquemment, quelle serait la pile technologique la plus adaptée pour répondre à ces besoins ? De plus, quelles technologies Big Data sont éprouvées dans ce domaine ?

Au laboratoire de recherche AmpLab, de l'université de Berkeley, les chercheurs ont mis au point un format moderne et efficace pour le traitement de données génétique. Ce nouveau format de données génétiques est appelé ADAM (Massie M, 2013 Dec 15). Ce format inclut une librairie comprenant les étapes de traitement pour le séquençage de données génomiques. Ce nouveau format est disponible librement et livré sous une licence Apache 2. Il requiert l'utilisation de deux technologies modernes du domaine du Big Data.

Premièrement, il utilise la technologie Apache Avro qui est un cadriciel d'appels de procédures distantes et de sérialisation de données utilisant le format JSON pour la définition des types de données et des protocoles, et sérialise les données dans un format binaire plus compact et adapté aux mégas données. (D. Cutting, 2011) Deuxièmement, il utilise aussi le format Apache Parquet (D. Vohra, 2016) qui est un format interopérable très efficace pour le stockage de données à l'aide des technologies de bases de données NoSQL (c.-à-d. base de données en colonnes popularisées par le mouvement Big Data) incontournables pour le traitement efficace de mégadonnées.

Le pipeline de ADAM a été testé avec la technologie Spark (Zaharia M, 2016), qui est un cadriciel de traitement distribué. L'utilisation conjointe de ces trois technologies comporte les avantages suivants :

- Avro procure un schéma explicite pour l'accès aux données, et l'extraction peut se faire en C, C++, C#, Java, Scala et Python ;
- Parquet permet l'accès comme à une base de données ;
- Spark améliore la performance par rapport aux technologies antérieures telle que HADOOP en utilisant une cache, et en réduisant les entrées et sorties sur le disque.

La performance de ce pipeline génétique, pour le traitement et l'extraction de données, a démontré qu'il était 100 fois plus rapide que les pipelines conventionnels. Pour les raisons susmentionnées, nous avons donc décidé d'expérimenter avec le format et la technologie émergente d'ADAM pour la conversion du génome de référence HG38 de 1000Genome (1000 Genomes Project Consortium, 2010) en format Parquet.

Les fichiers du génome de référence ont un format VCF (Variant Calling Format), et ADAM utilise un schéma différent pour décrire des séquences génomiques, des génotypes et autres paramètres avec ce type de fichier. Une fois les données traitées avec le format d'ADAM, elles sont sauvegardées avec l'aide du cadriciel Apache Parquet qui utilise un format par colonne.

Nous savons que les données patientes utilisées par les chercheurs contiennent des mutations, ou des données déviant de la norme. Il est donc important d'utiliser un génome de référence humain disponible à des fins de comparaison. Pour la première preuve de concept de ce projet de recherche, le génome de référence HG38 de 1000Genome sera converti au format ADAM afin de tester la performance d'extraction de données génomiques. Ce sera le premier objectif du projet de recherche GNOMEViewer. L'objectif initial est que le visualisateur GNOMEViewer puisse être utilisé avec n'importe quel pipeline de séquençage utilisant ADAM. Ainsi il sera possible de visualiser les données et les résultats qui pourront être téléchargés et sauvegardés dans le dossier des patients, dont les génomes sont en cours

d'analyse. Cette proposition originale de conception est schématisée par les étapes (c.-à-d. le pipeline) présentées à la figure 3.1 ci-après.

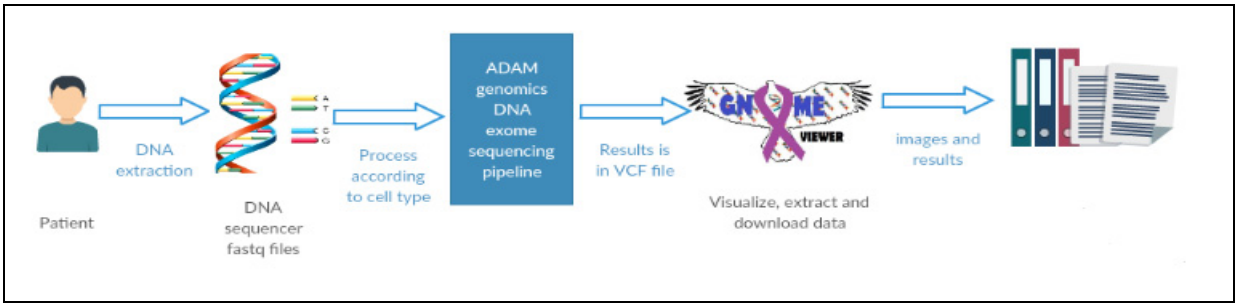

Figure 3.1 Flux de travail d'un pipeline de séquençage

GNOMEViewer serait ainsi intégré au bout du pipeline afin de visualiser les résultats. Tel que décrit la figure 3.1, on peut remarquer que : 1) l'ADN d'un patient sera prélevé; 2) il sera ensuite séquencé; 3) par la suite, le chercheur pourra utiliser ADAM pour traduire le résultat du séquençage (c.-à-d. le fichier en format VCF) en un fichier qui sera lu par GNOMEViewer; 4) d'où il pourra effectuer la visualisation avancée et en tirer des conclusions. La prochaine section présente les technologies nécessaires pour mettre en œuvre GNOMEviewer.

3.2 Utilisation d'EMR sur AWS

Pour exécuter la visualisation de données génétiques à grande échelle, il est nécessaire de sélectionner une infrastructure matérielle facilement extensible. La technologie *Elastic Map Reduce* (EMR) est disponible sur la plateforme offerte par AWS (Amazon Web Services) (Cloud, 2011) qui fournit une infrastructure Hadoop permettant le traitement de grandes quantités de données sur des instances EC2 (Elastic Compute Cloud) et ce dynamiquement (c.-à-d. vous avez la possibilité d'agrandir ou réduire la puissance à votre guise). Cette plateforme est flexible et, il est aussi possible d'utiliser d'autres cadriciels courants tels qu'Apache Spark, HBase (HBase A., 2016), Presto (GUIDE, 2015) et Flink (Carbone P, 2015) et d'interagir avec d'autres instances d'AWS telles que Amazon S3 ou Amazon DynamoDB pour faciliter la gestion de cette infrastructure matérielle sur demande.

Dans le cadre de cette recherche, puisque ADAM a été testé par Berkeley avec la technologie Spark et non Hadoop et que la conversion du génome de référence va engendrer des fichiers Parquet, nous avons choisi d'expérimenter avec la technologie AWS EMR avec des instances S3 pour le stockage des fichiers Parquet. Un schéma de l'infrastructure conçue pour l'expérimentation de GNOMEViewer est représenté à la figure 3.2.

Lorsqu'EMR est offert avec Spark préinstallé, la technologie YARN (Vavilapalli VK) est préinstallée sur les instances AWS à titre de gestionnaire de la grappe de calculs qui rassemble les ressources dans un seul conteneur. Ce dernier est simplement un ensemble de ressources (dans notre cas l'ensemble des instances EC2) qui sont assemblées et colocalisées pour un traitement efficace. De plus, une fonctionnalité intéressante de YARN, sur EMR, est que l'allocation dynamique de ressources est offerte par défaut. Donc, il n'y a pas de nécessité de sélectionner le nombre d'exécuteurs (c.-à-d. d'instances) sur un nœud, car YARN met à l'échelle automatiquement le nombre d'exécuteurs sur les grappes de calculs EC2 qui sont disponibles.

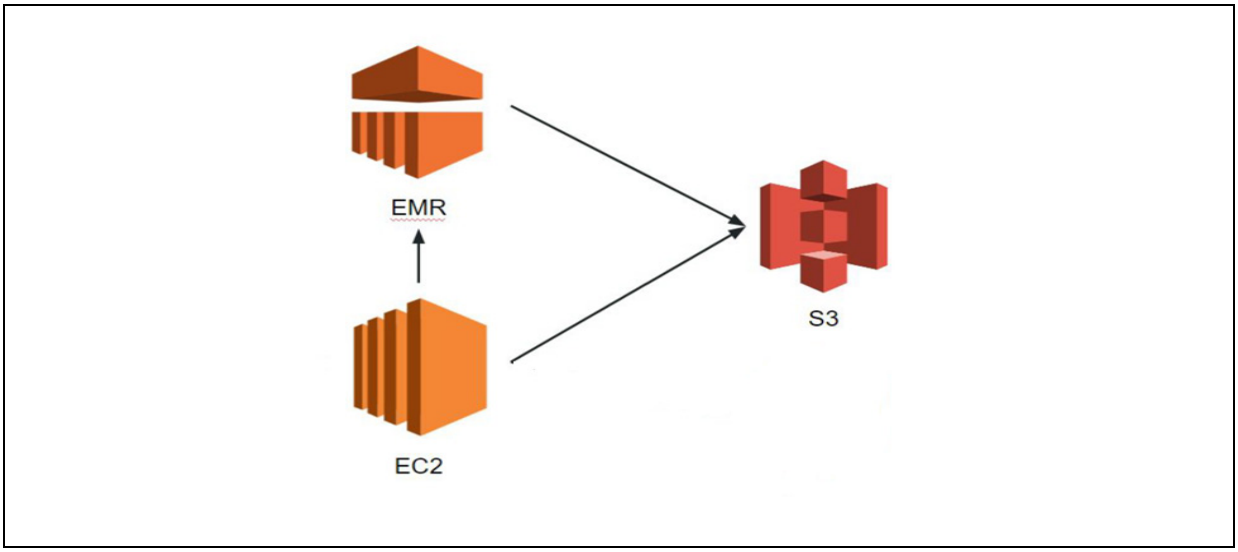

Figure 3.2 Infrastructure AWS utilisée pour la conversion vers le format ADAM

Afin d'effectuer la conversion des fichiers de format VCF vers le génome de référence, et ce d'une manière efficace, nous avons expérimenté avec la fonction « adam2vcf » qui prend un

fichier VCF en entrée et produit un dossier pour l'écriture du fichier Parquet en sortie. La commande utilisée pour exécuter cette conversion est :

adam-submit vcf2adam fichier_vcf dossier_sortie_hadoop

Il est à noter cependant que cinq étapes sont nécessaires pour la conversion (voir la figure 3-3).

