TD DNA-Seq - Etude d'un pipeline d'appel de variants constitutionnels

Ghislain Bidaut, Plateforme Cibi, CRCM, Aix-Marseille Université

24/03/2022

1 Introduction

Ce TD consiste en l'étude d'un pipeline de détection de variants constitutionnels.

Pour la mise en place pratique de ce pipeline, nous allons analyser des données générées par Niktin AG *et al*: Lynch Syndrome Germline Mutations in Breast Cancer: Next Generation Sequencing Case-Control Study of 1,263 Participants. *Front Oncol*. 2020; 10: 666.

Publication: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7273971/ (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7273971/)

Ces données représentent **des séquences d'ADN** recueillies chez des patients atteint du syndrome de Lynch.

C'est une maladie associée à un fort risque de développer un cancer, notamment un cancer du sein. Cette maladie est associée à la présence de mutations constitutionnelles dans les gènes du pathway *MMR*. Niktin *et al.* ont donc profilé 711 patients atteints d'un cancer du sein héréditaire, 60 avec un cancer du sein sporadique, et 492 donneurs sains pour comprendre le rôle des mutations génétiques dans l'apparition de ce syndrome.

En particulier, nous allons **détecter et identifier** les **SNPs (Single Nucleotides Polymorphisms)** et **INDELS** (Insertion-délétions de petites taille) grâce aux séquences d'ADN.

Les données décrites dans cette publication sont disponibles sur le serveur Européan Nucleotide Archive (ENA) à l'URL

https://www.ebi.ac.uk/ena/browser/view/PRJNA588789
(https://www.ebi.ac.uk/ena/browser/view/PRJNA588789)

Ce sont des **données de séquençage ciblé** sur les gènes du pathways *MMR*: **MLH1, MSH2, MSH6, EPCAM, and PMS2** avec un séquenceur Illumina (Capture).

Pour des raisons de rapidité d'analyse nous allons analyser uniquement les données pour 4 patients. Ce sont des données de type *paired-end* (PE) donc 2 fichiers FASTQ par patient.

1.1 Données

Nous n'avons à disposition que des données **patients**; les auteurs n'ayant pas mis à disposition les données contrôles:

SRA	Library Name	Strand	File
SRR10426968	323RMG_S18	Forward	SRR10426968_1.fastq.gz
SRR10426968	323RMG_S18	Reverse	SRR10426968_2.fastq.gz
SRR10426969	322RMG_S17	Forward	SRR10426969_1.fastq.gz
SRR10426969	322RMG_S17	Reverse	SRR10426969_2.fastq.gz
SRR10427737	179RMG_S23	Forward	SRR10427737_1.fastq.gz
SRR10427737	179RMG_S23	Reverse	SRR10427737_2.fastq.gz
SRR10427738	436contrS38	Forward	SRR10427738_1.fastq.gz
SRR10427738	436contrS38	Reverse	SRR10427738_2.fastq.gz

Les données sont disponibles directement sur le cluster IFB (**IFB-Core**) dans le chemin suivant:
`/shared/projects/form_2022_09/data/DNA-seq_PRJNA588789/`

2 Environnement informatique

2.1 Programmes à installer

- Se connecter au cluster **IFB-Core**.
- Copier l'arborescence de travail dans votre **répertoire personnel**:
  `cp -rp /shared/projects/form_2022_09/data/DNA-seq_PRJNA588789 ~`
- Activer conda (environnement `base`) puis créer et activer l'environnement `dnaseq`
- Installer les utilitaires suivants dans cet environnement:
- **bwa**
- **samtools**
- **trimmomatic**
- **picard**
- **fastqc**, **multiqc**
- **r-ggplot2**
- **snpeff**

- **r-base** version **4.1.0**

Installation de **GATK**:

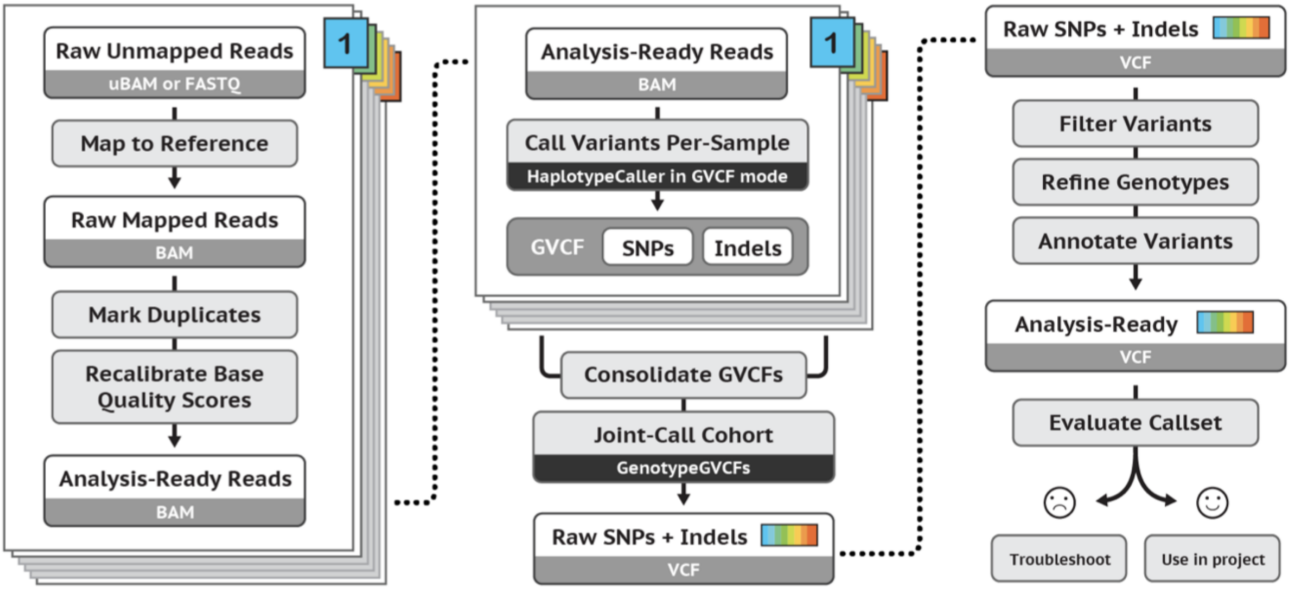
GATK est déjà téléchargé dans le répertoire `/shared/projects/form_2022_09/data/gatk-4.2.2.0` .

Il faut ajouter le chemin de **GATK** dans la variable d'environnement **PATH** (fichier `.bash_profile`).

2.1.1 Solution:

3 Structure du pipeline d'appels de SNPs et d'INDELS en constitutionnel.

Nous allons utiliser le pipeline suivant, issu **des bonnes pratiques définies par le Broad Institute** et mises en place sur leurs pipelines de production.



Pipeline SNP INDELS Germline

L'ensemble de ce pipeline est documenté sur le site du **Broad Institute**: https://gatk.broadinstitute.org/hc/en-us/articles/360035535932-Germline-short-variant-discovery-SNPs-Indels- (https://gatk.broadinstitute.org/hc/en-us/articles/360035535932-Germline-short-variant-discovery-SNPs-Indels-)

Nous allons y **ajouter les étapes supplémentaires suivantes**:

- **Contrôle qualité** avec les outils `fastqc` et `multiqc`
- **Trimming** avec l'outil `Trimmomatic`

4 Etapes préliminaires

4.1 Contrôle qualité Fastqc

TD:

1. Faire le contrôle qualité des données avec `fastqc` .

Le script `fastqc_all.sh` exécute `fastqc` .

```
./fastqc_all.sh
```

2. Grouper les résultats sous un rapport unique avec `multiqc` (On peut lancer `multiqc` une fois les jobs précédent finis).

3. Puis ouvrir le fichier `multiqc.html` avec Firefox et faire un bilan de la qualité des données, en particulier:

- Quelle information a-t-on à disposition ?
- Que peut-on dire de ces échantillons ?

4.2 Trimming

De manière à supprimer les séquences adaptatrices et assurer la qualité de nos séquences, nous allons **opérer un trimming** à l'aide de l'utilitaire **Trimmomatic**.

TD:

Le script `trim_all.sh` exécute un appel **Trimmomatic** sur l'ensemble des fichiers Fastq.

1. Pouvez vous décrire exactement ce qui est trimmmé à partir de la documentation de Trimmomatic ?
2. Exécuter le trimmming.

5 Alignement

5.1 Alignement sur génome de référence avec BWA.

Nous utilisons l'utilitaire **BWA**, un aligneur spécifiquement conçu pour le NGS et adapté au DNA-seq.

Dans le cadre de son intégration dans un pipeline bioinformatique, BWA fonctionne en 2 étapes:

- **Génération d'un index du génome de référence**. A cette étape, l'utilisateur doit apporter la séquence du génome de référence. **BWA** va générer les indexes qui seront utilisés lors de la seconde partie, l'alignement proprement dit. Cet index a besoin d'être généré une seule fois et peut être réutilisé.
- **L'alignement sur le génome**. A cette étape, l'utilisateur doit apporter les indexes générés, les séquences à aligner et le programme va aligner les fragments et générer un fichier BAM.

