Évaluation de la profondeur de lecture sur la capacité à détecter des variants dans le cadre d'une expérience de séquençage à haut-débit

Diedhiou Ahmed Bachir

UNIVERSITÉ DE STRASBOURG

igbmc
institut de génétique et de biologie moléculaire et cellulaire

Evaluation de l'influence de la profondeur de lecture sur la capacité à détecter des variants dans le cadre d'une expérience de séquençage haut débit

Rapport de stage

Diedhiou Ahmed Bachir

M1 biostatistique et statistiques industrielles

Maitres de stage :
Dr. Jocelyn Laporte
Dr. Jean Muller

Remerciements

Je tiens à remercier tout particulièrement mes tuteurs, Dr. Jean Muller, Maître de Conférences des Universités, Praticien Hospitalier et membre du LBGI, et Dr. Jocelyn Laporte, Responsable de l'équipe Voie des myotubularines et mécanismes des maladies neuromusculaires associées, sans qui ce stage n'aurait pu voir le jour. Je les remercie de m'avoir accordé toute leur confiance, de m'avoir guidée et transmis leur savoir tout au long de cette expérience à l'IGBMC.

D'autre part, je voudrais également remercier les personnes suivantes pour m'avoir fait partager leurs connaissances et pour l'expérience enrichissante qu'elles m'ont fait vivre au cours de ces deux mois à l'IGBMC :
Stéphanie Le Gras, Ingénieur d'Etude au sein de la plateforme Biopuce et Séquençage de l'IGBMC, pour ses explications concernant les logiciels de bio-informatique et la réalisation de scripts.

Amélie Piton chercheure postdoctorale et Angélique Quartier étudiante en thèse dans l'équipe du Dr. Jean Louis Mandel de m'avoir aidé à mieux comprendre certains concepts en génétique.

Florent Colin, Nicolas Haumesser, Karim Hnia, pour la bonne ambiance qui régnait dans le labo.
Enfin je remercie Serge Uge, pour ses aides quand je le sollicitais, ainsi que tous les membres des équipes Laporte et Jean Louis Mandel.

SOMMAIRE

GLOSSAIRE

Nucléotide	une molécule organique qui est l'élément de base de l'ADN ou l'ARN
Base	une molécule qui fait partie des nucléotides, il existe quatre principales bases dans l'ADN : adénine, cytosine, guanine, thymine, leurs abréviations respectives sont **A, C, T** et **G**
chromosome	Structure composé d'ADN, support de l'information génétique. Les chromosomes sont localisés dans le noyau des cellules de notre organisme. Ils sont porteurs des gènes qui déterminent toutes les caractéristiques d'un individu : nombre de membres, couleurs des yeux, de la peau, etc. Il en existe normalement 23 paires dans le corps humain, chaque chromosome ayant une forme caractéristique
Séquençage d'ADN	Le séquençage de l'ADN, est la détermination de la succession des nucléotides le composant
Séquence d'ADN	C'est un long fil composé par un enchaînement de quatre bases : **A, C, G** et **T**
Variant	Est une variation génétique c'est-à-dire un changement de base (s) à une position sur le génome par rapport à une référence. Les variations peuvent être des polymorphismes (responsables de modifications légères comme la couleur des cheveux) ou bien des mutations responsables de maladies génétiques.

INTRODUCTION

I. Présentation de la structure d'accueil

1. IGBMC

L'Institut de Génétique et de Biologie Moléculaire et Cellulaire (IGBMC), situé à Strasbourg, a été fondé en 1994 par Pierre Chambon et est depuis Janvier 2001 une Unité Mixte de Recherches du CNRS, de l'Inserm et de l'Université de Strasbourg. Dirigé par Pierre Chambon de 1994 à 2002, codirigé de 2002 à 2009 par Dino Moras et Jean-Louis Mandel, dirigé par Olivier Pourquié de 2009 à 2013 puis codirigé Bertrand Séraphin et Yann Hérault. L'IGBMC est l'un des centres de recherche en biomédecine les plus prestigieux d'Europe. Il se consacre à l'étude du génome des eucaryotes supérieurs, au contrôle de l'expression génétique ainsi qu'à l'analyse de la fonction des gènes et protéines. Ces connaissances sont appliquées à l'étude de pathologies humaines telles que les cancers, les maladies monogéniques ou encore les maladies métaboliques. La recherche au sein de l'Institut s'articule autour de quatre grands programmes scientifiques : le département de Biologie du développement et cellules souches s'intéresse aux moyens de régénérer des tissus ou des organes lésés, le département de Biologie structurale intégrative s'intéresse à l'architecture des cellules pour mieux comprendre leur fonction, le département de Génomique fonctionnelle et Cancer cherche à comprendre l'expression génique et certains étatspathologiques liés au dérèglement cellulaire et le département de Médecine translationnelle et Neurogénétique étudie les maladies génétiques humaines et la neurobiologie. En complément de ces quatre départements de recherche, l'Institut est doté de plusieurs plateformes technologiques : la plateforme de Biologie génomique et structurale s'intéresse à la production et à la caractérisation structurale des protéines et autres complexes macromoléculaires, la plateforme Biopuces et Séquençage analyse le transcriptome et le génome à grande échelle grâce à la technologie des puces à ADN et du séquençage de nouvelle génération, le Centre d'imagerie propose un ensemble d'outils en microscopie électronique, optique et confocale ainsi qu'en traitement et analyse d'images, la plateforme Puces à cellules transfectées est une infrastructure post-génomique permettant l'identification de gènes responsables d'un phénotype cellulaire particulier. Ces plateformes sont également ouvertes aux laboratoires extérieurs régionaux, nationaux et internationaux.

2. Département de médecine transrationnelle et neurogénétique

Ce département au sein duquel j'ai été accueilli durant mon stage étudie des maladies génétiques rares et sévères causés par des mutations dans différents gènes. Tout en centrant son attention sur ces maladies génétiques, il met en œuvre des approches multidisciplinaires qui comprennent l'identification des gènes impliqués par séquençage à haut débit, l'étude des fonctions moléculaires et cellulaires de ces protéines dans les cellules et dans C. elegans, la validation des modèles expérimentaux chez des mammifères, ainsi que l'utilisation de vecteurs viraux (AAV) pour les études de physiopathologie et les essais thérapeutiques précliniques. En parallèle, ce département étudie la fonction de ces protéines dans les muscles squelettiques en conditions normales et pathologiques, grâce au développement de nouvelles méthodes d'imagerie (microscopie corrélative et imagerie in vivo), en étroite liaison avec les plateformes de l'IGBMC.

I.CONTEXTE DU STAGE

Le syndrome de Bardet-Biedl : une maladie génétique hétérogène

. Présentation de la pathologie

. Signes cliniques et généralités

Le syndrome de Bardet-Biedl (BBS) est une ciliopathie polymalformative rare, à transmission autosomique récessive. Décrite pour la première fois en 1920 par Georges Bardet puis en 1922 par Arthur Biedl, cette maladie génétique est caractérisée par une combinaison de plusieurs signes cliniques, dont les plus importants sont : une rétinopathie pigmentaire précoce, une polydactylie post-axiale, une obésité, un hypogonadisme chez les sujets masculins, un déficit cognitif variable (se limitant le plus souvent à des difficultés d'apprentissage) et des anomalies rénales. Les manifestations et la sévérité du syndrome varient considérablement d'une personne à une autre, mais seule la rétinite pigmentaire semble constante après l'enfance. Les patients BBS ont donc un pronostic visuel très compromis à l'âge adulte et sont considérés comme malvoyants. De plus, l'obésité dont souffre ces personnes est responsable en parallèle d'un risque élevé de diabète, d'hyperlipidémie et d'hypertension artérielle. L'ensemble de ces symptômes compromettent donc fortement l'intégration sociale et la qualité de vie du malade, qui se voit également soumis à un suivi médical lourd. L'insuffisance rénale représente la principale cause de décès des patients atteints de BBS, puisque l'atteinte rénale peut, à terme, nécessiter le recours à un rein artificiel et conduire à une greffe de rein.
La prévalence du syndrome de Bardet-Biedl reste toutefois faible en Europe, où elle est comprise entre 1/125 000 et 1/175 000. Elle est néanmoins estimée à 1/13 500 dans certaines populations isolées comme les populations bédouines du Koweït, où l'importance du nombre de familles consanguines constitue un facteur de risque de transmission important.