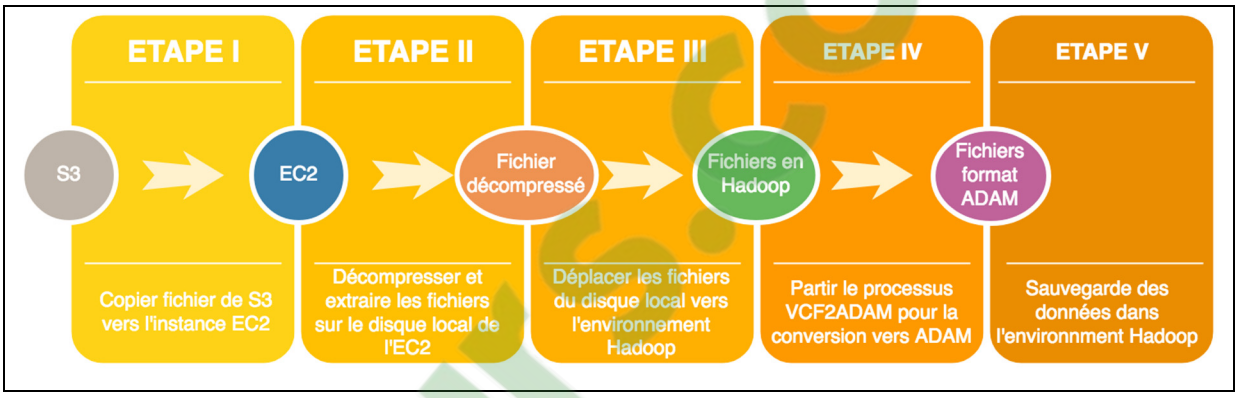

Figure 3.3 Étapes de conversion des fichiers VCF vers le format ADAM, pour obtenir les fichiers Parquet. Image inspirée des étapes prise du document GenomeViewerBack-end en annexe II

Les fichiers, au format VCF, qui contiennent les données du génome de référence HG38 ont été téléchargés à l'aide d'un format compressé et ont été sauvegardés sur une instance AWS de type S3 au préalable. À partir de là, les étapes de conversion présentées à la figure 3.3 ont été effectuées :

1)Copier le fichier VCF compressé dans le dossier Hadoop d'EMR ;
2)Décompresser ce dernier ;
3)Mettre le fichier compressé en Hadoop, car il s'agit d'un prérequis d'ADAM, que tout fichier donné en entrée avec une fonction ADAM doit être en Hadoop ;
4)Convertir les fichiers en parquet avec la commande de conversion « adam2vcf » ;
5)Sauvegarder dans les instances S3 afin de pouvoir les réutiliser par la suite puisque les instances EMR ne conservent pas de fichiers.

Ces étapes ont permis la conversion d'un fichier de format VCF vers le format Parquet, puis la sauvegarde dans une instance S3. Pour voir les détails des étapes complètes du processus de conversion, veuillez consulter l'Annexe II – GenomeViewer_back-end.

3.3 Compte rendu de la conversion

Pour la conversion de tous les chromosomes humains du génome de référence HG38 de 1000Genome, nous avons fait face à plusieurs difficultés et documenté plusieurs observations. Avant tout, traçons un peu le portrait de la situation, à l'aide de la figure 3-4 et du tableau 3-1 suivant. Les interprétations de cette figure seront données dans cette section.

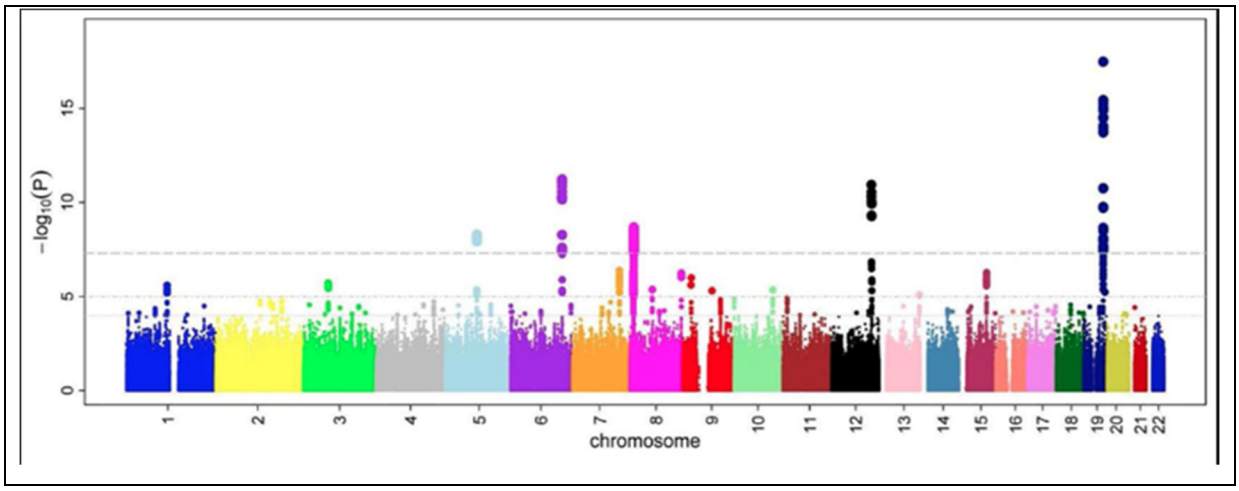

Figure 3.4 Manhattan plot utilisé pour montrer la taille des chromosomes les uns par rapport aux autres. Tirée de Gibson Greg, 2010

Tel que décrit à la figure 3.4, on peut constater que les chromosomes humains ne sont pas de la même taille. Le test d'association qui a permis d'obtenir cette image se nomme test de *Genome Wide Association Study (GWAS)*, et s'effectue typiquement à la grandeur du génome, d'où la possibilité de constater la taille des chromosomes.

Tableau 3.1 Résumé des particularités des fichiers VCF du génome de référence humaine HG38 et types d'instances AWS utilisées pour la conversion au format ADAM, avec le temps

Chromosome	Taille compressée	Taille non compressée	Temps de conversion	Instance AWS
1	1200 MB ~ 1.2 GB	61 GB	6 h	R4
2	1300 MB ~ 1.3 GB	67 GB	6h 40 min	R4
3	1100 MB ~ 1.1 GB	56 GB	4h 33 min	R4
4	1100 MB ~ 1.1 GB	55 GB	4h 20 min	R4
5	1000 MB ~ 1.0 GB	50 GB	3h 55 min	M4
6	1000 MB ~ 1.0 GB	48 GB	3h 20 min	M4
7	921 MB	45 GB	2h 50 min	M4
8	876 MB	44 GB	2h 54 min	M4
9	681 MB	34 GB	2h 8 min	M4
10	784 MB	38 GB	2h 18 min	M4
11	779 MB	38 GB	2h 18 min	M4
12	733 MB	36 GB	2h 14 min	M4
13	565 MB	27 GB	2h	M4
14	514 MB	25 GB	1h 56 min	M4
15	504 MB	23 BG	1h 53 min	M4
16	504 MB	26 GB	1h 57	M4
17	441 MB	22 GB	1h 52 min	M4
18	443 MB	22 GB	1h 48 min	M4
19	364 MB	19 GB	1h 17 min	M4
20	347 MB	17 GB	1h 17 min	M4
21	221 MB	11 Gb	1h 1 min	M4
22	217 MB	10 GB	45 min	M4
Total	5 594 MB ~ 15.6 GB	757 GB		

Les données de la figure 3.4 et celles du tableau 3.1 témoignent aussi de la différence de taille des chromosomes. Lors de cette expérimentation, ce facteur a eu une incidence directe sur le

temps de traitement, la taille de données à traiter et le type d'instance AWS qu'on a dû sélectionner pour la conversion de données vers le format ADAM.

3.3.1 Observations liées à la taille des fichiers VCF une fois décompressé

Les étapes de conversion décrites à la figure 3.3 visent, entre autres, à décompresser un fichier de format VCF. Pour ce faire, il faut d'abord le copier sur le disque des instances EC2, car il n'est pas possible de décompresser un fichier directement sur les instances S3 d'AWS (Amazon Web Services, Inc. or its affiliates, , 2013). L'étape suivant la décompression vise à copier le fichier au format VCF en Hadoop, car toutes les fonctions de ADAM requièrent que les fichiers donnés en entrant soient dans cet environnement. La taille du disque local, lors de notre preuve de concept, était de 250Gb.

3.3.2 Observations liées à la conversion des fichiers VCF vers ADAM

Pour la conversion des chromosomes 18 à 22, une mémoire vive de 32 GB était suffisante ; par contre pour les chromosomes suivants, la mémoire vive a dû être augmentée à 150 GB.

De plus pour la conversion des chromosomes 1 à 4, tel que décrit au tableau 3.1, les instances M4 (c.-à-d. 10 instances) ne suffisaient pas et il y avait des pertes de nœuds lors du processus de conversion dès les premières secondes. Ces instances ont été conçues pour l'exécution de tâches générales et offrent un ensemble équilibré de ressources de calcul, de mémoire et de puissance de réseau. Elles sont utilisées, en général, pour les tâches de traitement de données qui nécessitent des capacités de mémoires supplémentaires, et pour les grappes de calculs. Néanmoins pour la conversion de données vers Parquet, ces dernières n'étaient pas adaptées, donc des instances R4 ont dû été utilisées pour la conversion de ces chromosomes. Les instances R4 de AWS sont optimisées pour la mémoire et peuvent être utilisées pour « cacher » de grandes quantités de données en mémoire et faire du traitement de données en parallèle. La conversion a pris 6h et 40 minutes pour traiter le plus gros chromosome.

À la suite de ces difficultés, quelques tests supplémentaires ont été réalisés. La première conversion devait prendre 30 min avec des instances M3 (3 nœuds) qui sont pour des applications d'usage générales (Amazon Web Services, Inc. or its affiliates, , 2013). À la suite de plusieurs tests, nous n'avions pas réussi à répéter cette conversion en 30 min. C'est pourquoi nous avons réalisé d'autres tests en modifiant les paramètres et options (en ligne de commande) tels que présenté à la figure 3.5.

```
adam-submit
        \--master yarn-client
        \--num-executors 1024
        \--executor-memory 8g
        \--executor-cores 1
        \--driver-memory 14g
        \--conf spark.yarn.executor.memoryOverhead=5000
        \--conf spark.default.parallelism=132
        \-- vcf2adam -parquet_compression_codec SNAPPY
        \-coalesce 7
        \/user/hadoop/HG38/ALL.chr1.phase3_shapeit2_mvncall_integrated_v3plus_nounphased.rsID.genotypes.GRCh38_dbSNP_no_SVs.vcf
        \/user/hadoop/Chromosome1/
```

Figure 3.5 Commande améliorée pour la conversion de fichiers VCF vers ADAM en utilisant la fonction « vcf2adam »

Les explications concernant les options utilisées pour la commande décrite à la figure 3.5 sont les suivantes :

1) **master yarn-client** → permet de sélectionner le gestionnaire de tâche YARN préinstallé sur l'instance EMR d'AWS ;
2) **num-executors 1024** →nombres d'exécuteurs pour la tâche ;
3) **executor-memory 8g** → mémoire allouée pour chaque exécuteur ;
4) **executor-cores 1** → nombre de microprocesseurs ;
5) **driver-memory 14g** → quantité de mémoire pour le « driver » ;
6) **conf spark.yarn.executor.memoryOverhead=5000** → quantité de mémoire off-heap ;

7) **conf spark.default.parallelism=132** → nombre de partitions par défaut dans les RDD de Spark quand retournés par des commandes telles que Join, ReduceByKey et Parallelize quand utilisé ;
8) **vcf2adam –parquet_compression_codec SNAPPY** → commande ADAM avec type de compression SNAPPY (facilite compression et décompression rapide de données lors de l'extraction) ;
9) **coalesce 7** → permet de réduire le nombre de partitions dans les DataFrame Spark.

