essai-format

Elise GUERET

25 Mai 2018

Table of Contents

[Remerciements 2](#_Toc517869608)

[Avant-propos 2](#_Toc517869609)

[Table des matières 2](#_Toc517869610)

[Liste des tableaux et des figures 2](#_Toc517869611)

[Liste des abbréviations 2](#_Toc517869612)

[Résumé et mots-clés (en français et en anglais) en 4ème de couverture 2](#_Toc517869613)

[Introduction 2](#_Toc517869614)

[Problématique - Objectifs 3](#_Toc517869615)

[Matériels et méthodes 3](#_Toc517869616)

[Design de l’étude 3](#_Toc517869617)

[Vue générale des pipelines 4](#_Toc517869618)

[Description des pipelines 4](#_Toc517869619)

[Correction Annotation par Python 4](#_Toc517869620)

[Pipeline de génotypage 5](#_Toc517869621)

[Résultats - Discussion 11](#_Toc517869622)

[Résultats du génotypage 11](#_Toc517869623)

[Résultats de l’étude d’association 11](#_Toc517869624)

[Fréquence allèlique et données manquantes 11](#_Toc517869625)

[Estimation de l’“ancestry” 11](#_Toc517869626)

[POpulation stratification 11](#_Toc517869627)

[Top SNPs 11](#_Toc517869628)

[Evaluation des top SNPs 11](#_Toc517869629)

[Analyse fonctionnelle des top SNPs 11](#_Toc517869630)

[Conclusion 11](#_Toc517869631)

[Options de blocs de code R en R Markdown : 12](#_Toc517869632)

[References 12](#_Toc517869633)

header includes = Pour avoir des noms automatiques d???éléments en français lors de la création d???un document final au format PDF (par exemple le titre de la table des matières)

Insertion de saut de page Word : <https://datascienceplus.com/r-markdown-how-to-insert-page-breaks-in-a-ms-word-document/> # Page de couverture

* les logos
* intitulé “Mémoire de stage de fin d’études”
* Elise GUERET
* Nom des tuteurs
* adresse de la fac
* adresse du stage
* année universitaire en cours
* titre du mémoire

# Remerciements

# Avant-propos

## Table des matières

## Liste des tableaux et des figures

## Liste des abbréviations

# Résumé et mots-clés (en français et en anglais) en 4ème de couverture

# Introduction

Question à se poser avant la rédaction du rapport :  
\* Quels étaient les objectifs de mon travail?  
\* Quel est le bilan de mon travail?  
\* Quelles sont les informations essentielles et les informations secodaires (mais nécessaires) relatives à mon travail?  
\* Comment organiser ces informations pour les rendre compréhensible à un tier?  
Definir le sujet en termes précis et concis, énoncer les objectifs du travail personnel et les moyens mis en oeuvre et présenter le plan adopté pour la suite du sujet d’étude.

Replacer le sujet d’études dans un contexte plus général. Résumer l’état de l’art via une étude des travaux antérieurs.

Décrire les différents projets européens (RobustBass et CRECHE 2016). Dire à quoi ils servent. Comment les données ont été obtenues? Pour CRECHE voir le rapport dans les archives ifremer et pour RobustBass voir les diapos KoM.

Commencer par parler de Dicentrarchus labrax, de l’aquaculture et du projet CRECHE 2016. Parler ensuite des SNPs et des GWAS (définition et existants). Parler des NGS et plus particulièrement du RNA-seq.

# Problématique - Objectifs

Objectifs : Génotyper les individus considérés, trouver les SNPs liés à une lignée et corriger l’annotation Présentation du cahier des charges : correction de l’annotation, mise en place d’un script pour la découverte de variants à partir de RNA-seq et faire des analyses d’association.

# Matériels et méthodes

Mettre en valeur l’acquisition d’une technique, d’un savoir-faire enrichissant mes connaissances initiales. Décrire la méthode ou le logiciel. Pour la correction de l’annotation : Mise en place de scripts Python. Pour le Variant Calling : script en langage wdl (workflow description language) et les suites d’outils Picard et GATK (Genome Analysis ToolKit). Sans oublier les visualisation à l’aide de R. L’analyse d’association PLINK et rda.