5.2 Récupération du génome et données d'annotation

Nous avons déjà récupéré les fichiers du génome de référence **hg38** à partir du site de **GATK** du **Broad Institute**, mis à disposition ici: https://console.cloud.google.com/storage/browser/genomics-public-data/resources/broad/hg38/v0;tab=objects?pli=1&prefix=&forceOnObjectsSortingFiltering=false (https://console.cloud.google.com/storage/browser/genomics-public-data/resources/broad/hg38/v0;tab=objects?pli=1&prefix=&forceOnObjectsSortingFiltering=false)

Il faut récupérer le fichier **fasta** mais aussi les indexes associés (Fichiers **dict**, **alt**, **amb ann**, **bwt**, **pac**, **sa** et **fai**).

Ces données sont dans le répertoire `hg38.genome` .

De plus, nous allons filtrer les alignements sur les gènes utilisés lors de la capture. Pour cela, il faut spécifier les régions séquencées grâce à un fichier **BED**.

Il est possible de créer ce fichier sur l'outil **hgTables** du site de l'**UCSC**: https://genome.ucsc.edu/cgi-bin/hgTables (https://genome.ucsc.edu/cgi-bin/hgTables). Pour les besoins de ce TD, il a déjà été créé et est présent dans le fichier: `nikitin_et_al.bed` .

TD:

1. Analyser le script `bwa_align_all.sh` :

- Déterminer les outils utilisés et leur fonction.
- Déterminer les différentes étapes de ce script et les différents fichiers générés.

2. Exécuter l'alignement
3. Refaire un point qualité avec **multiqc**.

6 Réalignement par GATK

Nous venons de faire un premier alignement général rapide. Hors, cet alignement ne **suffit pas** pour détecter les **INDELs**.

Les prochaines étapes consistent à préparer les échantillons pour faire tourner **GATK**.

6.1 Marquer les duplicats et les groupes

Pour la suite, nous allons effectuer une étape technique en deux parties:

1. **marquage de duplicats**
2. **ajout d'identifiants de groupe** avec l'outil **picard**.

Le marquage des duplicats ainsi que l'ajout des identifiants de groupe par **picard AddOrReplaceReadGroups** permet d'identifier les variables confondantes techniques afin d'améliorer les tests statistiques permettant d'identifier les variants.

TD:

1. Exécuter le script `mark_duplicate_all.sh` .
2. S'en inspirer pour concevoir le script `add_readgroups_all.sh` . (Aide: ajouter les options suivantes à **picard**: `–RGID 4 –RGLB lib1 –RGPL ILLUMINA –RGPU unit1 –RGSM 20`).

6.1.1 Solution

6.2 Recalibration de la qualité des bases

Cette étape consiste à **corriger la qualité des bases** en abaissant la qualité des bases si on s'éloigne de la séquence de référence en étant hors des sites des polymorphismes.

Il faut donc utiliser les bases de données publiques pour cette étape.

Nous allons utiliser les données provenant des bases suivantes:

- **dbSNP**. Base de référence de l'ensemble des polymorphysmes et INDELS
- **1000 genomes**. Base de référence des sites polymorphiques de sujets sains.
- **Omni 1000 genomes**. Base de référence des sites polymorphiques de sujets sains revalidés sur la plateforme Omni par le Broad Institute.
- **hapMap**: Base de donnés d'haplotypes décrivant les patrons communs des variations génétiques du génome humain.

Les fichiers correspondant à ces bases ont été rendus disponibles par les auteurs de **GATK**: https://console.cloud.google.com/storage/browser/genomics-public-data/resources/broad/hg38/v0;tab=objects?pli=1&prefix=&forceOnObjectsSortingFiltering=false (https://console.cloud.google.com/storage/browser/genomics-public-data/resources/broad/hg38/v0;tab=objects?pli=1&prefix=&forceOnObjectsSortingFiltering=false)

En pratique, cette correction se fait par deux appels successifs à **GATK**. Le premier appel est un appel à **GATK Recalibrator**, qui va calculer les changements à appliquer sur les fichiers **BAM**. Le second consiste à appliquer ces changements par un appel à **GATK ApplyBQSR**.