À l'aide de ces optimisations, et en utilisant trois instances M4, de 32 Gb de RAM, il a été possible de convertir le chromosome 22 en 19 minutes. Il faut cependant noter que Spark utilisera uniquement le nombre d'exécuteurs requis pour le travail, car, comme stipulé dans une section précédente, l'allocation dynamique est activée sur les instances EMR d'AWS. Toutefois, lorsque cette combinaison d'option n'est pas mentionnée, la conversion prend une heure et n'utilise pas toutes les ressources. Nous pensons qu'il y aurait quelques modifications à faire dans les fichiers de configuration de YARN du côté d'AWS afin de corriger cette différence.

3.4 Création du module « matching » dans ADAM

Afin de pouvoir rechercher le ou les biomarqueurs correspondant aux informations passées en paramètres de l'interface, il a été nécessaire de rajouter la fonctionnalité dans ADAM, car elle était inexistante. Il a fallu localiser le dossier « adam-cli » qui permet de faire des requêtes en utilisant le format ADAM, et ajouter une fonction « *matching* » qui fait la correspondance entre le ou les biomarqueurs passés en paramètre et le bon fichier Parquet contenant les informations. La fonction ajoutée dans « adam-cli » est la suivante :

```
1  package org.bdgenomics.adam.cli
2
3  import org.apache.spark.SparkContext
4  import org.apache.spark.sql.SQLContext
5  import org.apache.spark.sql.SparkSession
6  import org.bdgenomics.adam.rdd.ADAMContext._
7  import org.bdgenomics.utils.cli._
8  import org.bdgenomics.utils.misc.Logging
9  import org.kohsuke.args4j.{ Argument, Option => Args4jOption }
10
11 object Matching extends BDGCommandCompanion {
12   val commandName = "matching"
13   val commandDescription = "Finds all variants with a given rsid"
14
15   def apply(cmdLine: Array[String]) = {
16     new Matching(Args4j[MatchingArgs](cmdLine))
17   }
18 }
19
20 class MatchingArgs extends Args4jBase with ParquetArgs {
21   @Argument(required = true, metaVar = "INPUT", usage = "The ADAM or FASTA file to match variants from", index = 0)
22   var inputPath: String = null
23   @Argument(required = true, metaVar = "OUTPUT", usage = "The path to save result to", index = 1)
24   var outputPath: String = null
25   @Argument(required = true, metaVar = "RSID", usage = "The rsids that you want to match against the database", index = 2)
26   var rsid: List[String] = null
27 }
28
29 class Matching(protected val args: MatchingArgs) extends BDGSparkCommand[MatchingArgs] with Logging {
30   val companion = Matching
31
32
33   def run(sc: SparkContext) {
34
35       val sqlContext = new SQLContext(sc)
36
37       val genotypes = sqlContext.read.parquet(args.inputPath).filter(col(namecolumn).isin(args.rsid))
38
39       genotuypes.saveAsTextFile(args.outputPath)
```

Figure 3.6 Image de la fonction « matching » ajoutée à « adam-cli »

Cette nouvelle fonction « *matching* » se termine par la création d'un fichier qui sera lu par l'application Django. Si on tient compte du fait que les RDD en Spark sont immuables, et que la création de chacun d'entre eux entraine la création et le « shuffling » de données en mémoire, il fallait garder ce mouvement de données au minimum, en filtrant pour le rsID recherchés (voir le code de la figure 3.9). Cela a été possible grâce à la fonction « *isin »,* localisée à la ligne 37, qui reçoit en paramètre une liste de données de type « String » contenant soit un seul rsID pour la recherche d'une seule mutation, soit la liste de toutes les mutations contenues dans le fichier .csv du chercheur.

Cette conception permet à Spark de pointer uniquement vers la partition contenant les rsID pour lesquels on lui demande de filtrer. Ce processus est illustré à la figure suivante :

Figure 3.7 Comportements de Spark lors de la lecture de fichier

La partie de droite de la figure 3.7 décrit que, quand on filtre lors de la lecture de fichier, Spark pointera uniquement vers la partition contenant le rsID recherché au lieu de pointer toutes les partitions (partie gauche de la figure), ce qui permettra de sauver du temps. Le temps final pour la recherche de rsID de la fonction « *matching* » à l'aide de cette approche de conception a permis d'atteindre une performance de moins d'une seconde. (~1 seconde) pour un ou pour jusqu'à 50 000 rsID**.**

3.5 Connexion de l'interface avec Spark pour la recherche de mutations dans ADAM et mesure de la performance

Le processus permettant l'affichage final du profil génétique d'un patient est illustré de manière simple à la figure 3.8 :

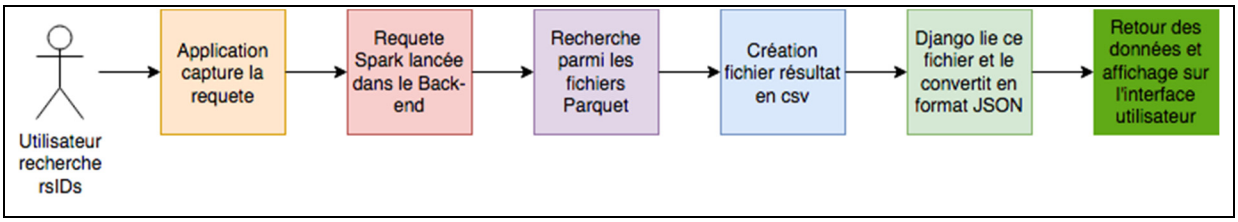

Figure 3.8 Processus d'affichage du profil génétique d'un patient dans l'application GNOMEViewer

Les étapes suivantes sont exécutées:

1) Le chercheur (ou un médecin) lance une recherche de mutation comprenant un seul rsID ou un fichier contenant le profil du patient;
2) L'application capture la requête GET (ou POST) asynchrone à l'aide de l'URL utilisé par le programme Django qui fait parvenir les données à la méthode correspondante;
3) La méthode Django lance un processus Spark (qui démarre des tâches en parallèle) pour la recherche rapide des mutations dans ADAM Parquet;
4) La création d'un fichier au format .csv est effectuée en moins de ~1 seconde et qui est lue par la méthode « *matching* » de Django;

```
def matching(rsid,chromosome,position):
    call(['rm', '-r','/Users/beatrizkanzki/Documents/GenomeViewer/result'])
    return call(['adam-submit','matching','/Users/beatrizkanzki/Documents/GenomeViewer/chr22/chr22-adam','/Users/beatrizkanzki/Documents/GenomeViewer/result',rsid])
```

Figure 3.9 Fonction « matching » dans Django qui lance le processus Spark et prend en paramètre les fichiers en entrés, le dossier de résultats, et la liste des rsID à rechercher

5) La lecture du fichier .csv est effectué à même la requête HTTP, et déclenche l'évènement qui modifiera l'application par le composant de base React. Et c'est ce dernier qui gère et utilise les données pour mettre à jour l'interface utilisateur de GNOMEViewer;
6) Le programme Django capte par la suite la requête HTTP et renvoie le fichier résultant sous forme de JSON;

7) Finalement l'affichage du graphe, décrit à la figure 2.4, est effectué à l'aide des résultats contenus dans le fichier csv. Dans le cas où il y a une seule mutation, il y aura juste un point affiché sur le graphe résultant.

```
data = urllib2.urlopen("http://ec2-52-35-68-107.us-west-2.compute.amazonaws.com:8000/matching?" + params).read()
print "open url"
rsidArray = []
rsidArray.append(json.loads(['rsid'] for o in data))

chrBoundaries = getChromosomeBoundaries()

rsidArray = json.dumps(rsidArray, ensure_ascii=False, encoding="utf-8").replace("\\", "")

response = json.dumps({
        'data' : {
            'jsonChrBoundaries' : chrBoundaries,
            'jsonValidRsids' : rsidArray
        }
    },
    sort_keys=True,
    indent=4,
    separators=(',', ': ')
)

return HttpResponse(response)
```

Figure 3.10 Fonction Django lançant et recevant la réponse HTTP de l'application Django lors de la recherche de rsID. Comme le démontre le code, la réponse est renvoyée sous forme de JSON à l'interface

La performance enregistrée lors de l'exécution de ce processus a été observée à environ 2 secondes pour la recherche d'une seule mutation et d'environ 3 secondes pour jusqu'à 50 000 mutations observées.

3.6 Conclusion

Tel que mentionné dans ce chapitre, la conversion et la migration des données vers le format ADAM a été possible grâce à l'expérimentation de l'environnement Big Data distribué EMR sur AWS (voir détails à l'Annexe II). À l'aide de cette infrastructure, il a été possible de déployer ADAM et d'effectuer la conversion des fichiers du génome de référence vers le format ADAM en format Parquet avec la fonction *vcf2adam*.

À l'aide d'EMR, il a aussi été possible d'explorer les paramètres optimaux pour la conversion des fichiers en utilisant la technologie de traitement parallèle en mémoire Spark.

À la suite de la conversion des fichiers en Parquet avec ADAM, le prototype expérimental GNOMEViewer avec Django a été testé. La vitesse d'extraction des données à l'aide de Spark a démontré qu'un cadriciel distribué permettait d'avoir un extrait rapide de données vers l'interface et ce, malgré la taille massive des données traitées. L'attente maximale observée, au niveau de la performance, a été de 3 secondes.

CHAPITRE 4

INTERPRÉTATION DES RÉSULTATS ET DIRECTION FUTURE

4.1 Résultats obtenus par rapport aux objectifs fixés

Au chapitre 2, nous avons décrit une conception d'interface utilisateur qui permet d'afficher le profil génétique complet d'un patient dans un délai raisonnable. Le type de visualisation choisi, pour cette preuve de concept, a été l'affichage à l'aide d'une représentation d'un caryotype humain. Est-ce que cette représentation est suffisante pour l'objectif des chercheurs en génétique?

Tel que décrit au chapitre 1, les outils actuellement disponibles aux chercheurs, tels que IGV, LocusZoom, et UCSC Genome Browser, permettent seulement de visualiser une mutation à la fois. Donc, ils affichent une seule position dans le génome et ce en moins de 5 secondes. L'approche utilisée pour la preuve de concept de GNOMEViewer permet d'afficher 30 000 à 50 000 mutations instantanément, et ce à l'échelle du génome au complet, en moins de 5 secondes. Ainsi, avec cette nouvelle approche, la recherche des variants contenus dans le profil du patient a été grandement améliorée. Ceci a été possible grâce à l'utilisation de technologies modernes Big Data, telle que Spark conjointement avec le nouveau format génétique proposé par l'Université Berkeley (ADAM).

Les résultats de cette recherche ont fait l'objet d'une démonstration et d'une présentation au « *PhD Track* » de la conférence « *Digital Health* » en 2017 à Londres. Aux sections 3.2, et 3.3, nous avons décrit qu'il était possible d'afficher la figure 2.4 en moins de 5 secondes. À la suite de démonstrations lors de cette conférence, nous avons observé que ces résultats ont surpris agréablement les participants présents à la conférence.