## Design de l’étude

Description des données : comment elles ont été obtenues? Description des étapes (annotation de la puce : pourquoi?, GATK pour rechercher les SNPs, snpEff pour annoter les SNPs, PLINK pour faire calcul LD et Association)

Pour accomplir les objectifs mentionnés ci-dessus, les étapes suivantes ont été réalisées: développer un pipeline pour corriger l’annotation du génome, développer un pipeline pour extraire les génotypes des individus et réaliser les analyses d’association par famille et d’association génotype/phénotype.

Les données de RNA-seq sont issus du Projet CRECHE 2016. Les ARN ont été extraits à partir de foie prélévés sur des individus sacrifiés appartenant aux quatres lignées suivantes J-/J+ (résistant ou non au jeûne) ou C-/C+ (divergence de croissance). Les étapes préliminaires au séquençage ont été réalisées par la plateforme GenSeq. Ces étapes sont : la conservation des échantillons, l’extraction des ARN ainsi que les contrôles qualités des ces ARN. La plateforme GET (Génomique et Transcriptomique) a quant à elle réalisée la création des banques d’ADN complémentaires ainsi que le contrôle qualité des ses banques. Elle a également réalisée le séquençage et les pré-traitement bioinformatiques. L’obtention des reads a été effectué en paired-end 150 sur 5 lignes de séquenceur Illumina HiSeq3000. Parmi les 65 fichiers d’alignement, 24 ont été sélectionnés pour mener cette étude. Ces fichiers proviennent de 23 individus différents (1 individus est en duplica). Chaque lignée est représentée par 6 individus. Les phénotypes disponibles pour ces individus sont : la lignée, le sexe, l’EAi moyen, la classe de l’EAi ainsi que leur comportement.

Le génome de référence de *Dicentrarchus labrax* utilisé das cette étude est le génome qui a été publié. Assemblage jusq’à contigs incomplets plus ou moins longs.

L’annotation du génome de référence de *Dicentrarchus labrax* utilisée est une annotation plus complète que celle qui a été publiée puiqsue des données de transcriptome y ont été ajoutés.

Une puce à ADN contenant 57000 SNPs est en cours de fabrication. Pour ces SNPs seule leur position sur le génome est connu. Il serait utile de déterminer les effets de ces SNPs au sein du génome de *Dicentrarchus labrax*.

## Vue générale des pipelines

Que font-ils ? Quels outils ou logiciels ils utilisent?

Les données d’alignement proviennent d’une analyse bioinformatique qui a été réalisée par la plateforme MBB. Cette analyse comprend deux étapes : l’élimination des séquences de mauvaise qualité par TRIMMOMATIC et l’alignement des reads sur le génome de *Dicentrarchus labrax* par l’outils ou le logiciel HISAT2. Une seconde analyse bioinformatique est nécessaire pour effectuer une recherche de variants. Cette analyse consiste à réaliser une recalibration des reads alignés (Picard et GATK) afin de générer des fichiers BAM qui sont le point de départ de cette étude.

Pour étendre et complémenter l’analyse de ces données, un pipeline complémentaire a été programmé. Il utilise les outils suivants :  
\* Snpeff;  
\* VCFtools;  
\* PLINK;  
\* Package R qqman;  
\* Package R ggplot2;  
\* Package R vegan.

## Description des pipelines

**Penser à impliquer la méthodo c’est-à-dire pour GATK : de comparer les réplicas ainsi que de voir la différence avec des petits bam (peu de reads) et des gros bam (bcp de reads) pour voir si on trouve les mêmes SNPs ou si il y a une différence notable.** A quoi ils servent?

Mettre un schéma général des principales étapes.