2. Séquençage à haut-débit

Le **séquençage de l'ADN** consiste à déterminer l'ordre d'enchaînement des nucléotides pour un fragment d'ADN donné.

La séquence d'ADN contient l'information nécessaire aux êtres vivants pour se développer, survivre et se reproduire. Déterminer cette séquence est donc utile aussi bien pour les recherches fondamentales visant à savoir comment fonctionnent la cellule que pour des sujets plus appliqués. Ainsi en médecine, elle peut être utilisée pour identifier, diagnostiquer et potentiellement trouver des traitements à des maladies génétiques.

Le principe de base dans tout séquençage d'un génome consiste donc à fragmenter de façon aléatoire ce génome ou de grands morceaux d'ADN dérivés pour obtenir des morceaux d'ADN de quelques centaines de paires de bases. Les extrémités d'un grand nombre de ces petits fragments sont alors séquencées. La séquence complète du génome d'un patient est ensuite reconstruite à partir de ces séquences unitaires, ou lectures, en les alignant sur le génome de référence.

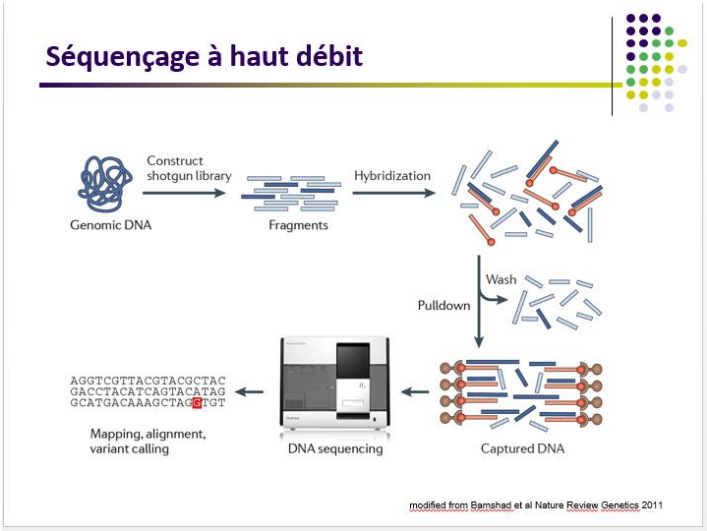


Fig1 : Principaux étapes du séquençage à haut débit

Dans un projet de séquençage, le rapport entre la longueur de l'ensemble des séquences lues mises bout à bout et la longueur du génome cible est nommé profondeur. Par exemple, si l'on séquence 25 millions de bases (Mb) pour un génome de 5 Mb, on a une profondeur de 5 équivalents génome, ce que l'on note 5X. Plus la profondeur est importante, plus nombreuses seront les lectures chevauchantes que l'on pourra assembler, et plus grande sera la fraction du génome couverte. Ceci permet d'obtenir une séquence finale la plus complète possible, avec un minimum de "trous" ou régions non séquencées. Toutefois, si l'augmentation de la profondeur du séquençage permet de diminuer ces lacunes de séquence, il arrive un seuil où il est plus économique de boucher les quelques trous restants de façon ciblée. Par ailleurs, il peut y avoir des biais de représentations qui font que certaines régions sont moins couvertes, voire pas du tout. De plus le nombre de séquences observées à une position du génome est un critère de qualité qui peut avoir une influence sur les variants observés chez un patient.

3. Traitement informatique des séquences

Définitions	
Lecture	Terme générique désignant les séquences sortant du séquenceur
Génome de référence	Séquence connue, supposée être la plus proche possible des données étudiées, utilisée comme pour ancrer et organiser l'information des lectures simples
Couverture	Fraction de la référence couverte par au moins une lecture
Profondeur de lecture	Nombre moyen de lecture par base à une position donnée (elle est exprimée en nombre de X)

Format de fichier SAM	Fichier sous format texte tabulé regroupant les données d'alignements de séquences
Format de fichier BAM	Version binaire d'un fichier .SAM, format conseillé pour le partage des fichiers (fichier de référence)

Tableau1 : Principales définitions en bio-informatique

Dans le cadre d'une expérience de séquençage haut débit l'amplification des fragments d'ADN permet d'obtenir plusieurs copies d'un même fragment. Ainsi pour obtenir une séquence fiable il est indispensable de séquencer plusieurs fois un même fragment. Une fois toutes les séquences obtenues, un traitement informatique leur est appliquée suivant un certain procédé :

-Une comparaison des séquences est effectuée afin d'aligner les parties qui se recouvrent partiellement ou chevauchantes.

-Ensuite vient l'étape du Read mapping qui est l'assemblage des séquences sur le génome de référence

4. Mission

Afin de déterminer la profondeur de lecture minimale nécessaire pour détecter tous les variants chez un patient nous avons utilisé un jeu de données de très haute qualité (couverture >2000X) pour lequel nous allons effectuer une réduction de la qualité. Cette tâche consiste à générer des échantillons aléatoires à partir des séquences d'ADN d'un patient contenues dans un fichier Bam. Ce type de fichier comporte la liste de toutes les lectures (appariées et non appariées) issue du séquençage, et chaque lecture est caractérisée par son identifiant, sa position sur le génome, sa position chromosomique etc…

Nous allons donc échantillonner un fichier Bam de référence en plusieurs sous jeu de données pour lesquels on réalisera les étapes suivantes : 1/évaluation de la profondeur de lecture, 2/détermination des variants et 3/comparaison à l'échantillon de référence non échantillonné afin de déterminer l'impact de la profondeur de lecture sur le nombre de variants .

II.REALISATION DE L'ECHANTILLONNAGE ALEATOIRE

1. Présentation des données

Les données avec lesquelles j'ai travaillées sont des données de séquençage à haut débit issues du séquençage d'un patient porteur du syndrome de Bardet-Biedl. Le fichier Bam mis à disposition a pour identifiant « ASJ-25_1_AUD99 » et toutes ses séquences sont regroupées dans un fichier qui s'appelle ASJ-25_1_AUD99.realigned.recalibrated.bam

Nombre total de lectures	4363718
Nombre de lectures alignées	4363718
Nombre de lectures appariées	4321065
Nombre de lectures non appariées	42653

Tableau 2 : Quelques statistiques des données de séquençage du patient

2. Samtools

Samtools est un ensemble de logiciels utilitaires qui permettent de manipuler des alignements au format SAM/BAM .Il peut importer et exporter les données depuis le format SAM (Sequence Alignment/Map) et permet de trier, mélanger, indexer, récupérer les lectures de n'importe quelle région rapidement. Le format BAM est le résultat de la compression du format SAM (algorithme BGZF par Bob Handsaker) et constitue une version binarisée de SAM ce qui la rend plus rapide pour l'accès aux données. Les principales commandes de samtools sont :

- samtools sort : trie les alignements (par les coordonnées de départ) d'un fichier BAM et produit un BAM trié. Opération requise pour certains programmes (par exemple le visualisateur IGV)
- samtools index : produit un fichier d'index (.bai) à partir d'un fichier BAM trié (voir commande précédente)
- samtools view : imprime dans la sortie standard les alignements (ou portions) demandés à partir d'un SAM ou d'un BAM
- samtools depth : calcule la couverture de chaque base
- samtools merge : fusion de plusieurs BAM triés
- samtools faidx : indexe la séquence de référence (.fasta) et créé un fichier d'index (.fai)
- samtools pileup : imprime l'alignement au format pileup : les séquences sont imprimées en colonnes, chaque ligne correspondant à une position. Les informations telles que identité, insertion, délétion sont représentées aidant ainsi à trouver des SNP

IGV : IGV est un logiciel que j'ai utlisé durant mon stage ,pour la visualisation des données génomiques.