Le type de représentation graphique du profil génétique des patients à l'aide de la représentation d'un caryotype humain répondait non seulement aux besoins des chercheurs,

présents à la conférence, mais offre aussi la possibilité d'afficher entre 30 000 et 50 000 mutations en même temps. Il s'agit d'un avancement important par rapport aux autres outils disponibles.

Il faut aussi rappeler que les chercheurs, jusqu'ici, n'exploraient jamais toutes les mutations détectées dans le profil des patients. Ils exploraient les plus communes, car, tel que mentionné à la section 1.2, il leur aurait fallu entre 13 heures et 11 jours pour toutes les examiner avec les logiciels disponibles aujourd'hui. Ce qui est irréaliste! Maintenant avec GNOMEViewer, le temps d'exploration est réduit à moins de 5 secondes, et cette fois pour le génome au complet.

Pour reprendre l'expression d'une des juges : « *This is huge!* ». Et, un industriel, qui participait à la conférence, a même insisté et dit « *That's exactly what I need, can I contact you, as I would like to discuss further* » (Craig Carty, Chef exécutif de « *The relevance Network* »).

Il est donc possible de constater que tant au niveau de la performance que visuellement, GNOMEViewer répond aux besoins de visualisation de mutations somatiques pour les chercheurs en oncologie. Mais qu'en est-il de la convivialité de ce choix de représentation de l'interface utilisateur? Le caryotype humain affiché est entièrement interactif et permet aux chercheurs de repérer rapidement les grappes de mutations présentes dans le dossier du patient, comme le montre la figure 2.4. De plus, les nouvelles fonctionnalités présentent sur cette interface, permettent de détecter les patrons mutagéniques liés à des maladies et qui correspondent au profil génétique du patient.

Cette information est cruciale pour les médecins chercheurs en oncologie qui pourront désormais voir à quoi correspond le profil de leurs patients. Cela pourra même permettre de faire de la prévention, dans le cas où le patient ne démontrerait pas encore des signes de la maladie, mais que son profil génétique affiche un patron semblable à une maladie répertoriée.

La performance de GNOMEViewer pour l'affichage des profils génétique le positionne comme outil de choix, autant pour les laboratoires de recherche, que dans les cliniques, et les hôpitaux qui offrent des services de consultation en génétique.

GNOMEViewer pourrait permettre de faire avancer le domaine de la médecine personnalisée, et particulièrement celui de la médecine de prévention en permettant d'afficher le profil génétique des patients dans un délai rapide. Sans négliger le fait que maintenant, grâce à cette interface utilisateur, il est possible de comparer le profil du patient interactivement avec des maladies dont on connaît déjà les patrons d'expression à l'échelle du génome au complet.

4.2 Directions futures

Le chapitre 2 décrit l'enjeu principal qu'il fallait trouver une manière d'afficher le profil génétique au complet d'un patient dans un délai raisonnable. Compte tenu des résultats encourageants de cette preuve de concept, ce prototype logiciel pourrait encore être amélioré. Lors de recherches subséquentes, il serait intéressant d'investiguer les quatre améliorations suivantes :

1) *GenomeCrawler* : l'ajout d'une fonctionnalité de visualisation qui permettrait de voir le taux de recombinaison au niveau du génome autour des mutations détectées, et surtout offrir plus de détails sur le type et le nom du gène affecté. Ces informations sont cruciales, car une recombinaison entre séquences d'ADN, peut ne pas avoir de séquence homologue ce qui cause une translocation chromosomique (c.-à-d. un échange d'une séquence d'ADN entre deux chromosomes différents) qui peut causer un cancer (Andersen SL, 2010). L'interface utilisateur de visualisation future serait semblable à celle proposée à la figure 4.1;

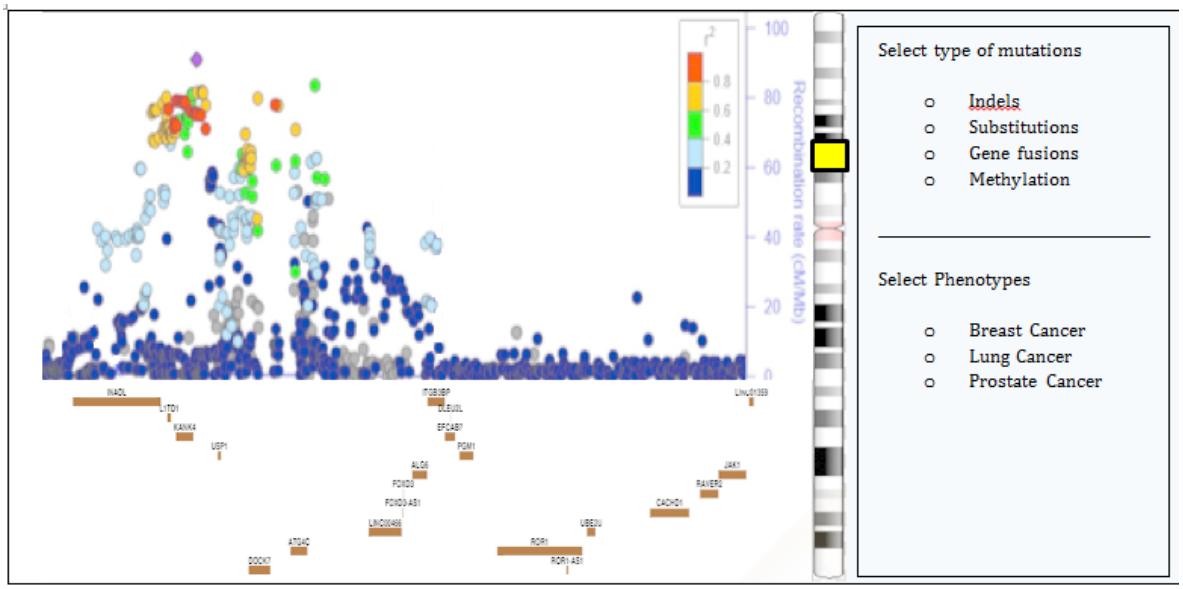

Figure 4.1 Maquette du futur module GenomeCrawler qui permettrait de voir le taux de recombinaison tout en faisant glisser la case jaune sur le chromosome. L'image s'ajusterait à la région visualisée

2) *Overlapping loci* : l'ajout d'une fonctionnalité de visualisation qui permettrait de voir toutes les mutations qui affectent plusieurs gènes en même temps;
3) *GeneticPattern* : l'ajout d'une fonctionnalité de visualisation qui permettrait de prédire le patron d'expression d'un cancer spécifique. Elle permettrait de comparer le profil génétique d'un patient aux profils d'expressions déjà connus, dans la communauté, pour une maladie donnée; et finalement;
4) L'ajout d'une fonctionnalité de visualisation LD Graph qui permettrait de voir les liens entre les mutations et les variants voisins, afin de faire ressortir l'endroit où l'expression génique est la plus forte.

Pour la conception de ces nouvelles fonctionnalités de visualisation, il faudra intégrer d'autres bases de données publiques. Le but de ces améliorations est de faire une deuxième version du prototype expérimental GNOMEViewer qui pourrait devenir un logiciel libre incontournable en oncogénétique.

4.3 Utilisation de ADAM

L'utilisation du format ADAM a permis d'améliorer significativement le temps de recherche de variants génétiques pour le chercheur. Ce format proposé et expérimenté par l'Université Berkeley, indexe les variants d'abord à l'aide du format Avro, avant de les convertir en format Parquet qui permet une recherche rapide avec le nom du variant et ce à l'aide d'une puissante grappe de calcul distribué. Pour la preuve de concept de cette recherche, l'utilisation d'EMR d'AWS a été expérimentée pour ce projet. Dans une seconde recherche, il serait possible de mieux planifier et optimiser l'étape de conversion des fichiers du génome de référence à l'aide du format et de la technologie d'ADAM. Par exemple, nous avons découvert que la conversion de fichiers vers le format Apache Parquet requiert beaucoup de mémoire et de ressources (Oliveros, 2016), tel que présenté au tableau 3.1. Il serait donc plus avantageux d'utiliser ces instances dès le début des expérimentations. Il est important, pour le futur de cette recherche, de considérer le coût d'utilisation, la taille des fichiers, le type d'instances et la quantité de mémoire à utiliser pour guider l'utilisation potentielle du prototype.

Pour aider le chercheur (ou le médecin) qui n'a que peu de connaissance en informatique ou en génie logiciel, il serait important d'automatiser le déploiement de plus d'une configuration matérielle de GenomeViewer selon : 1) la performance requise ; et 2) le budget disponible. Une solution possible serait de créer un script Terraform (Brikman, 2017) encapsulé dans un conteneur de type Docker (Merkel, 2014) qui déploierait l'infrastructure choisie par le chercheur et ferait la conversion de données automatiquement pour lui lors de l'installation de GNOMEViewer.

Une autre amélioration possible se situa au niveau de l'utilisation de la fonction « *auto-scale* » d'EMR qui pourrait pu être activée afin d'accélérer le processus de conversion des données dans le but de diminuer le temps d'attente qui est actuellement d'environ 4 heures. Cette fonction disponible chez AWS permet d'augmenter le nombre d'instances (c.-à-d. serveurs) nécessaires lors de la conversion et permettrait d'ajuster la quantité d'instances nécessaires par rapport à la taille des données à convertir.

Finalement, une autre amélioration possible vise le génome de référence qui est typiquement actualisé tous les 5 ans. L'expérimentation lors de cette recherche a utilisé le HG38 qui est sorti en avril 2013. Lors de l'écriture de ce rapport, en octobre 2018, tout porte à croire que sa mise à jour est imminente et qu'il faudra répéter cette expérience bientôt. Il serait donc intéressant d'avoir une fonctionnalité de mise à jour du génome de référence dans GNOMEViewer.

4.4 Itérations futures pour la complétion du projet et aperçu de l'architecture finale

Tel que discuté à la section précédente, pour une prochaine itération d'amélioration du prototype GNOMEViewer, il serait intéressant d'ajouter d'autres modules de visualisation. Pour ce faire il faudra importer l'information de bases de données publiques tel que décrit à la figure 2.3. Certaines de ces bases de données publiques peuvent être intégrées sans problèmes. Cependant, les développeurs de la base de données TCGA ont inclus une clause empêchant d'intégrer cette base de données dans un projet logiciel que ce soit un logiciel libre ou pas. Pour ce cas particulier, il faudra trouver une manière de l'intégrer.

Aussi, tel qu'abordé à la section 3.3, l'interrogation des fichiers Parquet contenant le génome de référence se fait présentement à l'aide de deux processus :

1) L'application Django lance un sous-processus appelant ADAM pour la recherche de biomarqueurs et créera un fichier contenant le résultat ;
2) L'application Django liera ce fichier et retournera la réponse à l'interface sous forme de JSON.