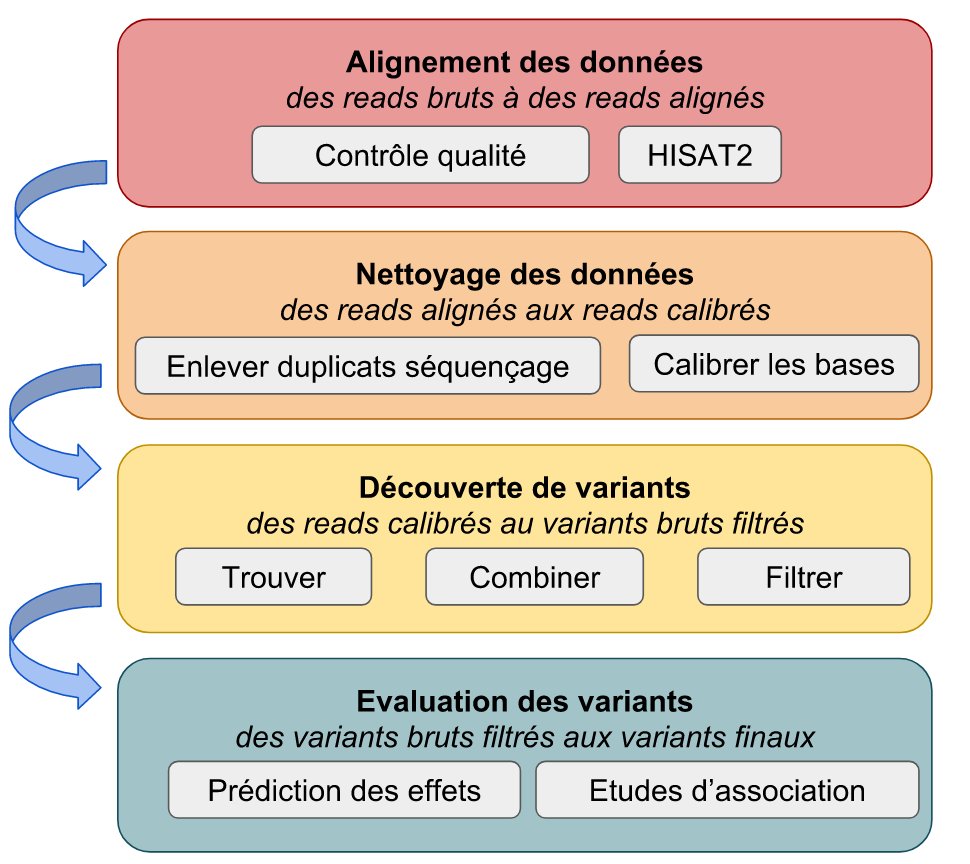
Les avancées technologiques en séquençage à haut-débit et les outils bioinformatiques qui en découlent rendent l’identification et la quantification de variants de plus en plus facile. Cependant, le génotypage de ces variants reste un challenge car chaque outils de recherche de variants est basé sur son propre algorithme de détection de variants. Pour un même jeu de données, il est donc possible d’obtenir des variants différents en fonction de l’algorithme choisi. Il faut donc que l’utilisateur soit conscient que les variants obtenus par un outils peuvent ne pas être retrouvé lors de l’utilisation d’un autre outils. Ceci a été pris en compte lors de l’ecriture de pipeline de découverte de variants utilisant GATK, par le respect des [bonnes pratiques de découverte de variants à partir de RNA-seq](https://software.broadinstitute.org/gatk/documentation/article?id=4067) suggérer par l’équipe développant GATK au Broad Institute.

### Correction Annotation par Python

Le pipeline python permet de corriger l’annotation du génome existant. Il utilise des packages biopython. Détails de chaque étape + Schéma

### Pipeline de génotypage

Ce pipeline est décomposé en 4 phases distinctes qui utilisent différentes resources et qui permettent de faire de la parallèlisation pour optimiser les temps de calculs. Ces 4 phases sont les suivantes :  
\* Alignement des données (réalisé par l’équipe de la plateforme MBB);  
\* Nettoyage des données;  
\* Découverte des variants;  
\* Evaluation des variants.

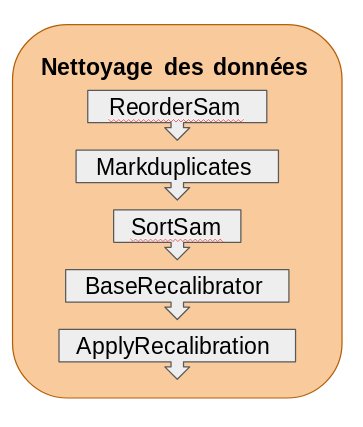


Vue générale du pipeline de génotypage

Ce pipeline est une collection de script wdl, bash et json à exécuter dans l’ordre sur une machine Linux avec une gestion des jobs de type gridengine comme un cluster de calcul. Il est également exécutable sur une machine Linux sans gestion des jobs tels qu’une station de travail.