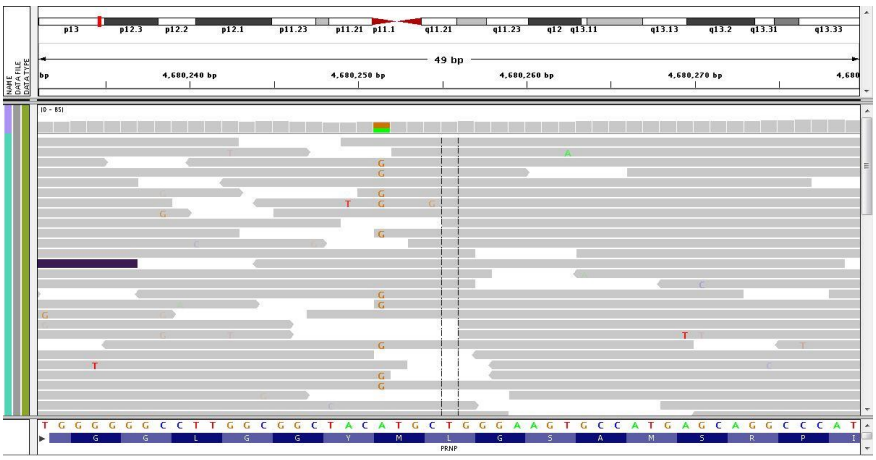


Fig2. Aperçu d'un fichier Bam sous IGV

Pour la première phase de manipulation des données, il a été nécessaire de trouver le meilleur outil, Notre première option a été d'utiliser samtools. J'ai ainsi commencé à me familiariser avec l'outil pour mieux appréhender le contenu et l'architecture des données.

En appliquant la commande samtools view au fichier Bam on parvient à visualiser les séquences ainsi que leurs caractéristiques. Sous samtools une lecture est représentée comme suit :

HWI-ST1136:146:HS071:8:2108:12971:80053 147 chr1 5923188 60 100M = 5923121 -167
TCAGCCAAGTGCACAGGTCAGCCAGGTGGGCACTGAAGTGAAAGGCTGCAGAGAGGCGGGGAGGACAGCCTGCAGGGC
AGGAGGGGCACAGACAGGCCCC
D>AEED@DF@G?ADDGFECCFDCCFE=DDDA?@=EADE=EBAEDEEADDDEDEDDE6CCDCBDECDEDBFACCCBBDCCB
DCBBBCCADEFBCCBCA;=? X0:i:1 X1:i:0
BD:Z:PMPPQQOPOQPPNMQNPPMPPPMQNOQFPPPQQOPOPOOPIOQPQQQPMQMQMQPOIGGMMQMMNLPP
QQQPMQGPPMQMMPGFOOPMLQMMMQPPIINN MD:Z:100 RG:Z:1 XG:i:0
BI:Z:ONNOMKOOPOPOPOOOPPOOPMOOOPOMNKDC=AFMNPONJOOOPMOPONONOOOONNNNOONNPOOPL
MOPOONOPOONOONNOPOPONOPOOOPKKMM AM:i:37 NM:i:0 SM:i:37 XM:i:0 XO:i:0 XT:A:U

Sur ce tableau est transcrit la signification de quelques éléments constituants les champs d'une lecture .Seuls les plus importants pour notre étude sont mentionnés.

Premier champ	HWI-ST1136:146:HS071:8:2108:12971:80053	Identifiant de la lecture
Deuxième champ	147	Flag (numéro indiquant si une lecture est appariée) de la lecture
Troisième champ	Chr1	Chromosome sur lequel la lecture est localisée
Quatrième champ	5923188	Position chromosomique
Cinquième champ	100M	Cigar

Après avoir acquis les notions nécessaires pour manipuler les données, nous avons recherché un moyen de réaliser l'échantillonnage des lectures. J'ai ainsi utilisé la fonction d'échantillonnage aléatoire disponible dans samtools. Cependant la méthode proposée n'est pas très performante. En effet elle est basée sur l'utilisation d'une graine aléatoire (qui est un nombre) que l'on doit fixer avant de lancer le programme d'échantillonnage. Ainsi en exécutant plusieurs fois le programme avec la même graine, on obtient toujours le même échantillon. Jugeant ce procédé pas assez aléatoire, j'ai décidé de travailler sous R, en utilisant la fonction *sample* qui parait plus apte à générer des objets d'une manière plus aléatoire.

Sous R j'ai principalement travaillé avec *Rsamtools* qui est un package disposant de fonctions qui permettent de manipuler des fichiers Bam. Je me suis aussi servie de la fonction

ParseCommandargs() de la librairie **Batch** pour la lecture des variables de mon script passées en argument.

3. Mis en œuvre du script R

3.1. Transformation des données

Sous R, il a été nécessaire de transformer les données afin de pouvoir les manipuler de manières efficaces. J'ai utilisé un script se trouvant dans la documentation de Rsamtools pour parvenir à mon but.

-Tout d'abord le fichier est chargé et transformé en un objet de classe « list » avec la fonction ***ScanBam().*** Après quelques étapes de transformation**,** le Bam est changé en dataframe, que j'appellerai dans le script bam_df.En exécutant la ligne de commande :

> *bam_df<-as.data.frame(bam_df)*

On obtient comme sortie : Un dataframe de treize colonnes et de 4363718 lignes.

```
> head(bam_df)
                                qname flag rname strand      pos qwidth
1 HWI-ST1136:146:HS071:8:1101:12946:61803  147  chr8      - 94767263    100
2 HWI-ST1136:146:HS071:8:1101:12946:61803   99  chr8      + 94767222    100
3  HWI-ST1136:146:HS071:8:1101:15253:5210 1107 chr14      - 89338717     66
4  HWI-ST1136:146:HS071:8:1101:15253:5210 1187 chr14      + 89338506    100
5 HWI-ST1136:146:HS071:8:1101:18606:70177  147 chr11      - 66299497    100
6 HWI-ST1136:146:HS071:8:1101:18606:70177   99 chr11      + 66299413    100
  mapq cigar  mrnm     mpos isize
1   60  100M  chr8 94767222  -141
2   60  100M  chr8 94767263   141
3   60   66M chr14 89338506  -277
4   60  100M chr14 89338717   277
5   60  100M chr11 66299413  -184
6   60  100M chr11 66299497   184
                                                                                                    seq
1 TTCCCTTTCCAGCAGCCGGAGAAGTGCGACAACAACCAGTACTTTGATATCTCCGCCCTCTCGTGTGTTCCTTGTGGAGCTAACCAGAGGCAAGATGCCC
2 TGTTGTTCCTCCCTCGCTTCTTACAGGCCCAGACCTTCTCTTTCCCTTTCCAGCAGCCGGAGAAGTGCGACAACAACCAGTACTTTGATATCTCCGCCCT
3                                 CTCTGGTCAACAACAACAACCACGCCGAGGCCTACAACAACCTGGCTGTGCTGGAGATGCGGAAGG
4 TCATTTTATATTCTTAGGTTAAATCCACTTACGTAGAATTCACATTGACTTCTATAATTGTATGGTACTTGATGCTTTTTGTCTGGTAGCAGAAGAGATA
5 GCGGCCGCCTGAGACCTGGGCTGCTGTGAAAGCCCCTGCACAATCAGCCAGGGAGAACTGGGCGGGTTTAGTGGCCCCAGGCCCACTCCTCATGCAGCAG
6 CCTCCATAGGTGCTGGTGCTTCGAGAAGGCCAAAGTGCACCCCTGCTGAGTGCCCACGTCAACATGCCTGGGAGCGAGGGGCTGGCGGCCGCCTGAGACC
                                                                                                   qual
1 ?>BCEABAE?DGDDGE;GECE?CE>F>A?B>?A>?CBCE@@C>??D@A@DDDDADCCFDFEADADADBDBFD@C@BDCCE@AABCCEDDEDBDD>;C;<>
2 A?>?B@@FDCFDDCD?DDBEDA@CAFBDCCADDDCDBBCD>BABABCBCECAFDAAEC:CDEDBDABB?DE>@E??AA?CAAFD?@DFABAGDFG@EDF<
3                                 69:59;A;<@C@@B@BA==BB??CB<DDDDCFBA=BBDBACFABCFADADF@CFECA@F=DD?@CA
4 ?@;>>>>?@AAABEDA@EBAA@AAADB@CCBAD@BAADBBBE?D@ABDDDDBGEBAABABDBAADCB@ADBDDADDDBBBBCBDFEECAFF>EECBDAC@A
5 F?DEE@GEE@FDBAEE?GFFD>FD>E=D>>B?BC@A=EC@C@?DDEECDECCEDEBBFACCDADCDBBADDBBCBBCCDCCBBCBEDCEDDA=DABC;A?
6 AC@DC=@AED?@EDCCACDDBD@CFDBFCDB@BBEADDACACBCDCDDDFCDDCCADACEACE@ADEDEDDDDDG;AEECEGDDDG@F?AABG><;@ABA
```

Chaque nom de colonne correspond à une information sur la séquence .On observe donc :

-qname : qui est l'indentifiant de la lecture

-flag : flag de la lecture

-rname : nom du chromosome auquel la lecture appartient

-Cigar :indique si la lecture s'aligne ou ne s'aligne pas ,sous R une lecture qui ne s'aligne a pour cigar une donnée maquante (NA)

…..