Il serait possible d'interroger les fichiers Parquet, de manière distribuée, sans initier deux processus. Une autre solution possible serait d'utiliser la technologie de base de données Cassandra (Cassandra, 2015), une base de données NoSQL, qui permet l'extraction de données de manière distribuée sur une grappe de calcul Hadoop. Inspirée de l'architecture d'Amazon Dynamo (DeCandia G, 2007) et du modèle de donnée BigTable (Chang F, 2008) de Google, cette base de données distribuée est reconnue pour sa capacité d'évolution avec la taille

grandissante des données. Cette dernière est aussi reconnue pour avoir une grande disponibilité, une tolérance à la perte de données, et on peut aussi la paramétrer si on désire une réponse consistante comme dans le cas d'application en temps réel. Cette dernière a déjà été évaluée pour l'extraction, et l'insertion de données génomiques. Les résultats d'expérimentation ont confirmé que la technologie de bases de données Cassandra démontrait une plus grande performance pour l'extraction de données génomiques non structurées. Ce qui signifie que le type de nœuds, utilisés pour enregistrer les données et leurs nombres, influence grandement la vitesse d'extraction. Ainsi il serait possible d'insérer les données Parquet en utilisant Spark, en sélectionnant le chromosome comme clé primaire, et le biomarqueur comme partition. Cela permettrait de lire et d'extraire les données en utilisant une complexité O (1).

L'intention de l'équipe de recherche est que GNOMEViewer reste et demeure gratuit sous une License de logiciel libre. L'installation de Cassandra sur le serveur privé des chercheurs serait aussi possible, cependant, il serait nécessaire d'expérimenter ce choix technologique pour les limitations suivantes :

- L'évolution de Cassandra et sa vitesse d'extraction sont limitées par le nombre de nœuds que le chercheur peut allouer à son propre serveur;
- L'installation complète de Cassandra, sa maintenance et l'évolution de la base de données serait à la charge des employés de soutien du chercheur sur le serveur privé, comme sur AWS.

Compte tenu de ces limitations, il serait aussi possible de considérer l'utilisation d'autres technologies plus spécialisées :

- DynamoDB (Sivasubramanian S., 2012): est une base de données distribuée par AWS dont la maintenance incombe à AWS. Pour sélectionner cette option, il est important que le chercheur comprenne qu'une fois ses données migrées vers DynamoDB, son fournisseur de solutions infonuagique sera AWS uniquement et qu'il devra payer pour ses services à l'usage;

- DataStax (United States patent application Patent No. US 14/856,001., 2017): la version payante de Cassandra, pour laquelle il est possible d'avoir du support (qui est effectué conjointement par AWS, DataStax et l'équipe de soutien du chercheur). Le plus grand bénéfice de l'utilisation de cette technologie est que cela permet au chercheur de choisir son fournisseur infonuagique.

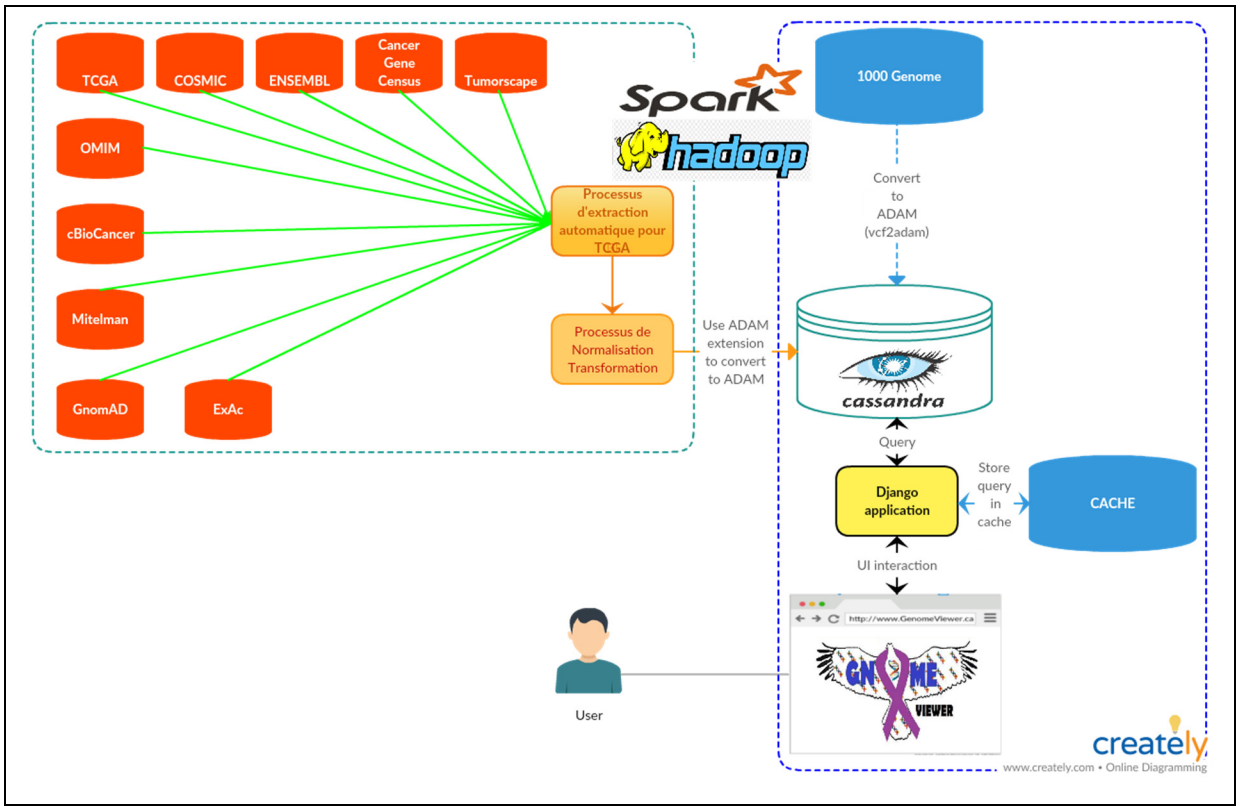

Figure 4.2 Infrastructure finale de GNOMEViewer

À la figure 4.2, on remarque que l'installation de GNOMEViewer nécessite actuellement deux processus :

1) Un processus pour l'extraction des données phénotypiques se retrouvant dans les bases de données publiques qui apparaissent en rouge. Cela implique aussi la transformation et l'insertion des données en format ADAM dans Cassandra. Il s'agirait d'un processus qui se ferait uniquement lors de la première installation;

2) Un deuxième processus pour le déploiement automatique du pipeline complet contenant le site Web, développé et affiché à la section 3.2, la cache, et les « buckets » S3 contenant les données de Cassandra sauvegardées. Pour cette preuve de concept, la conversion du génome de référence vers le format ADAM et l'interrogation de ce dernier, à l'aide de la technologie Spark, est expérimentée (voir chapitre III), mais, tel que présenté ci-haut, une meilleure approche de solution pourrait être expérimentée.

Ainsi lors d'une expérimentation future, il serait possible de faire le déploiement automatique des deux processus avec les outils suivants :

- Docker + Terraform + S3;
- ECR (Patent No. U.S. Patent Application 10/002,247., 2018) + Terraform + S3.

Docker est un conteneur logiciel léger, autonome et exécutable qui inclut tout ce dont on a besoin pour exécuter une application : code, pipeline, outils système, bibliothèques système et paramètres. ECR (*Elastic Container Registry*) est un service de conteneur similaire à Docker, mais sur AWS. Terraform, est aussi une alternative technologie intéressante. Terraform est un cadriciel qui permet d'effectuer de l'« *infrastructure as code (IAC) ».* Cette technologie de scriptage permet de créer, changer, déployer et améliorer une infrastructure rapidement, car il est seulement nécessaire de modifier et exécuter le script lors de changements (c.-à-d. la technologie « infrastructure as code).

La solution impliquant Docker permettrait de garder une copie du pipeline sur place dans le laboratoire du chercheur. L'approche impliquant ECR permettrait d'enregistrer une copie sur AWS. Le choix entre ces deux solutions alternatives est une question de préférence, car, dans les deux cas, le chercheur ne paiera que : 1) pour le stockage des données sur AWS S3 ; et 2) les instances EC2 déployées, soit le coût de l'infrastructure matérielle.

Revenant au sujet de la figure 4.1, on voit que les données du génome de référence, converti en ADAM, seront aussi stockées dans la base de données Cassandra.

Cette solution permet aussi la présence d'une cache qui sera utilisée afin de sauvegarder des recherches récurrentes ou le futur fichier .vcf contenant les 30 000 à 50 000 mutations génétiques issues de l'*exome sequencing* pour l'affichage sur l'interface. Cela devrait permettre une meilleure performance de recherche et d'affichage ; soit une meilleure expérience pour l'utilisateur de l'interface.

4.5 Conclusion

La conception et l'expérimentation du prototype expérimental GNOMEViewer ont permis d'explorer plusieurs technologies telles que ADAM, AWS, et Spark pour améliorer la visualisation de données génétiques. Il s'agit ici d'une innovation importante par rapport aux solutions disponibles aujourd'hui. Cette recherche a permis de mettre en lumière les particularités liées à chacune de ces technologies tant au niveau de la mémoire utilisée, que des ressources utilisées lors de l'expérimentation.

Tel qu'abordé au chapitre 4, il y a encore beaucoup d'avenues de recherche possible dans ce domaine, et ce, non seulement pour améliorer cette deuxième preuve de concept GNOMEViewer lui-même, mais aussi pour faciliter son utilisation future à l'aide de pratiques exemplaires DevOps (M. Httermann, 2012). Plusieurs améliorations possibles ont été présentées telles que l'automatisation du déploiement des différentes parties de l'application à l'aide des technologies Docker, Terraform et S3.

CONCLUSION

La présentation de l'interface de GNOMEViewer, les tests de performance effectués pour son affichage, et les commentaires recueillis lors de la conférence Digital Health à Londres en 2017 permettent rapidement de conclure que GNOMEViewer permet une avancée dans le domaine de la visualisation génomique qui n'était pas possible avant.

Nous sommes conscients qu'il faudra rajouter d'autres fonctionnalités, car le domaine de la génomique, génétique qui tend vers la médecine personnalisée est un domaine exigeant.

Tout ce qui touche aux outils décisionnels par rapport à la santé d'un patient est hautement régulé par les ordres professionnels médicaux et les gouvernements. Le but de cette expérimentation en laboratoire était donc de repousser les limites actuelles quant à l'affichage du profil des patients afin de pouvoir enrichir la deuxième version de ce prototype logiciel avec d'autres fonctionnalités désirables par les chercheurs du domaine médical. Nous souhaitons ainsi que cette preuve de concept ouvre de nouveaux horizons et les esprits afin de réévaluer les méthodes d'exploration génomique dans ce domaine.

Nous planifions, en 2019, l'ajout d'autres fonctionnalités et la création de partenariats avec des experts du domaine médical, génétique et génomique afin de finaliser une première version opérable de GNOMEViewer et de le mettre au service de la communauté.