Pour rendre les données analysables pour la découverte de variants quelques étapes des phases précédentes nécessaires sont détaillées ci-après.

#### Nettoyage des données

Le nettoyage des données est une phase nécessaire pour pouvoir réaliser la suite des analyses. En effet, elle permet d’obtenir des reads calibrés nécessaire à la phase 3. Cette phase comprend des étapes réalisées par Picard ou par GATK. 

La première étape de cette phase appelée “ReorderSam” est réalisée par Picard. Cette étape range les reads en fonction de leur appartenance à un groupe de liaison dans l’ordre des contigs du génome de référence. En effet, si les contigs ne sont pas rangés dans le même ordre cela pose problème à GATK. C’est pour pallier à ce problème que l’on réalise cette étape. Elle est notamment nécessaire lorsque l’alignement a utilisé un ordre différents des groupes de liaisons.

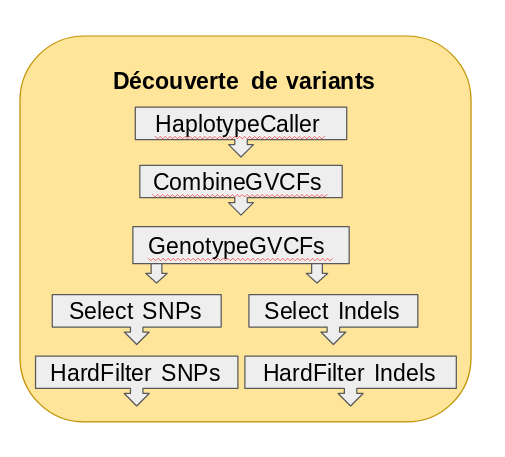
La seconde étape nommée “Markduplicates” utilise Picard. Cette étape est importante car elle marque puis élimine les duplicats de séquençage. Ces duplicas sont des artéfacts dû à la PCR d’enrichissement des banques qui est réalisée en amont du séquençage. Ils sont retirés pour éviter que ces reads soient considérés comme des morceaux de transcrits plus représentés que d’autres.

La troisième étape intitulée “SortSam” est également réalisée par Picard. Cette étape range de nouveau les reads car certains ont été supprimés (masqués) par l’étape précédente. Cela permet d eles mettre en fin de fichier pour ne pas qu’il soit pris en compte dans l’étape qui suit.

Les deux étapes qui vont suivre sont nécessaires car les algorithmes de découverte de variants utilisent la qualité des bases de chaque reads pour déterminer s’il s’agit bel et bien d’une variation ou d’un artéfact. En effet, les scores de qualité sont dépendants du nombre de reads présents. Or certains reads considérés comme des duplicats de séquençage ont été retirés. Il ne faut donc plus les prendre en compte. D’autre part, à ces scores de qualité sont appliqués des erreurs systématiques qui doivent être recalibrés. POur faire cela GATK utilise un algorithmle de “machine-learning” pour modeller empiriquement ces erreurs et ajuster les scores de qualité. Ces deux étapes sont “BaseRecalibrator”et “ApplyBQSR”. La première est réalisée 2 fois afin de visualiser l’efficacité de la recalibration des reads. Cette étape crée une table qui sera utilisée pour la recalibration. La seconde (“ApplyBQSR”) utilise donc la table de recalibration produite par l’étape précédente pour effectuer la recalibration de chacune des bases de chaque reads.

La dernière étape nommée “AnalyzeCovariates” eégalement effectuée par GATK, produit des graphiques montrant l’efficacité de cette recalibration.

#### Découverte de variants

Cette phase est entièrement réalisée par GATK et est la phase clé puisque c’est elle qui extrait les génotypes de chacun des individus. Le succès de cette phase dépend à la fois de la minimisation des Faux positifs et à la fois des Faux négatifs. POur faire cela, GATK procède en plusieurs étapes: la découverte de variants (étape “HaplotypeCaller”) par individus, la fusion des génotypes de chaque individus (étapes “CombineGVCFs” et “GenotypeGVCFs”) par lignée puis la filtration des variants par type de variations (SNPs ou INDELs). Les deux premières étapes ont été designés pour maximiser la sensibilité alors que la filtration permet de maximiser la spécificité par un choix de filtre adaptables à chaque jeu de données. Bien entendu, la découverte de variants dépends du type de données (Whole genome, transcriptome, exome, etc.) mais aussi de d’autres paramètres provenant du séquençage comme la couverture ou la profondeur de séquençage. 