-Seq :La séquence de la lecture Etc.

3.2 Echantillonnage aléatoire et Obtention des identifiants des lectures

Le dataframe ainsi obtenu on peut effectuer l'étape de l'echantillonnage sous certaines conditions :

1.Exclusion des lectures qui ne s'alignent pas (cigar=NA)

2.Tirage aléatoire sans remise

3.Lors du tirage aléatoire, obligation de selectionner la deuxiéme paires (si elle existe) de chaque reads tirer au hasard (la réalité des données veut que 2 lectures sont toujours pas paires).

Tout d'abord j'ai commencé à créer un sous-dataframe comportant uniquement les colonnes qname, flag, rname, pos et cigar à partir de mon tableau de départ (bamd_df).

>bamqfrpc<-subset(bam_df,select=c(qname,flag,rname,pos,cigar))

Ensuite pour satisfaire à la première condition j'ai fait un na.exclude à ce dataframe afin de supprimer les lignes contenant des données manquantes .

>bamqfrpc<-na.exclude(bamqfrpc)

>head(bamqfrpc)

	qname	flag	rname	pos	cigar
1	HWI-ST1136:146:HS071:8:1101:12946:61803	147	chr8	94767263	100M
2	HWI-ST1136:146:HS071:8:1101:12946:61803	99	chr8	94767222	100M
3	HWI-ST1136:146:HS071:8:1101:15253:5210	1107	chr14	89338717	66M
4	HWI-ST1136:146:HS071:8:1101:15253:5210	1187	chr14	89338506	100M
5	HWI-ST1136:146:HS071:8:1101:18606:70177	147	chr11	66299497	100M
6	HWI-ST1136:146:HS071:8:1101:18606:70177	99	chr11	66299413	100M

En parallèle un second sous-dataframe est créé ne contenant cette fois-ci qu'une seule colonne ,celle des indentifiants :

>bamq<-subset(bamqfrpc,select=c(qname))

	qname
1	HWI-ST1136:146:HS071:8:1101:12946:61803
2	HWI-ST1136:146:HS071:8:1101:12946:61803
3	HWI-ST1136:146:HS071:8:1101:15253:5210
4	HWI-ST1136:146:HS071:8:1101:15253:5210
5	HWI-ST1136:146:HS071:8:1101:18606:70177
6	HWI-ST1136:146:HS071:8:1101:18606:70177

Pour satisfaire à la troisième condition j'ai crée une fonction qui s'appelle *sampling* , faisant dans un premier temps un échantillonnage sur bamq à l'aide de la fonction *sample*, dans un

deuxième temps l'intersection du résultat obtenu avec le tableau bamqfrpc afin de récuper la deuxième paire de chaque lecture échantillonnées.

Exemple illustratif de la fonction *sampling*

1.Sample ()

Tableau 1 (échantillons)

Qname
Lecture A, paire1
Lecture E, paire 2
Lecture C, paire 2
Lecture D, paire 2
Lecture A, paire 2
Lecture E, paire 1

Tableau 2

qname	flag	rname	Pos	cigar
Lecture A, paire1	47	Chr1	5662323	100M
Lecture A, paire2	174	Chr1	5756562	100M
Lecture B, paire1	99	Chr16	4565656	99M
Lecture B, paire2	128	Chr16	4876623	99M
Lecture C,paire 1	230	Chr12	25656255	100M
Lecture C, paire2	18	Chr12	25856562	100M
Lecture D,paire 1	46	Chr4	8555656	5S12M
Lecture D,paire 2	86	Chr4	8545656	5S12M
Lecture E,paire 1	53	Chr14	9565656	99M
Lecture E,paire 2	123	Chr14	9567465	100M

2.merge()

Jointure des tableaux 1 et 2

Qname	flag	rname	Pos	cigar
Lecture A,paire1	47	Chr1	5662323	100M
Lecture A,paire2	174	Chr 1	5756562	100M
Lecture C, paire 1	230	Chr12	25656255	99M
Lecture C,paire 2	18	Chr12	25856562	99M
Lecture D,paire 1	46	Chr4	8555656	100M
Lecture D,paire 2	86	Chr4	8545656	100M
Lecture E,paire 1	53	Chr14	9565656	5S12M
Lecture E,paire 2	123	Chr14	9567465	100M

Fig.3

II. Réalisation du script bash

1. Présentation et usage de bash

Bash est interprétateur de commande du système UNIX, il permet de transmettre des commandes au système d'exploitation afin de réaliser certaines actions d'intérêt.

Un script bash permet d'automatiser une série d'opérations. Il se présente sous la forme d'un fichier (.sh) contenant une ou plusieurs commandes qui seront exécutées de manière séquentielle.

2. Etapes d'enchainement des scripts

Afin d'automatiser les différents étapes de calcul, j'ai écrit cinq script bash dont ceux qui seront exécutés en avant-plan sont mentionnés dans le schéma ci-dessous :