ANNEXE I

BASE DE DONNÉES PUBLIQUE

Tableau A-1 Outils et ressources pour visualiser des données multidimensionnelles en oncogénomique. Tirée de Schroeder MP, 2013

Name	Description	Visualization type	Tool type	Data that can be visualized
cBio Cancer Genomics Portal [32] http://www.cbioportal.org	Resource for visualizing TCGA and other data sets with many features, of which the network viewer and OncoPrint are of special interest. In the network viewer, the portal overlays multidimensional genomics data onto all nodes that are representing genes. This provides the frequency of mutations and copy number alterations (and optionally, mRNA up-/downregulation). OncoPrint shows the same alteration data in a matrix heatmap	Networks Matrix Heatmaps	Web tool	Pre-calculated TCGA and other data sets
CircleMap [8] http://sysbiosoe.ucsc.edu/nets	Tool that produces heatmaps with a circular layout. Different data sets coming from the same samples can be plotted as different layered circles that form a node. The data layers are plotted maintaining the sample order, which can be adjusted by the user	Circular heatmaps	Command line application web tool	Any user-prepared data
Circos [24] http://circos.ca/	Tool for visualizing data and information in a circular layout. It allows intuitive exploration of the relationships between genomic positions, which are depicted as ribbons. Different genomic data types can be represented in different layers of the circle. To a great extent, the color code and plot style for each layer (or data set) can be adjusted by the user	Circular genomic coordinates	Command line application	Any user-prepared data
Caleydo StratomeX [34] http://stratomex.caleydo.org	Tool prepared for the visualization of interdependencies between multiple datasets. It allows exploration of relationships between multiple groupings and different datasets. It can cluster genomics data of different alterations and represents them as matrix heatmaps. The different groupings are connected by ribbons whose width corresponds to the number of samples shared by the connected clusters. Clinical data and pathway maps can be integrated to characterize the clusters	Matrix heatmap with option to visualize pathway maps	Desktop application (Java)	Any user-prepared data (matrices, clusterings). Prepared TCGA data available at http://compbio.med.harvard.edu/tcga/stratomex
Cytoscape [36] http://www.cytoscape.org	Software for visualizing complex networks and integrating these with any type of attribute data such as genomics data and clinical patient information. An extensive library of community-developed plugins is available, some of which (for example, Reactome Fis) focus on cancer data analysis [38]	Networks	Desktop application (Java)	The stand-alone application supports any user-prepared network or attribute data. Additional data are available via various plugins (for example, GeneMANIA [72] for networks)
Genomica [73] http://genomica.weizmann.ac.il	Tool that can be used to analyze and visualize genomic data. Data can be visualized as heatmaps or along genomic coordinates. Module maps and module networks can be created from expression data and can integrate gene expression data, DNA sequence data, and gene and experiment annotations	Matrix heatmap Genomic coordinates	Desktop application (Java)	User-prepared data
Gitools [31] http://www.gitools.org	Tool for analysis and visualization of genomic data using interactive heatmaps. It allows loading of multidimensional matrices (with several values per cell), and thus is very well suited for the visualization and exploration of multidimensional cancer genomics data. It contains several analyses and options that are specifically designed for the exploration of cancer genomics data	Matrix heatmap with interactive features	Desktop application (Java)	Any user-prepared data and data imported from IntOGen [33] database, as well as any Biomart [69,73] database
Integrative Genomics Viewer (IGV) [20] http://www.broadinstitute.org/igv	Visualization tool for interactive exploration of integrated genomics datasets, with a focus on good performance when working with large data sets. All tracks can be annotated with color-coded sample and clinical information; genomic regions can be annotated with text labels. All of the common genomic file formats are supported, including array-based data, next-generation sequence data formats and genomic annotations	Genomic coordinates	Desktop application (Java)	User-prepared data and data from the IGV server, including some TCGA data. In addition, IGV can be accessed from external tools such as GenePattern [68]

Tableau A-1 Outils et ressources pour visualiser des données multidimensionnelles en oncogénomique. Tirée de Schroeder MP, 2013 (suite)

Name	Description	Visualization type	Tool type	Data that can be visualized
IntOGen [33] http://beta.intogen.org	Resource that is used to analyze and visualize cancer genomics data, including expression, copy number variation and somatic mutation data from cancer genomic projects. Various visualization options are offered, of which web-interactive heatmaps (using jheatmap [74]) are of special interest. These are used to display alterations per gene in a cohort of tumor samples or in a set of tumor types	Matrix heatmaps with interactive features	Web tool	Pre-calculated data from more than 300 cancer genomic experiments and user-prepared data for somatic mutations in tumors
NAViGaTOR [75] http://ophid.utoronto.ca/navigator	Tool for visualizing and analyzing protein-protein interaction networks (Network Analysis, Visualization and Graphing TORonto). The network visualization options can be customized to represent genomic data properties by automatically mapping attribute values to visual properties	Networks	Desktop application (Java)	User-prepared data. Data can also be loaded via plugins from multiple portals (such as Reactome [76] or KEGG [77])
Regulome Explorer [70] http://explorer.cancerregulome.org	Tool for the integrative exploration of associations between clinical and molecular features of data from the TCGA project. The visualization is interactive and the displayed data can be filtered according to different criteria. Visualization options include circular and linear genomic coordinates and networks	Circular and linear genomic coordinates Networks	Web tool	Pre-calculated TCGA data
Savant Genome Browser [22] http://genomesavant.com/savant	Desktop visualization and analysis browser for genomics data. This tool was primarily developed for the effective visualization of large sets of high-throughput sequencing data, similar to IGV. Multiple visualization modes enable the exploration of genome-based sequence, points, intervals, or continuous datasets. Plugins are available, amongst which is the WikiPathways [78] plugin, which aids the navigation of the data by the integration of pathways	Genomic coordinates	Desktop application (Java)	User-prepared data or data that can be downloaded through plugins such as the UCSC Explorer plugin
The Cancer Genome Workbench (CGWB) [79] https://cgwb.nci.nih.gov/	Host for mutation, copy number, expression, and methylation data from a number of projects. It has tools for visualizing sample-level genomic and transcription alterations in various cancers. The main viewers in CGWB are Integrated tracks view, Heatmap view and Bambino, an alignment viewer. The interface of CGWB is based on the UCSC Genome Browser [80]	Genomic coordinates Heatmap	Web tool	Pre-calculated data from various resources (such as Cosmic, NCI-60 and TCGA [4065,81]) The user can also add custom data tracks for visualization
UCSC Cancer Genomics Browser [21] https://genome-cancer.ucsc.edu	Tool for hosting, visualizing, and analyzing cancer genomics datasets. The browser can display genome-wide experimental measurements for multiple samples, which can originate from multiple data sets alongside their associated color-coded clinical information. The browser provides interactive views of data from genomic regions to annotated biological pathways and user-contributed collections of genes. Integrated statistical tools provide quantitative analysis within all available datasets	Genomic coordinates Heatmap	Web tool	TCGA data and data from independent publications available from the UCSC server. In addition to open access to public datasets, the browser provides controlled access to private project data

Tableau A-2 Caractéristiques principales de plusieurs bases de données publiques couramment utilisées en oncogénomiques, ainsi que le type de données qu'on y retrouve. Tirée de Klonowska K, 2016

database	data source	sites of analysed cancer[1]	organisation of data[2]	oncogenomic data/ analyses	link/literature
Tumorscape	Broad Institute	Bd; Bld; Br; Bra; Clr; Eso; GIST; HN; Htp; Kd; Lng; Lvr; Lymph; Msh; Ov; Pnc; Prst; Sk; ST; Stc; Swn; Thr; Utr; also in: cancer cell lines	level i-iii	copy number alterations	http://www.broadinstitute.org/tumorscape/pages/portalHome.jsf; [12]
UCSC Cancer Genomics Browser	TCGA, SU2C Breast Cell Line, Cancer Cell Line Encyclopedia, The Connectivity Map, TARGET, cancer data from literature	Bd; Bld; Br; Bra; Chl; Col; Clr; EG; Eso; HN; Kd; Lng; Lvr; Lymph; Msh; Ov; Pan; Pnc; Prc / Prn; Prst; Rc; Sk; ST; Stc; Thm; Thr; Utr; also: cancer cell lines; cancer data from mouse models	level i-iii	DNA copy number, miRNA/exon/gene/ protein expression, DNA methylation, gene-level mutations, PARADIGM pathway activity; clinical, epidemiological, and molecular information	https://genome-cancer.ucsc.edu; [14-18]
ICGC Data Portal	ICGC, TCGA ,TARGET	Bd; Bld; Bo; Br; Bra; Clr; Col; Eso; HN; Kd; Lng; Lvr; Lymph; Nb; Ov; Pnc; Prst; Rc; Sk; ST; Stc; Thr; Utr;	level i-iv	simple somatic mutations, copy number somatic alterations, structural somatic mutations, simple germline variants, DNA methylation, gene/ protein expression, miRNA expression, exon junction; epidemiological and clinical data	https://dcc.icgc.org; [32]
COSMIC	TCGA, ICGC, cancer data from literature	Bo; Br; EA; Eso; GIST; Htp; Kd; Lvr; Lng; Ov; Pnc; Prst; Sk; Stc; Tst; Thm; Thr; Utr	level iii-iv	somatic mutations, copy number alterations, gene expression	http://www.sanger.ac.uk/genetics/CGP/cosmic; [39-43]
cBioPortal	AMC, BCCRC, BGI, British Columbia, Broad, Broad/ Cornell, CCLE, CLCGP, Genentech, ICGC, JHU, Michigan, MKSCC, MKSCC/ Broad, NCCS, NUS, PCGP, Pfizer UHK, Riken, Sanger, Singapore, TCGA, TSP, UTokyo, Yale	ACC; Bd; Bld; Br; Bra; Chl; Clr; Eso; HN; Kd; Lng; Lvr; Lymph; MM; Npx; Ov; Pnc; Prst; Sk; ST; Stc; Thr; Utr; also: cancer cell lines	level iii-iv	mutations, putative copy number alterations; mRNA expression, protein/ phosphoprotein level; survival analyses	http://www.cbioportal.org; [57, 58]
IntOGen (2014.12)	TCGA, ICGC, cancer data from literature	Bd; Bld; Br; Bra; Clr; Eso; HN; Kd; Lng; Lvr; Lymph; Ov; Pnc; Prst; Sk; Stc; Thr; Utr	level iii-iv	results of the analyses indicating driver alterations and genes; therapies tailored to the mutation profiles of the analyzed patients	http://www.intogen.org/; [67-70]
BioProfiling.de					
PPISURV	for gene expression: Gene Expression Omnibus; for interactome: IntAct, HPRD, Reactome, HumanCyc, NCI_NATURE, PhosphoSitePlus	Bd; Bld; Br; Bra; Col; Htp; Lng; Lvr; Lymph; Ov; Prst; ST; Utr	level iv	survival analyses	http://bioprofiling.de/GEO/PPISURV/ppisurv.html; [81]
MIRUMIR	Gene Expression Omnibus	Br; Eso; Lvr; Lng; Npx; Ov; Prst; Sk	level iv	survival analyses	http://www.bioprofiling.de/GEO/MIRUMIR/mirumir.html; [83]
DRUGSURV	for gene expression: Gene Expression Omnibus; for drugs modulating a gene of interest: DrugBank, Pubchem Bioassay	Bld; Br; Bd; Col; Bra; Lng; Lvr; Lymph; Prst;; ST; Utr	level iv	list of drugs targeting specific genes/ cancer types; survival analyses	http://www.bioprofiling.de/GEO/DRUGSURV/index.html; [85]

[1]List of abbrieviations of cancer sites. In the brackets there are exemplary cancer subtypes included in the portals.
ACC – adenoid cystic carcinoma; Bd – bladder; Bld – blood; Bo – bone; Br – breast; Bra – brain; Chl – cholangiocarcinoma; Clr – colorectal; Col – colon; EA – eye and adnexa; EG - endocrine glands; Eso – esophagus; GIST – gastrointestinal; HN – head and neck; Htp – hematopoietic; Kd – kidney; Lng – lung; Lvr – liver and biliary tract; Lymph – Lymphoma; Msh – mesothelioma; Mth – mouth; Nb – neuroblastoma; Npx – nasopharynx; Ov – ovary; Pan – pancancer; Pnc – pancreas; Pnx – pharynx; Prc/Prn - pheochromocytoma and paraganglioma; Prst – prostate; Rc – rectum; Sk – skin; ST – soft tissues; Stc – stomach; Swn – schwannoma; Thm – thymus; Thr – thyroid; Tst – testis; Utr – uterine (cerxix and corpus).
[2]In oncogenomic portals cancer resources are arranged in different levels of organisation, including: (i) raw, (ii) computationally processed/normalized, (iii) interpreted and (iv) summarized data [3].