La première étape “HaplotypeCaller” réalise en même temps la recherche de SNPs et d’INDELs par un ré-assemblage *de novo* des haplotypes des séquences actives de la région ciblée. Cet outils est capable de détecter 5 types de variations différentes. [Source1](https://software.broadinstitute.org/gatk/documentation/article?id=3682) Il s’agit des SNPs, MNPs, INDELs, Mixed, et Symbolic. Les SNPs représentent des “single nucleotide polymorphism”. Les MNPs représentent les “multi-nucleotide polymorphism”. Les INDELs représentent des évènements d’insertions ou de délétions de nucléotides. Les Mixed représentent une combinaison de SNPs et d’INDELs à une seule et même position. Les Symbolic montrent qu’il se passe quelque chose à cette position mais qu’on ne sait pas exactement ce qu’il représente. Parfois, HaplotypeCaller utilise l’allèle “non ref” ou \* pour signifier la présence d’une suppression étendue ou des évènements indéfinis comme un trés grand allèle, la perte d’un seul allèle, la perte de deux allèles, etc. [Source 2](https://software.broadinstitute.org/gatk/documentation/article.php?id=6926) HaplotypeCaller produit un fichier au format gVCF (Variant Calling Format) pour chaque individus.

La seconde étape “CombineGVCFs” combine en un seul fichier VCF tous les gVCFs produits dans l’étape précédente.

La troisième étape “GenotypeGVCFs” utilise le fichier VCF de tous les gVCFs combinés de chaque individus pour faire un recalibrage des scores de qualité des variants. Cette analyse réalisée par lignée permet la détection de variants au niveau des locus complexes. Il s’agit d’une aggrégation conjointe multi-échantillons qui fusionnent les enregistrement de manière sophistiquée : pour chaque position du fichier VCF (gVCFs combinés), cet outils combinent tous les enregistrment couvrant le même position, produit des probabilités de génotype correct, re-génotype l’enregistrement nouvellement fusionné puis le ré-annote.

La quatrième étape “SelectSNPs” ou “SelectINDELs” est la sélection des variations selon leur type (SNPs ou INDELs). Cette étape va permettre de séparer les variants afin de leur appliquer des filtres.

La cinquième étape “hardfilterSNPs” ou “hardfilterINDELs” est l’application de filtres.En effet, selon les bonnes pratiques de découverte de Variants à partir de données de RNA-seq, il faut réaliser un “hard-filtering”. C’est-à-dire choisir des valeurs seuils qui définissent quelles données sont considérées comme correcte de celles qui ne le sont pas. Pour cela, ils recommandent différentes valeurs pour différents champs en fonction du type de variation (SNPs ou Indels). Ces valeurs sont présentées dans le tableau suivant :

|  |  |  |
| --- | --- | --- |
| Sigle du champs utilisé | Valeur si SNPs | Valeur si INDELs |
| QD | <2.0 | <2.0 |
| FS | >60.0 | >200.0 |
| SOR | >3.0 | >10.0 |
| ReadPosRankSum | <-8.0 | <20.0 |
| MQ | <40.0 |  |
| MQRankSum | <-12.5 |  |
| InbreedingCoeff |  | <-0.8 |

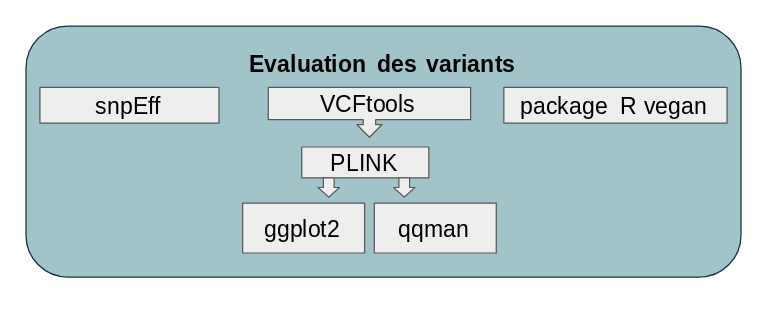
Ces différents sigles représentent différentes annotation qui signifient :