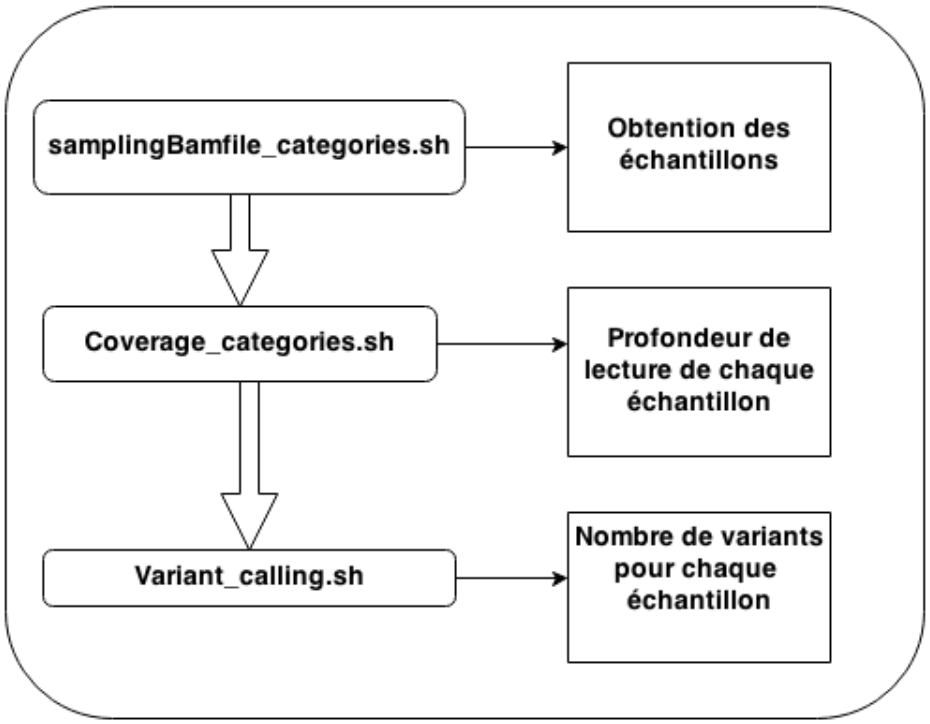


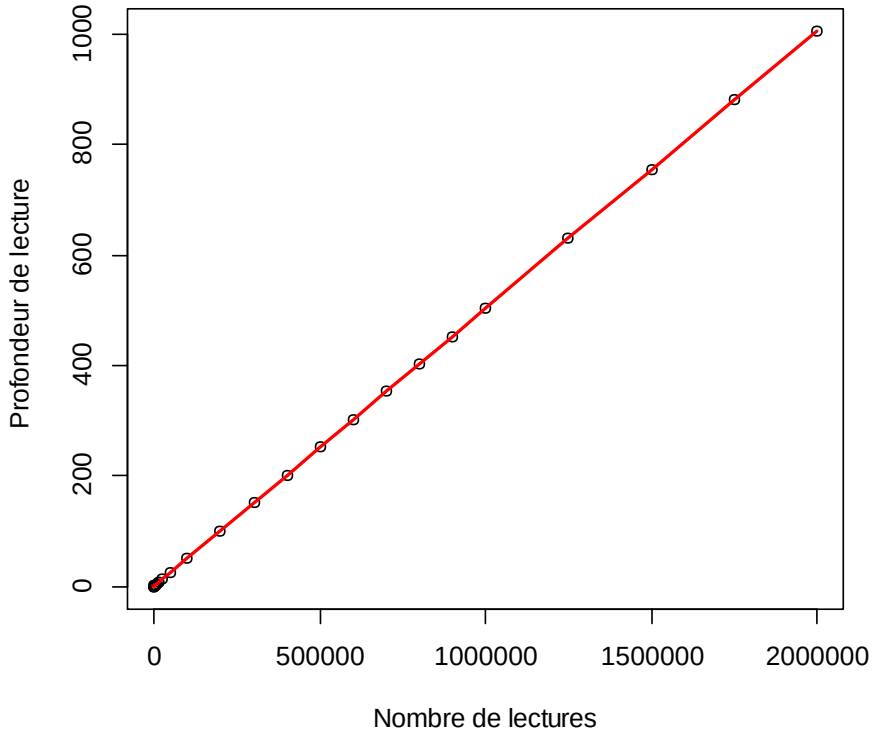
Fig. 4 : Schéma général d'enchainement des scripts bash

IV. DETECTION DES VARIANTS

Le but de cette étape est de déterminer les variants au niveau des régions des gènes ciblés .En effet, il s'agit d'identifier les différences de bases par rapport au génome de référence en fonction des séquences obtenues

1. Evaluation de la profondeur de lecture

Après avoir calculé la profondeur de lecture de chaque échantillon, Notre première approche a été de voir le rapport entre le nombre de lecture et la profondeur .Ceci nous permettra, pour des raisons pratiques de connaitre le nombre de lectures à échantillonner si on veut obtenir un échantillon d'une profondeur donnée.

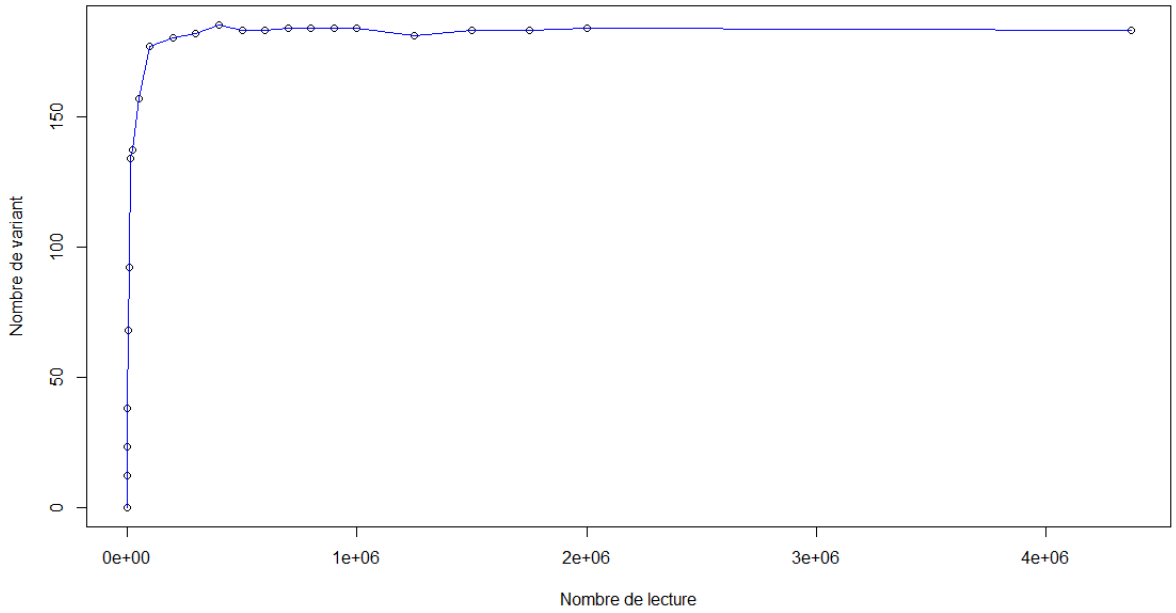


D'après le graphique nous remarquons que la courbe présente une allure linéaire, cela facilite ainsi la connaissance du nombre de lectures échantillonnées si on décide travailler avec un échantillon d'une certaine profondeur.

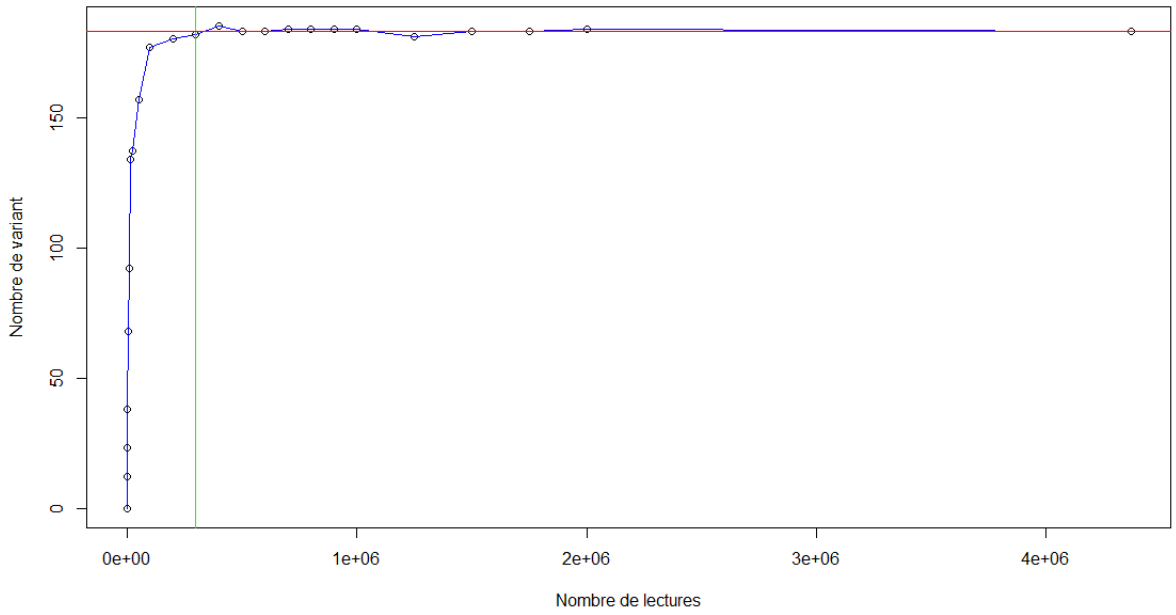
Nous avons ainsi retenu les catégories suivantes : **10 ,500 ,1000 ,2000 ,5000,10000,15000,25000,50000,100000,200000,300000,400000,500000,600000,700000,800000,900000,1000000,12500000,1500000,17500000, 2000000 ,** ceci afin d'évaluer la capacité de détection des variants en fonction du nombre de lecture.

Pour déterminer les variants de chaque échantillon, j'ai utilisé **GATK** qui est une suite de logiciel développée par le Broad Institute pour l'analyse des données de séquençage haut-débit.

Les résultats obtenus avec GATK sur les différentes catégories pour un patient nous ont permis de tracer ce graphe ci-dessous :



On observe au début une augmentation du nombre de variants jusqu'à un point de saturation (voir graphe ci-dessous), ou le nombre de variants n'augmente plus. Nous faisons l'hypothèse que le fichier complet (sans échantillonnage) correspond à la référence (183 variants). Seuls deux échantillons contiennent des variants non présents dans le fichier de référence



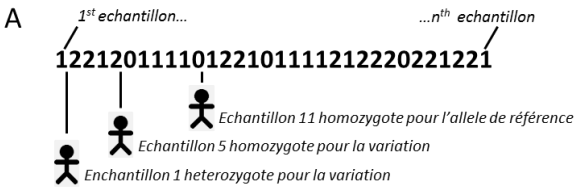
D'après le graphique le point de saturation est 300000, ce qui veut dire dans ce cas, que tous les variants ont été détectés dans un échantillon de 300000 lectures, ce qui en effet correspond à une profondeur de lecture de 115x.