Tableau A-3 Bases de données publiques dans lesquels on peut trouver des jeux de données gratuitement, et le type d'accès qu'on y retrouve. Tirée de Chin L, 2011

Database	Link	Data type	Type of information	Access
ICGC	http://dcc.icgc.org/	Levels I-IV	Copy number, rearrangement, expression, and mutation data	Open and controlled
TCGA	http://cancergenome.nih.gov/dataportal	Levels I-III	Copy number, expression (mRNA and miRNA), promoter methylation, and mutation sequencing	Open and controlled
NCBI dbGAP	http://www.ncbi.nlm.nih.gov/gap	Levels I-II	Raw sequencing traces; second-generation sequencing BAM files by TCGA	Controlled
COSMIC	http://www.sanger.ac.uk/genetics/CGP/cosmic	Levels III-IV	Somatic mutations and copy number alterations by gene: amino acid position, tumor type, literature references	Open
Cancer Gene Census	http://www.sanger.ac.uk/genetics/CGP/Census	Level IV	Annotation of mutated or genomically altered genes	Open
WTSI CGP	http://www.sanger.ac.uk/genetics/CGP/Archive	Levels I-II	First-generation trace archive; SNP genotype profiles	Controlled
EGA	http://www.ebi.ac.uk/ega	Levels I-II	Second-generation sequencing BAM files generated by WTSI CGP	Controlled
Tumorscape	http://www.broadinstitute.org/tumorscape	Levels I-IV	Browsable, searchable cancer copy number viewer using SNP array data	Open
Oncomine	http://www.oncomine.org	Level IV	Gene expression and copy number data in readily searchable and comparable fashion	Password-protected
GEO	http://ncbi.nlm.nih.gov/geo	Level I	Gene expression data	Password-protected
caArray	http://caarray.nci.nih.gov	Level I	Gene expression data	Password-protected
UCSC Cancer Genome Browser	https://genome-cancer.soe.ucsc.edu	Levels III-IV	Browsable viewer for cancer copy number and expression data	Open
The cBio Cancer Genomics Portal	http://cbioportal.org	Levels III-IV	Browsable and searchable viewer for cancer copy number and expression data	Open
OMIM	http://www.ncbi.nlm.nih.gov/omim		Inherited syndromes and causative genes for cancer and other diseases, with extensive literature review	Open
Mitelman	http://cgap.nci.nih.gov/Chromosomes/Mitelman		Copy number alterations and translocations based on cytogenetic data	Open

[Level I] Raw; [Level II] normalized/processed; [Level III] interpreted; [Level IV] summarized.

ANNEXE II

BASE DE DONNÉES PUBLIQUE PROCESSUS DE CONVERSION D'ADAM À L'AIDE D'UNE GRAPPE EMR

1. At General configuration:

- *Give cluster a name*
- *Select cluster*

Software configuration:

- *Take the latest EMR version with first option on applications*

Hardware configuration:

- *Select the right instance taking in account the size of the files you will process on your instance. In my case I selected an m4.xlarge.*
- *3 instances.*

Security Acces:

- *Link your EC2 acces key here. All the rest should be kept default. See figure 1 below.*

You could either choose to install Spark on your cluster, or add Spark from the console with the add step option, and select the Spark application.

- ***You might want to edit the security credential group at the inbound rules to make sure that you can connect to your cluster. Or else you'll get an error message that your connection timed out.***
- ***For this click add rule on your master node, select SSH, protocol TCP, at source make sure that it is locked to your IP address by selecting MyIP. Then click save.***

2. *To connect by SSH, click cluster details then SSH, the ssh command for your connexion to your master node will be displayed whether you have Linux/MAC or Windows. This is how you'll access your EMR cluster through a shell.*

To connect to your EMR cluster from your computer, you may navigate to the folder containing you .pem file with your EC2 key, or put it on your path like this:

```
1. ssh -i ~/GenomeViewer.pem hadoop@ec2-54-215-245-50.us-west-
   1.compute.amazonaws.com
```

Once connected to your EMR it would be good practice to configure your .bashrc file with all terminal configuration and path to applications that you'll be running. Before you begin, this is the content of the .bashrc file.

```
# Source global definitions
if [ -f /etc./bashrc ]; then
  . /etc./bashrc
fi

# set the default region for the AWS CLI
export AWS_DEFAULT_REGION=$(curl --retry 5 --silent --connect-timeout 2
http://169.254.169.254/latest/dynamic/instance-identity/document | grep région | awk -F\"
'{print $4}')
export JAVA_HOME=/etc/alternatives/jre
```

You will need Git installed on your cluster, so the command to do that is (make sure you are root with: sudo -s):

```
yum install git
```

Maven installation:

- *wget http://mirror.its.dal.ca/apache/maven/maven-3/3.5.0/binaries/apache-maven-3.5.0-bin.tar.gz*
- *tar -zvxf apache-maven-3.5.0-bin.tar.gz*
- *rm apache-maven-3.5.0-bin.tar.gz*
- ***Edit your bashrc file to put the following Maven environment variables:***
  - *vi ~/.bashrc*
  - ***bashrc looks like this for now:***