* **Quality by depth (QD)** : Cette annotation met en perspective le score QUAL de la variante en la normalisant pour la quantité de couverture disponible. Parce que chaque lecture contribue un peu au score QUAL, les variantes dans les régions avec une couverture profonde peuvent avoir des scores QUAL artificiellement gonflés, donnant l’impression que l’appel est soutenu par plus de preuves qu’il ne l’est réellement. Pour compenser cela, nous normalisons la confiance de la variante par la profondeur, ce qui nous donne une image plus objective de la qualité de l’appel.
* **Fischer’s Strand bias (FS)** : Le biais de brin est un type de biais de séquençage dans lequel un brin d’ADN est favorisé par rapport à l’autre, ce qui peut entraîner une évaluation incorrecte de la quantité de preuve observée pour un allèle par rapport à l’autre. L’annotation FisherStrand est l’une de plusieurs méthodes qui vise à évaluer s’il y a un biais de brin dans les données. Il utilise le test exact de Fisher pour déterminer s’il existe un biais de brin entre les brins aller et retour pour l’allèle de référence ou l’allèle alternatif. La sortie est une valeur p de Phred. Plus la valeur de sortie est élevée, plus il y a de risque de biais. Plus de biais indique des faux positifs.
* **Strand Odds Ratio (SOR)** : C’est une autre façon d’estimer le biais de brin en utilisant un test similaire au test de rapport de cotes symétrique. SOR a été créé parce que FS tend à pénaliser les variantes qui se produisent aux extrémités des exons. Les lectures aux extrémités des exons tendent à n’être couvertes que par des lectures dans une direction et FS donne un mauvais score à ces variantes. SOR prendra en compte les ratios de lectures qui couvrent les deux allèles.
* **ReadPosRankSum Rank Sum Test for relative positioning of REF versus ALT alleles within reads (ReadPosRankSum)** : Cette annotation au niveau des variantes teste s’il existe des preuves de biais dans la position des allèles dans les lectures qui les supportent, entre les allèles de référence et les allèles alternatifs. Voir un allèle seulement près des extrémités des lectures indique une erreur, car c’est là que les séquenceurs ont tendance à faire le plus d’erreurs. Cependant, certaines variantes situées près des bords des régions séquencées seront nécessairement couvertes par les extrémités des lectures, donc nous ne pouvons pas simplement définir un seuil absolu de «distance minimale à partir de la fin de la lecture». C’est pourquoi nous utilisons un test de somme de rang pour évaluer s’il y a une différence dans la façon dont l’allèle de référence et l’allèle alternatif sont supportés. Le résultat idéal est une valeur proche de zéro, ce qui indique qu’il y a peu ou pas de différence dans la localisation des allèles par rapport à la fin des lectures. Une valeur négative indique que l’allèle alternatif se trouve aux extrémités des lectures plus souvent que l’allèle de référence. Inversement, une valeur positive indique que l’allèle de référence se trouve plus souvent à l’extrémité des lectures que l’allèle alternatif.
* **Root Mean Square of the mapping quality of reads across all samples (MQ)** : Cette annotation fournit une estimation de la qualité de mappage globale des lectures prenant en charge un appel de variante. Il produit à la fois des données brutes (somme des carrés et nombre de lectures totales) et le carré moyen calculé. Les données brutes sont utilisées pour calculer avec précision le carré moyen en combinant plus d’un échantillon.
* **MappingQualityRankSum Rank Sum Test for mapping qualities of REF versus ALT reads (MQRankSum)**: Cette annotation au niveau des variants compare les qualités de cartographie des lectures supportant l’allèle de référence avec celles supportant l’allèle alternatif. Le résultat idéal est une valeur proche de zéro, ce qui indique qu’il y a peu ou pas de différence. Une valeur négative indique que les lectures supportant l’allèle alternatif ont des scores de qualité de cartographie inférieurs à ceux supportant l’allèle de référence. Inversement, une valeur positive indique que les lectures supportant l’allèle alternatif ont des scores de qualité de cartographie plus élevés que ceux supportant l’allèle de référence. Cette annotation peut être utilisée pour évaluer la confiance dans un appel de variante et est une covariable recommandée pour le recalibrage de variante (VQSR). Trouver une différence de qualité statistiquement significative suggère que le processus de séquençage et / ou de cartographie peut avoir été biaisé ou affecté par un artefact. En pratique, nous ne filtrons que les valeurs négatives faibles lors de l’évaluation de la qualité des variants car l’idée est de filtrer les variants pour lesquels la qualité des données supportant l’allèle alternatif est relativement faible. Le cas inverse, où c’est la qualité des données supportant l’allèle de référence qui est le plus bas (résultant en des scores de rangs positifs), n’est pas vraiment informatif pour les variantes de filtrage.
* **Inbreeding Coefficient (InbreedingCoeff)**: Cette annotation évalue s’il existe des preuves de consanguinité dans une population. Plus le score est élevé, plus le risque de consanguinité est élevé. Le calcul est une généralisation continue du test de Hardy-Weinberg pour le déséquilibre qui fonctionne bien avec une couverture limitée par échantillon. La sortie est la statistique F de l’exécution du test HW pour le déséquilibre avec les valeurs PL.