TD:

1. Lire la documentation correspondant à chaque outil:

- **BaseRecalibrator**: https://gatk.broadinstitute.org/hc/en-us/articles/360050815072-BaseRecalibrator (https://gatk.broadinstitute.org/hc/en-us/articles/360050815072-BaseRecalibrator)
- **ApplyBSQR** https://gatk.broadinstitute.org/hc/en-us/articles/360037225212-ApplyBQSR (https://gatk.broadinstitute.org/hc/en-us/articles/360037225212-ApplyBQSR)

2. Modifier le script `base_recalibrate_all.sh` pour

2.1. Bien utiliser les **4 ressources** cités précédemment dans l'étape **BaseRecalibrator**. 2.2. Ajouter l'étape **ApplyBSQR** et enregistrer les fichiers **BAM** recalibrés dans `*.recalibrated.bam` .

6.2.1 Solution

7 Appel de variants

Cette dernière étape consiste **à détecter les SNPs et INDELs**.

7.1 Appel de variants par échantillons

Maintenant que les fichiers **BAM** sont prêts, nous allons faire un appel de variants par échantillons et générer des fichiers **GVCF**.

Cela se fait par **GATK HaplotypeCaller**. Cet outil effectue une **Réalignement local** pour optimiser la recherche d'INDELS et renvoie un fichier **GVCF** et un fichier **BAM** contenant les **séquences réalignées**. C'est donc une opération **très coûteuse en temps de calcul**.

TD:

1. Lire la documentation de **GATK HaplotypeCaller**.
2. Prendre modèle sur l'un des scripts précédents pour créer un script appelé `variantcall_gvcf.sh` et l'utiliser pour faire l'appel de variants sur les fichiers `*bam.recalibrated` .

Attention: spécifier l'option suivante: `gatk --java-options "-Xmx16g" HaplotypeCaller...`

7.1.1 Solution

7.2 Consolidation des GVCFs en un fichier VCF multiéchantillons.

- **Etape 1:** Nous allons grouper les différents fichiers GVCFs dans une base de données. Pour cette étape, nous utilisons l'outil **GATK GenomicsDBImport**. Il prend en entrée:
    - Une **table de correspondance** des échantillons et des fichiers GVCF: `gvcf_sample_map.txt` .
    - Un fichier contenant les régions à analyser (appelées *intervalles*) généré par **picard BedToIntervalList**.
- **Etape 2:** Il faut ensuite convertir la base de données de variants en un fichier **VCF multi-échantillons**. Pour cela, nous faisons appel à **GATK GenotypeGVCFs**.

TD:

Le script effectuant cette opération est déjà prêt: `genomicsdbimport.sh` .`

1. **Se familiariser avec ce script** et répondre aux questions suivantes:.

- Où se trouve la base de données générées ?
- Quel est le nom du fichier VCF généré ?

2. L'exécuter

7.3 Filtrage des variants

Nous allons maintenant filtrer les SNPs/INDELS à l'aide de l'algorithme Variant Quality Score Recalibration (VQSR). Cet algorithme utilise les bases de données publiques pour attribuer un **score** aux variants. L'algorithme base ce score entre l'annotation de ce variant et la probabilité que ce soit un variant véritable ou simplement un artéfact de séquençage. Une fois attribué, ce score pourra ensuite être utilisé pour **filtrer** les variants.

Les bases sont présentes dans le sous répertoire `ressources` . ce sont les mêmes qui ont servi à recalibrer les scores qualité des reads.

Le filtrage se fait en deux étapes:

- **Etape 1**: Nous allons calculer les scores à l'aide de **GATK VariantRecalibrator**.
- **Etape 2**: Nous allons appliquer ces scores et régénérer un fichier **VCF** à l'aide de **GATK ApplyVQSR**.

TD:

Le script de filtrage des variants existe déjà (`filter_variants.sh`) mais ne comporte que l'appel à **GATK VariantRecalibrator**.

1. **Ajouter** un appel à **GATK ApplyVQSR** en tenant compte des sorties de **GATK VariantRecalibrator** et en spécifiant une sensibilité de 0.95.
2. **Exécuter** le script.

7.3.1 Solution

8 Visualisation sous IGV

Lors de cette dernière étape, nous allons **examiner les résultats avec l'outil graphique IGV**.

TD:

1. Exécuter IGV et sélectionner le génome correspondant à nos données.
2. Importer les fichiers suivants dans IGV:

- Les fichiers BAM issus du **réalignement GATK**.
- Le fichier **VCF** final avec l'ensemble des échantillons.
- Ensuite, se placer sur le gène **MLH1**. Identifier les **SNPS et INDELSs**.

8.0.1 Solution