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc./bashrc ]; then
        . /etc/bashrc
fi
export AWS_DEFAULT_REGION=$(curl --retry 5 --silent -connect-timeout 2
        http://169.254.169.254/latest/dynamic/instance-identity/document | grep région | awk
        -F\" '{print $4}')
export JAVA_HOME=/etc/alternatives/jre
export PATH=/home/hadoop/apache-maven-3.5.0/bin:$PATH
export MAVEN_OPTS="-Xmx512m -XX:MaxPermSize=256m"
```

- ***then do: source ~/.bashrc to load the environment variables to your workspace.***
- ***Test if maven is working properly with: mvn –v which will yield:***

Apache Maven 3.5.0 (ff8f5e7444045639af65f6095c62210b5713f426; 2017-04-03T19:39:06Z)

Maven home: /home/hadoop/apache-maven-3.5.0

Java version: 1.8.0_121, vendor: Oracle Corporation

Java home: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.121-0.b13.29.amzn1.x86_64/jre

Default locale: en_US, platform encoding: UTF-8

OS name: "linux", version: "4.4.35-33.55.amzn1.x86_64", arch: "amd64", family: "unix"

ADAM installation:

Get the latest version of ADAM.

- *git clone https://github.com/Emile-Filteau/adam.git*
- *cd adam*
- *mvn clean package -DskipTests*
- ***Then add the following to your .bashrc file:***

export ADAM_HOME=/home/hadoop/adam

alias adam-submit=$ADAM_HOME/bin/adam-submit

alias adam-shell=$ADAM_HOME/bin/adam-shell

then do:

source ~/.bashrc

- ***Test adam with : adam-submit and adam-shell to make sure that your installation is good.***

Setup S3 instance on EC2 for better I/O from S3 bucket:

- *yum install gcc libstdc++-devel gcc-c++ fuse fuse-devel curl-devel libxml2-devel mailcap git automake*
  - *git clone https://github.com/s3fs-fuse/s3fs-fuse.git*
  - *cd s3fs-fuse/*
  - *./autogen.sh*

- *./configure --prefix=/usr --with-openssl*
- *make*
- *make install*
- ***test if working:** s3fs **should yield:***

s3fs: missing BUCKET argument.

Usage: s3fs BUCKET:[PATH] MOUNTPOINT [OPTION]...

Get access key to S3 bucket:

https://console.aws.amazon.com/iam/home?#/home

- ***Select Users and click on your user name***
- ***Navigate to security credentials***
- ***Create access key which will generate a file with access keys.***
- *vi /etc/passwd-s3fs*
- *The s3fs password file has this format (use this format if you have only one set of credentials):*

  accessKeyId:secretAccessKey

  If have more than one set of credentials, then you can have default credentials as specified above, but this syntax will be recognized as well:

  bucketName:accessKeyId:secretAccessKey
- *chmod 600 /etc/passwd-s3fs*
- *sudo chmod 640 /etc/passwd-s3fs*
- ***add this line to your .bashrc which will enable you to use the s3fs command anywhere:***

alias s3fs=/usr/bin/s3fs

source ~/.bashrc

Mount the S3 bucket with:

s3fs goat-adam-genes GenomeViewer/ -ouse_cache=/tmp

***the ouse_cache option enables to minimize downloads.**

- ***Then you can navigate through your bucket with usual linux commands.***

Convert .vcf files to adam:

- *aws s3 cp s3://goat-adam-genes/ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/ALL.chr10.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz /home/hadoop/HG38_temp/*
- *gunzip /home/hadoop/HG38_temp/ ALL.chr10.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz*
- *hadoop –fs –put /home/hadoop/HG38_temp/ ALL.chr10.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.vcf /user/hadoop/HG38*
- *adam-submit vcf2adam "/user/hadoop/HG38//home/hadoop/HG38_temp/ ALL.chr10.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf" "/user/hadoop/adamfiles/"*
- ***To copy the adam parquet files to s3 bucket mounted on EC2 do:***

hadoop fs -get /user/hadoop/HG38_adam/ /mnt/GenomeViewer/HG38_adam/Chromosome22/

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

A.B. (2001). *MySQL.* MySQL reference manual. *Redwood Shores, CA: Oracle*.

Amazon Web Services, Inc. or its affiliates (2013). *Discussion Forums*. Consulté le 05 12, 2018, sur AWS: https://forums.aws.amazon.com/thread.jspa?threadID=46575

Andersen S.L., Sekelsky J. (2010). Meiotic versus mitotic recombination: two different routes for double-strand break repair: the different functions of meiotic versus mitotic DSB repair are reflected in different pathway usage and different outcomes. *BioEssays, 32(12)*:1058-1066.

Bedard P.L, Hanson A.R. et al. (2013). Tumour heterogeneity in the clinic. *Nature*, 501 (7467):355-364.

Billaud M., Guchet, X. (2015). L'invention de la médecine personnalisée: Entre mutations technologiques et utopie. *Médecine/sciences, 31*:797-803.

Bossa S., et al. (2017). *United States patent application Brevet n° US 14/856,001*.

Bowman R.L., Wang Qianghu et al. (2017). GlioVis data portal for visualization and analysis of brain tumor expression datasets. *Neuro-Oncology, 19(1):*139-141.

Brikman Y. (2017). Terraform: Up and Running*: Writing Infrastructure as Code. O'Reilly Media, Inc., first edition,* 206p.

Broad Institute, a. t. (2013-2018). *Integrative Genomics Viewer*. Consulté le 09 12, 2018, sur https://software.broadinstitute.org/software/igv/MainWindow

Carbone P., Ewen S. and Haridi S. (2015). Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering.*, *36(4*):28-38.

Cassandra A. (2015). *Apache Cassandra. Apache Software Foundation*. Consulté le 2 12, 2018, sur http://cassandra.apache.org

Chang F., Dean J. et al. (2008). Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS).*, 26(2):4:1-4:26.

Chin L, Hahn W. et al. (2012). Making sense of cancer genomic data. *Genes & development,* 25(6), 534-555.

Cloud A.E. (2011). Amazon web services. Consulté le 2 12 2018, sur https://aws.amazon.com

Cock P.J., Antao T. et al. (2009). Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422-1423.

Cutting D. (2011). *Data interoperability with apache avro*. Cloudera. Consulté le 2 12, 2018, sur https://blog.cloudera.com/blog/2011/07/avro-data-interop/

DeCandia G., Hastorun D. et al. (2007). Dynamo: Amazon's highly available key-value store. *ACM, 41*(6):205-220.

do Valle I.F., Giampieri E. et al. (2016). Optimized pipeline of MuTect and GATK tools to improve the detection of somatic single nucleotide polymorphisms in whole-exome sequencing data*. BMC Bioinformatics, 17(Suppl)12*:341.

1000 Genomes Project Consortium. (2010). A map of human genome variation from population-scale sequencing.*Nature, 467(7319):* 1061

Fedosejev A. (2015). React. js Essentials. *(Éd.) Packt Publishing Ltd.,* 210p.

Forcier J.E., Bissex P., Chun, W.J. (2008). *Python web development with Django*. Addison-Wesley Professional. 408p.

Gammal R.S., Court M.H. et al. (2015). Clinical Pharmacogenetics Implementation Consortium (CPIC) Guideline for UGT1A1 and Atazanavir Prescribing. *Clinical Pharmacology & Therapeutics. 99*(4):363-369.

Gibson G. (2010). *Hints of hidden heritability in GWAS" Nature Genetics.(doi:10.1038/ng0710-558. PMID 20581876.), 42(7):558–560.* . Consulté le 01 12, 2018, sur https://upload.wikimedia.org/wikipedia/commons/1/12/Manhattan_Plot.png

Goldman M., Craft B., et al. (2016). The UCSC Xena system for integrating and visualizing functional genomics. *Cancer Research. 76(14 Supplement)*:5270-5298.

Guay D., Filteau-Tessier O. et al. (2017) GOAT - *Genetic Output Analysis Tool GenomeViewer - An interactive Somatic Mutation Visualize*r. Consulté le 2 12 2018, http://publicationslist.org/data/a.april/ref-587/GTI795-LOG795-GOAT-Rapportfinal.pdf

GUIDE, U. (2015). *Presto.*

Hayton R. (2011). *Washington, DC: U.S. Patent and Trademark Office Brevet n° U.S. Patent No. 7,873,965.*

HBase A. (2016). Welcome to Apache HBase. *Viitattu*, 12.

Hunter J.E., Irving S.A. et al. (2016). A standardized, evidence-based protocol to assess clinical actionability of genetic disorders associated with genomic variation. *Genetics in Medicine*. *18(12)*:1258-1268.

Hussin J. (2013). *Genomic variation in recombination patterns: implications for disease and cancer.* Thèse de Ph.D., Université de Montréal, Biochimie. Montreal: Université de Montréal. Consulté le 11 12, 2018, sur http://www.iro.umontreal.ca/~hussinju/HussinJ_thesis_hyperlinks.pdf

Httermann M. (2012). DevOps for developers. *Apress*, 196p.

Kanzki B., April A. (2017). GNOMEVIEWER: An Interactive Genomic Somatic Mutation Visualizer. *ACM (Éd.)*, *In Proceedings of the 2017 International Conference on Digital Health 2017* , *July 5-7, London, United Kingdom*, pp:225-226.

Kanzki B., Dupuy V. et al. (2016). GOAT : Genetic Output Analysis Tool: An open source GWAS and genomic region visualization tool. *ACM (Éd.)*, *In Proceedings of the 6th International Conference on Digital Health Conference*, April 11-13, Montreal, Canada, pp:55-59.

Kent W.S., Sugnet C.W. et al. (2002). The human genome browser at UCSC. *Genome research, 6(12)*:996-1006.

Kircher M, Witten D.M. et al. (2014). A general framework for estimating the relative pathogenicity of human genetic variants. *Nature Genetics*, *46*:310-315

Klonowska K, Czubak K. et al. (2016). Oncogenomic portals for the visualization and analysis of genome-wide cancer data. *Oncotarget, 7(1)*:176-192.

Król K, Prus B. (2016). The comparative analysis of selected interactive data presentation techniques on the example of the land use structure in the commune of Tomice. *Polish Cartographical Review.*, *48*(3):115-127.

Lauzon D., Kanzki B. et al. (2016). Addressing Provenance issues in Big Data Genome Wide Association Studies (GWAS). *In Proceedings of the First International Conference on Connected Health (CHASE)*, *27-29 June, Washington DC*, pp:382-387.

Lek M., Karczewski K.L. et al. (2016). Analysis of protein-coding genetic variation in 60,706 humans. *Nature*, *536(7616)*:285-291.

Li H., Handsaker B., et al. (2009). The sequence alignment/map format and SAMtools. *Bioinformatics., 25*(16):2078-2079.

Li J., Doyle M.A. et al. (2014). *Bioinformatics Pipelines for Targeted Resequencing and Whole-Exome Sequencing of Human and Mouse Genomes: A Virtual Appliance Approach for Instant Deployment. PLOS, San Francisco*. Consulté le 09 12 2018, sur https://doi.org/10.1371/journal.pone.0095217

Lindberg H., et al. (2002). *US Brevet n° U.S. Patent Application No. 09/768,389.*

Massie M., Nothaft F. et al. (2013). ADAM: Genomics formats and processing patterns for cloud scale computing. *University of California at Berkeley, Technical Report No. UCB/EECS-2013-2017*, 22p.

Merkel D. (2014). *Docker: lightweight linux containers for consistent development and deployment. Linux journal*. Consulté le 2 12 2018 sur

https://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment

Mussio P., Finadri M. et al. (1992). A bootstrap approach to visual user-interface design and development. *The Visual Computer.*, *8*(2):75-93.

Nielsen C.B., Cantor M. et al. (2013). Visualizing genomes: techniques and challenges. *Nature methods*, *7*:S5-S15.

Oliveros S. (2016). *How to Choose a Data Format*. *Silicon Valley Data Science LLC.* Consulté le 01 12, 2018, sur https://svds.com/how-to-choose-a-data-format/

O'Rawe J., Jiang T. et al. (2013). Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing. *Genome medicine*, *5*(3):1-18.

Paci A., Bleton B. et al. e. (2013). Médecine personnalisée et cancer: organiser et financer l'accès à l'innovation. *Presses de l'Institut Gustave Roussy*, 150p.

PharmGKB and PGRN. (2018). *CPIC Clinical Pharmacogenetics Implementation Consortium*. Consulté le 11 12 2018, sur https://cpicpgx.org/

Pruim R.J. Welch R.P. et al. (2010). LocusZoom: regional visualization of genome-wide association scan results. *Bioinformatics, 18(26)*:2336-2337.

Quiñones M. (2015). *Variants calling and Exome-seq, National Institute of Allergy and Infectious Diseases*. Consulté le 09 12, 2018, sur https://www.slideshare.net/bcbbslides/variant-analysis-and-whole-exome-sequencing

Gentlemen R., G. (2008). R programming for bioinformatics. *Chapman & Hall/CRC,* 301p.

Robinson J.T., Thorvaldsdóttir H. et al. (2011). Integrative genomics viewer. *Nature biotechnology, 29*(1):24-26.

Schröck E., du Manoir S. et al. (1996). Multicolor Spectral Karyotyping (SKY) of Human Chromosomes. *Science*, *273*(5274):494-497.

Schroeder M.P., Gonzales-Perez A., Lopez-Bigas N. (2013). Visualizing multidimensional cancer genomics data. *Genome medicine, 5*:9.

Shifman A.R., Johnson R.M, Wilhelm B.T. (2016). Cascade: an RNA-seq visualization tool for cancer genomics. *BMC genomics., 1*(17):1-11.

Sivasubramanian S. (2012). Amazon dynamoDB: a seamlessly scalable non-relational database service. ACM (Éd.), *In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data,* May 20-24*,* Scottsdale, Arizona, pp. 729-730.

Stoesser G., Sterk P. et al. (1997). The EMBL Nucleotide Sequence Database. *Nucleic Acids Research, 25(1)*:7-14.

Suarez A.W. (2018). *Brevet n° U.S. Patent Application*. Consulté le 10 02*,*247.

THE REGENTS OF THE UNIVERSITY OF CALIFORNIA. (s.d.). *UCSC Genemo Browser Guide*. Consulté le 09 12 2018, sur https://genome.ucsc.edu/goldenpath/help/hgTracksHelp.html

Vavilapalli VK, Murthy A.C et al. (2013). Apache hadoop yarn: Yet another resource negotiator. ACM (Éd.), *In Proceedings of the 4th annual Symposium on Cloud Computing (SoCC' 13),* Oct 1-3, Santa Clara, California, pp. 5:1-5:16.

Vohra D. (2016). Apache Parquet. *In: Practical Hadoop Ecosystem, Apress Berkeley, CA*, 421p.

Walsh C.S. (2015). Two decades beyond BRCA1/2: homologous recombination, hereditary cancer risk and a target for ovarian cancer therapy. *Gynecologic oncology., 137(2)*:343-350.

Watson J.D., Baker T. (2012). Biologie moléculaire du gène. *6ième édition, France: Pearson Education.*

Weinstein J.N., Collison E.A. et al. (2013). *Cancer Genome Atlas Research Network. The cancer genome atlas pan-cancer analysis project*. Nature genetics, 10(46):1113-1120.

Zaharia M., Xin R.S. (2016). Apache spark: a unified engine for big data processing. *Communications of the ACM., 59(11):*56-65.

Zelle J. (2004). Python programming: an introduction to computer science. *Franklin, Beedle & Associates, Inc.,* 528p.

Zhang H., Meltzer P.S., Davis, S.R. (2016). caOmicsV: an R package for visualizing multidimensional cancer genomic data. *BMC bioinformatics., 17*:141.

Zhou X., Edmonston M. et al. (2016). Exploring genomic alterations in pediatric cancer using ProteinPaint. *Nat. Genet.. 48(1)*:4-6.