2. Détermination des variants

Définitions	
Variation hétérozygote	Une variation est dite hétérozygote si à une position donnée le nombre de lectures comportant une base différente de celle du génome de référence représente la moitié du nombre de lectures totales à cette position
Variation homozygote	Une variation est dite homozygote si de une position donnée, le nombre le nombre de lectures comportant une base différente de celle du génome de référence représente la quasi-totalité des lectures à cette position

Afin de comparer le nombre de variants, leur statut (hétérozygote/homozygote) pour chacune des catégories testées nous avons utilisé le logiciel VaRank (développé par Jean Muller et Véronique Geoffroy) qui permet de regrouper sous la forme d'un profil la présence d'un variant et son statut (cf. figure). L'ordre du code-barre correspond à l'ordre des catégories échantillonnées et le statut est défini comme suit : « 0 » homozygote normal, « 1 » hétérozygote pour le variant et « 2 » pour homozygote pour le variant.

A



B

Gene	Chr	Start	Ref	Mut	Zygosity	TotalRead Depth	VarRead Depth	cNomen	pNomen	Barcode	#Hom	#Het	#Allele	#Sample
BBS2	16	56548501	C	T	hom	142	142	c.209G>A	p.=	**22222222202222222222222222222**	31	0	62	32
ALMS1	2	73716993	-	C	hom	143	126	c.7911dup	p.Asn2638Glnfs*24	**00000000000000000000000000200000**	1	0	2	32
TTC21B	2	166797646	C	T	het	144	144	c.601G>A	p.Val201Met	**12212011110122101111212220221221**	13	16	42	32

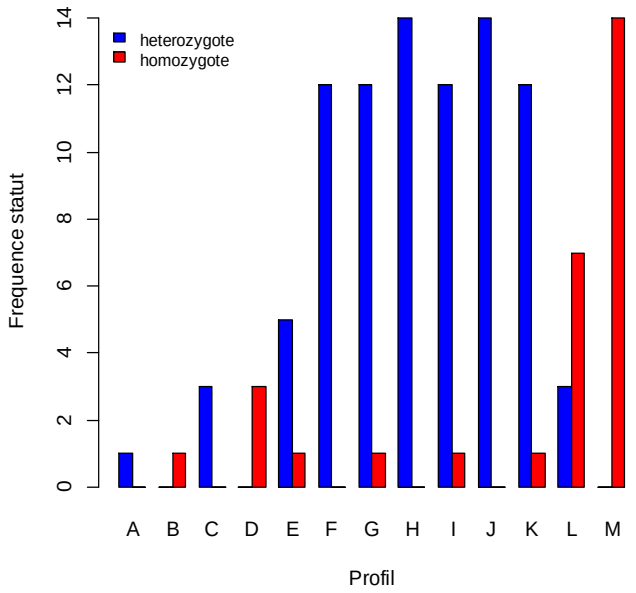
Fig.7 : explication du code-barre

J'ai ensuite défini des classes de profils afin de pouvoir mieux analyser les observations faites.

Profil	Nombre de variants	Fréquence statut Hétérozygote	Fréquence statut Homozygote	Ecart-type Fréquence hétérozygote	Ecart-type Fréquence homozygote
A	19	1	0	0	0
B	24	0	1	0	0
C	6	3	0	0.4	0
D	1	5	3	0	0
E	1	12	1	1.2	0
F	65	12	0	1.5	0
G	1	14	0	0	0
H	1	14	0	0	0
I	7	12	1	0	0
J	4	14	0	0	0
K	26	12	1	0.5	0
L	1	3	7	0	0
M	84	0	14	0	1.6

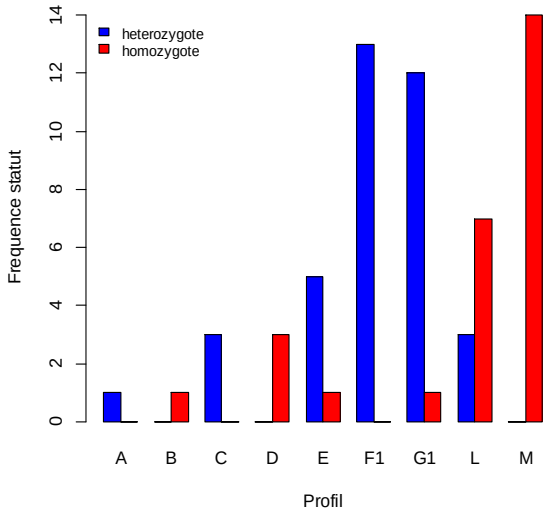
Tableau 5 : Dans ce tableau ce chaque profil est caractérisé par sa fréquence de statut (het /hom) dans les différentes catégories d'échantillon. Un variant peut apparaitre dans plusieurs catégories d'échantillon. Par exemple le profil A correspond aux variants qui sont détectés une seule fois hétérozygote dans différents échantillons. Le profil I correspond aux variants qui sont apparus 12 fois hétérozygote et une fois homozygote.

Pour bien visualiser les données, l'idée a été de tracer un graphique représentant la fréquence des statuts en fonction du Profil (cf. graphique ci-dessous)



D'après ce graphique nous remarquons d'une part une similarité entre certains profils (Ex : le G et le I et K), donc une fusion de ces derniers en un seul profil doit être effectuée afin d'avoir une classification plus optimale du jeu de données .La fusion des profils G ,I et K est directe ,mais en ce qui concerne les profils F ,H et I j'ai décidé de les regrouper en seul profil dont la fréquence de statut sera la moyenne des fréquences de chaque profil .Ce qui nous donne les résultats suivants :

Profil G, I, K←G1 Profil F, H, I←F1

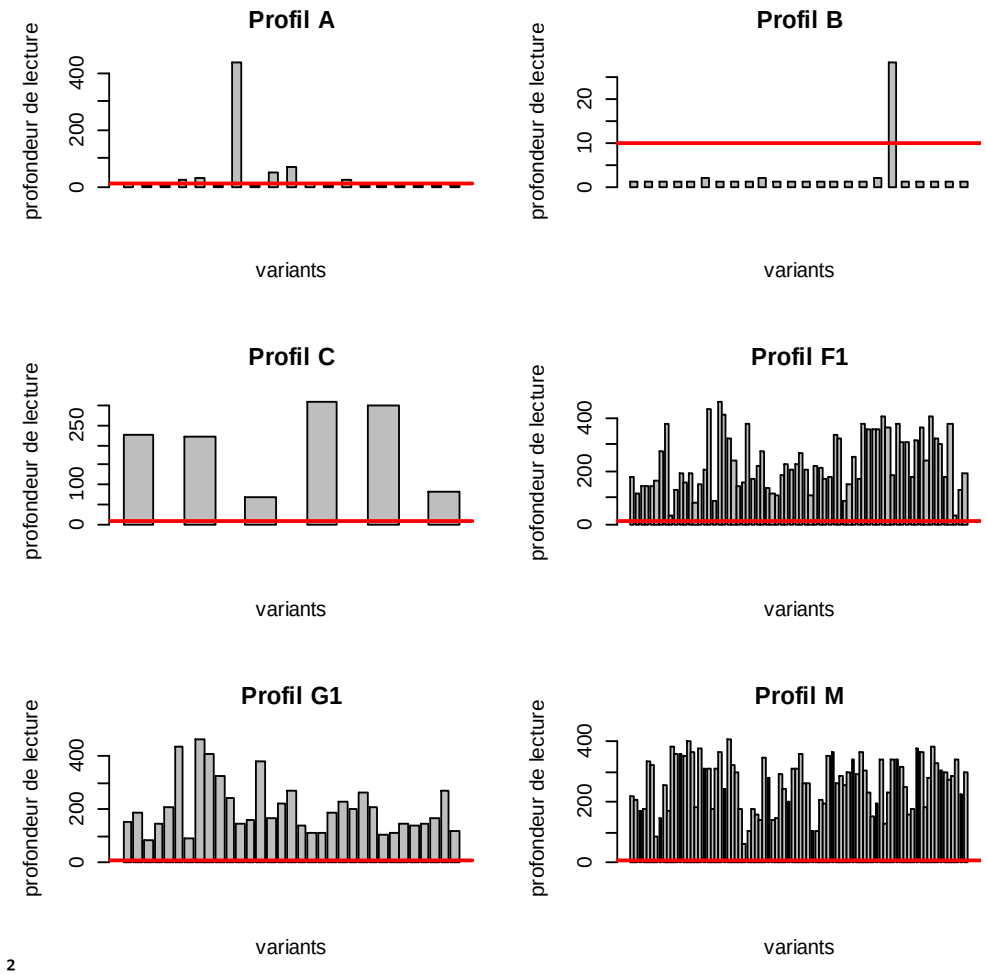


Profil	Nombre de variants	Fréquence Statut hétérozygote	Fréquence Statut homozygote
A	19	1	0
B	24	0	1
C	6	3	0
D	1	5	0
E	1	5	1
F1	70	13	0
G1	33	12	1
L	1	3	7
M	84	0	14