#### Evaluation des variants et études d’association

Une fois que les variants ont été annotés , il reste à vérifier que ce ne sont pas des artéfacts. Cette phase est nécessaire avant d’accepter un variant comme spécifique d’une lignée. Dans cette phases, une annotation fonctionnelle aisni que des contrôles qualités additionnels sont réalisés.

##### Design

Cette quatrième et dernière phase permet donc d’évaluer les variants en terme d’effet prédit, d’association par famille ou par phénotype/génotype. Cette dernière phase fait appel à différents outils et logiciels : SnpEff pour prédire les effets des SNPs trouvés, PLINK pour réaliser l’étude d’association par famille, le package R vegan pour réaliser l’analyse d’association génotype/phénotype, etc. 

La dernière étape de GATK (“hardfiltering”) produit deux VCFs (un par type de variations : SNPs et INDELs) qui sont les fihciers d’entrée pour les différents logiciels ou outils mentionnés précédemment.

La première étape consiste à une annotation fonctionnelle par la prédiction des effets des SNPs potentiels trouvés. Cette étape est réalisée par le logiciel snpEff qui analyse le fichier d’entrée, annote les variants et calcule les effets qu’ils produisent sur les gènes connus.

La seconde étape consiste à une analyse d’association par famille. Cette étape est consitituée de sous étape. En effet, PLINK a besoin de fichier au format ped et au format map pour procéder aux associations. Or, GATK produit des fichiers au format VCF. L’outils VCFtools est donc nécessaire pour produire les fichiers ped et map à partir de fichiers VCF.

La troisième étape consiste à une analyse d’association génotype/phénotype.

##### Population stratification

Expliquer et définir les problèmes de stratification de la population.

# Résultats - Discussion

Avoir un fil conducteur dans un ordre logique pour mettre en évidence un point précis ou un but recherché.

Discussion :  
\* Mes résultats sont-ils pertinents?  
\* Quelle est leur signification?  
\* Quelle en est la portée?  
\* Peuvent-ils être utile à d’autres?

## Résultats du génotypage

Parler de Picard et de GATK

## Résultats de l’étude d’association

### Fréquence allèlique et données manquantes

### Estimation de l’“ancestry”

### POpulation stratification

### Top SNPs

### Evaluation des top SNPs

### Analyse fonctionnelle des top SNPs

# Conclusion

Résume le travail accompli et fait apparaître si les objectifs ont été atteints

S’achève par des perspectives ou sur un bilan personnel

* Que dois-je retenir de ce travail?

etblahblah Pour la rédaction de doc en Rmd : <https://stt4230.rbind.io/communication_resultats/redaction_r_markdown/#creation-dun-document-r-markdown-en-rsudio>

# Options de blocs de code R en R Markdown :

Les options de blocs de code R les plus utiles sont les suivantes :

* eval (TRUE par défaut, ou FALSE) : détermine si le code R doit être évalué ou non,
* echo (TRUE par défaut, ou FALSE) : détermine si le code R doit être affiché ou non,
* results ('markup' par défaut, ou 'hide' ou 'hold' ou 'asis') : détermine comment les sorties doivent être affichées,
* error (FALSE par défaut, ou TRUE) : détermine si les messages d???erreur doivent être affichés.
* warning (TRUE par défaut, ou FALSE) : détermine si les messages d???avertissement doivent être affichés.

# References