Nous observons bien sur ce graphique pour chaque profil le nombre de fois que chacun des variants d'un profil a eu un statut hétérozygote et homozygote .Pour certains profils comme par l'exemple le profil F1, nous pouvons voir les que tous les variants appartenant à celui-ci sont apparus en moyenne 13 fois hétérozygote dans les différentes catégories d'échantillon, ce qui nous permet de conclure que tous les variants du profil F1 sont hétérozygotes. Ainsi par le même type de raisonnement Nous concluons que les variants du profil C, G1 sont hétérozygotes et ceux du profil M sont homozygotes.

Afin de déterminer les vrais variants, il é a été nécessaire de vérifier pour les profils ayant un nombre importants de variants, la profondeur de lecture à laquelle chaque variant a été découvert .Par convention un variant détecté à une profondeur inferieur à 10x est considéré comme faux positif (variation considéré comme variant alors qu'il n'en est pas un).Cette vérification nous permettra en effet d'avoir au final la liste ou les faux positifs seront exclus.

En ce qui concerne les profils D, E et F une vérification sous IGV nous permettra de valider le caractère positif ou négatif de leur variant. Cependant, j'ai représenté graphiquement la profondeur de lecture des variants de chaque profil en fixant par un trait rouge qui le seuil de profondeur de lecture minimal correspondant à 10x.



2

Profil	Nombre de variants	Profondeur de détection					
		Min	1st quartile	Médiane	Moyenne	3st quartile	Max
A	19	3	3	4	36	24	434
B	24	1	1	1	2	1	28
C	6	69	117	223	201	280	310
F1	70	33	151	206	233	324	463
G1	33	84	142	169	205	240	463
M	84	64	184	284	264	339	408

Nous remarquons que tous les variants des profils C, F1, G1 et M ont été détecté à des profondeurs de lecture supérieur à 10x. Concernant les profils A, seuls 6 variants ont été détecté à des profondeurs de lecture suffisante, ce nombre est d'autant plus faible avec le profil B, ou seulement un seul variant est intéressant. Une autre importante remarque est que lorsque l'on exclut les variants détectés à une profondeur inférieure à 10x (c'est-à-dire les faux positifs), on voit que tous les variants ont été détecté à une profondeur supérieure ou égale à 28x et que la plupart (plus de 50%) ont été détecté à des profondeurs supérieures à 115x.

Conclusion

L'objectif de mon stage a été de réaliser dans un premier temps de mettre au point un programme informatique permettant d'effectuer un échantillonnage aléatoire des données de séquençage à haut-débit, dans un deuxième temps ,de réaliser des scripts bash afin d'automatiser les différents étapes de calculs effectués sur les données, dans le but d'évaluer la profondeur de lecture de chaque échantillon et finalement effectuer une détermination des variants .Nous sommes partis d'un jeu de données de très haute qualité présentant une profondeur de lecture de plus de 2000X (plus de 4 millions de lectures) ,Après réduction de la qualité par échantillonnage aléatoire, nous avons obtenus des données avec des profondeurs allant de 1x à 1000x .Après une analyse des résultats on est parvenu à la conclusion que : afin de détecter tous les variants du patient ,une profondeur de lecture minimale moyenne de 115x (300000 lectures) est nécessaire. .Néanmoins ce résultat ne concerne qu'un seul patient, donc il va falloir pour avoir une approximation exacte de la profondeur de lecture minimale afin de détecter les variants impliqués dans le syndrome de Bardet-Biedl, faire le test sur plusieurs patients.

Ce stage m'aura aussi permis d'acquérir de nouvelles connaissances en informatique très utiles pour la fouille de données.

ANNEXE

1. Scripts réalisé avec R

```
##############################
#####  Sampling_script_1.R    ####
###Chargement des package###
 #load Library
 library(Rsamtools)
 library(batch)
 ParseCommandArgs ()

 #Chargement du fichier bam
 bam <- scanBam(bamfile)

 # fusionner les listes en une seule liste
 .unlist <- function (x){
  x1 <- x[[1L]]
 if (is.factor(x1)){
 structure(unlist(x), class = "factor", levels = levels(x1))
 } else {
 do.call(c, x)
 }
 }

 ###definier comme variable les noms de champs du bam####
 bam_field <- names(bam[[1]])

 ####parcourir les champs du bam délister###
 list <- lapply(bam_field, function(y) .unlist(lapply(bam, "[[", y)))

 ###Transformation du bam en dataframe###
 bam_df <- do.call("DataFrame", list)
 names(bam_df) <- bam_field
 bam_df<-as.data.frame(bam_df)

 ###création de deux dataframe qui seront utilisés pour la jointure###
 bamqfrpc<-subset(bam_df,select=c(qname,flag,rname,pos,cigar))
 bamqfrpc<-na.exclude(bamqfrpc)
 bamq<-subset(bamqfrpc,select=c(qname))
```

```
###Creation de la fonction Sampling###

sampling<-function(n){
a<-1000
j<-n
sampl<-bamq[sample(nrow(bamq), size=a), sample(ncol(bamq), size=1),(replace=F)]
sampl1<-merge(bamqfrpc,sampl,by="qname")
sampl1<-unique(sampl1)
sampl1<-sampl1[1:j,]
sampl1<-subset(sampl2,select=c(qname))
sampl1<-merge(bamqfrpc,sampl3,by="qname")
sampl1<-unique(sampl1)
return(sampl1)
}

Reads<-sampling(h)
####sauvegarde de l'echantillon contenant l'identifiant des lectures####
Mes_reads<-subset(Reads,select=c(qname))
write.table(Mes_reads ,file=nom,quote=FALSE,row.names=FALSE,col.names=FALSE)
q()
```

############################

#####Sampling_script_2.R####

```
##Ce script est exécuté si les fichiers R générés lors du premier échantillonnage##
#####Chargement des packages
library(batch)
parseCommandArgs()
####Chargement des fichiers générés lors du premier échantillonnage##
load(file1)
load(file2)
load(file3)
###echantillonnage avec la fonction sampling######
sampling<-function(n){
a<-1000
j<-n
sampl<-bamq[sample(nrow(bamq), size=a), sample(ncol(bamq), size=1),(replace=F)]
sampl1<-merge(bamqfrpc,sampl,by="qname")
sampl1<-unique(sampl1)
sampl1<-sampl1[1:j,]
```

```
sampl1<-subset(sampl2,select=c(qname))
sampl1<-merge(bamqfrpc,sampl3,by="qname")
sampl1<-unique(sampl1)
return(sampl1)
}



Reads<-sampling(s)
###sauvegarde de l'échantillon####
Mes_reads<-subset(Reads,select=c(qname))
write.table(Mes_reads ,file=nom,quote=FALSE,row.names=FALSE,col.names=FALSE)

q()
```

2. Scripts réalisé avec bash (et awk)

```
############################
#####SamplingBamfile.sh#####
#!/bin/bash
if [ $# -lt 0 ]
then
   echo ""
   echo "Usage: [fichier bam]  [Nombre de lectures ] [Nom du fichier bam de sortie ]"
   echo ""
   echo "example : samplingBamfile.sh <File.bam> <1000> <output_file_name> "
 exit 1
fi
########################################
# Creation de nom pour les fichiers de sortie        #
########################################
tete=$(echo $1 | sed 's/.bam*$//g' )_$2
head=$(echo $1 | sed 's/.bam*$//g' )
if [ "$3" == "" ]
then
output=$tete
else
output=$3
```

```
fi

##########################################
# Verifier si les fichiers de sortie existent déjà          #
##########################################
if  [ -e $output.sorted.bam  ]
then
echo " File already exist"
exit
fi
#############################################################
#  Creation de nom pour le fichier de sorte générés  par R                  #
#############################################################
R_file1=$head.1.Rda
R_file2=$head.2.Rda
R_file3=$head.3.Rda
################################################
#          Execution du script R                                    #
################################################

if  [ ! -e $R_file1 ] && [ ! -e $R_file2 ] && [ ! -e $R_file3 ]
then
R_surf --vanilla --args bamfile "$1" h $2 nom "$output.txt" file1 "$R_file1" file2 "$R_file2" file3
"$R_file3"    < sampling_script_1.R
else
R_surf --vanilla --args s $2 nom "$output.txt"  file1 "$R_file1" file2 "$R_file2" file3 "$R_file3"    <
sampling_script_2.R
fi


################################################
#         Conversion du fichier Bam en SAM                  #
################################################
while  [ ! -e $head.sorted.sam ]
do
samtools view -h -o  $head.sam $1
sort -k 1,1 $head.sam > $head.sorted.sam # mettre en ordre  le fichier sam
rm $head.sam
done
#############################################################
```

```
# Création du fichier Bam                                                    #
#############################################################
echo " Creating output bam file ....."

samtools view -H $1  > $output.head.sam  #Recuperation des en-tête du fichier bam
join -t $'\t' $head.sorted.sam $output.txt | uniq  >> $output.head.sam  # Jointure entre le fichier SAM
et le fichier R comportant les identifiants des lectures échantillonnées
samtools view -bS $output.head.sam > $output.bam # Conversion du fichier précédant en SAM
samtools sort $output.bam $output.sorted
################################################
## Suppression de queleques fichiers generes              #
## au cours de l'execution du progrramme                  #
################################################
rm $output.bam
rm $output.head.sam
rm $output.txt
###########################################
if [ -e $output.sorted.bam ]
then
echo "Bam file is created"
fi
```

##

#######SamplingBamfile_categories.sh######

######échantillonnage de plusieurs catégories pour plusieurs itérations####

```
#!/bin/bash

if [ $# -lt 3 ]
then
   echo "Usage: [bamfile] [categories] [iterations] "
   echo   "   _____________________________     "
   echo "Example: categories.sh <File.bam> <12000> <2> "
exit
fi


####iteration####
d=${@: -1}
###dernier categorie##
a=$(($# - 2))
```

```
for i in ${@:2:$(($a))}
do
header=$(echo $1 | sed 's/.bam*$//g' )_
outputname=$header$i
for ((j=1;j<=$d;j++))
do
#samplingBamfile.sh $1 $i  $outputname.$j


fichier_sbatch=tmp.$j.$i.sh

if [ ! -e $fichier_sbatch ]
then
echo $fichier_sbatch
echo "#!/bin/bash" > $fichier_sbatch
echo "samplingBamfile.sh $1 $i $outputname.$j" >> $fichier_sbatch
fi
sbatch -p debug $fichier_sbatch

rm tmp.$j.$i.sh
done
done
```

##############################

####Calculating_coverage.sh

#####Calcul de la profondeur de lecture d'un fichier bam####

```
#!/bin/bash
if [ $# -ne 2 ]
then

   echo "Usage: [fichier bam ] [Taille de la zone capturée] "
   echo"-----------------------------------------------------"
   echo " "

 exit 1
fi
heure=$(set $(date);echo $4 | awk '{gsub(/:/,"")}; 1')
```

```
samtools  view $1 | awk '{print length($10)}' >> tmp.$heure.txt
length=$(echo `awk '{sum+=$1} END {print sum }' tmp.txt`)
awk 'BEGIN{print '$length'/'$2' }'
rm tmp.$heure.txt
```

####################################

########Coverage_categories.sh######

####Cacul de la profondeur de lecture de plusieurs catégories d'échantillon###

```
#!/bin/bash
if [ $# -lt 3 ]
then
   echo "_____________________________________________________"
  echo "Usage: Main_SampleBamFileByCategory.sh [fichier bam] [taille de la zone capturée] [liste des categories]"
    echo "Usage: samplingBamfile_categories.sh test.bam 200000 250000"
    echo "This will sample 2 times the test .bam for 1, 10 and 100 reads"
  exit 1
fi

##############################################
# Creation de nom pour les fichiers de sortie              #
##############################################
heure=$(set $(date);echo $4 | awk '{gsub(/:/,"")}; 1')
head=$(echo $1 | sed 's/.bam*$//g' )_

#################################################################
#  Creation d'un fichier vide qui sera après fichie de sortie conteant toutes #
#  Toutes les profondeurs de lectures                                                    #
#################################################################
echo -e "Number of reads\t\t Depth of coverage" >> All.coverage.$head$heure.txt
#######################################################
######Cacul de la profondeur de lecture pour les  échantillons #
#######################################################

d=${@: -1}  #Dernière catégorie d'échantillon#
for i in  ${@:3:$(($d))}
do
for j in  1 2 3 4 5 6 7 8 9 10
```

```
do
if  [  -e $head$i.$j.sorted.bam ]
then
echo Bam file exist
echo Calculating coverage....
caculating_coverage_1.sh  $head$i.$j.sorted.bam $2  | xargs -I cov echo -e  "$i\t\t\t cov" >>
All.coverage.$head$heure.txt
echo Done
fi
done
```

Algorithme de coverage_for_categories.sh

Entrée : [Fichier Bam], [taille de la zone capturée], [liste des nombres de lectures]

- Si le nombre d'arguments est strictement inférieur à 3
  - Affichage de l'usage du programme
  - Arrêt de l'exécution
- Fin si
- Création d'un fichier texte vide
- D←dernier nombre de lectures saisi
- Boucle 1 : Pour chaque valeur de [liste des nombres de lectures]
  - Boucle 2 : Tant que compteur j≤15
    - Si le fichier Bam échantillonné correspondant existe alors
      - **Calculating_coverage.sh** [fichier Bam échantillonné] [taille de la zone capturée]
      - Redirection du résultat dans le fichier texte vide créé
    - Fin Si
  - Fin Boucle 2
- Fin Boucle 1

Afficher : « le fichier de sortie est créé »

INDEX

[1] David Sims, Ian Sudbery, Nicholas E. Ilott, Andreas_Heger Chris P. Ponting : *Sequencing depth and coverage: key considerations in genomic analyse*

[2] Subramanian S. Ajay, Stephen C.J. Parker, Hatice Ozel Abaan,Karin V. Fuentes Fajardo and Elliott H. Margulies :*Accurate and comprehensive sequencing of personal genomes*

[3] Ying Wang, Noushin Ghaffari, Charles D Johnson, Ulisses M Braga-Neto, Hui Wang, Rui Chen and Huaijun Zhou : *Evaluation of the coverage and depth of transcriptome by RNA-Seq in chickens*

[4] Professeur Lafleur : Le séquençage d'un ADN. Disponible sur : http://www.snv.jussieu.fr/vie/dossiers/sequencage/sequence.htm

[5] Documentation de samtools. Disponible sur : http://samtools.sourceforge.net/

[6] Documentation IGV : http://www.broadinstitute.org/igv/

[7] An introduction to Rsamtools .Disponible sur : http://www.bioconductor.org/packages/release/bioc/vignettes/Rsamtools/inst/doc/Rsamtools-Overview.pdf

[8] Cours de R http://www.statmethods.net/

[9] Documentation GATK .Disponible sur : https://www.broadinstitute.org/gatk/

[10] Introduction à la programmation bash .Disponible sur http://aral.iut-rodez.fr/fr/sanchis/enseignement/bash/

[11] Cours sur la commande. Disponible awk sur : http://www.shellunix.com/awk.html

[12] Les commandes Unix .Disponible sur : http://fr.wikipedia.org/wiki/Commandes_Unix

[13] Programmation R, quelques exemples .Disponible sur : http://fr.wikibooks.org/wiki/Programmation_statistiques_avec_R/Quelques_exemples