université PARIS-SACLAY

FACULTÉ DES SCIENCES D'ORSAY

L3 – Double Licence Mathématiques Informatique

PREPARATION ET ANALYSE DE DONNEES GENOMIQUES

George MARCHMENT

Clémence SEBE

Projet Bioinformatique – 2021

https://github.com/George-Marchment/Projet-Bioinformatique-L3

Sommaire

I Introduction

Aujourd'hui, les données biologiques sont en pleine explosion et toujours de plus en plus complexes et volumineuses. Elles sont difficilement analysables sur papier et c'est pourquoi, les biologistes font appel à des informaticiens pour les aider dans leur travail et ainsi mieux comprendre les données qu'ils obtiennent.

La biologie est composée de plusieurs branches ; l'une d'elles est la phylogénie. La phylogénie consiste en la recherche de liens de parenté pour tracer l'évolution d'un groupe d'organismes en supposant que les individus descendent tous d'un même ancêtre.

Une équipe de biologistes nous a confié une mission. Le but de ce projet est d'analyser des génomes de levures domestiquées et naturelles pour estimer l'histoire évolutive de ces dernières.

Nous avons à notre disposition une publication de 2018 intitulée *« Multiple Rounds of Artificial Selection PromoteMicrobe Secondary Domestication – The Case of Cachaça Yeasts »*. Cet article porte sur l'étude du micro-organisme, Saccharomyces cerevisiae. Cette levure est un bon modèle pour l'étude et la compréhension de l'émergence des phénotypes sélectionnés artificiellement et nous possédons une banque de données riche et variée ainsi que de nombreuses connaissances à son sujet. Cette publication porte plus précisément sur un ensemble de souches utilisées pour la fermentation de Cachaça.

Cet article est illustré par des arbres phylogéniques. Nous allons vérifier que nous obtenons les mêmes résultats en travaillant sur vingt-six échantillons de levures.

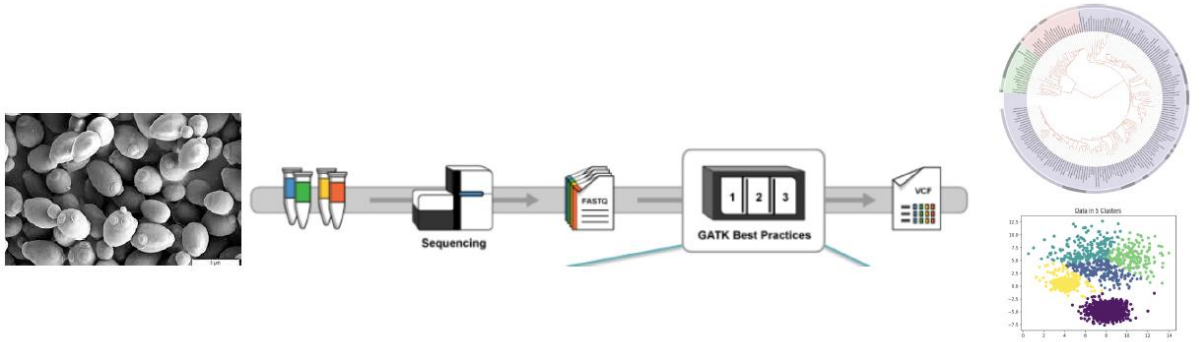


Figure 1 - Exemple de Pipeline

Pour répondre au mieux à cette problématique, nous avons conçu un pipeline de préparation et d'analyses de données. Notre script permet le téléchargement des données, en effectue l'analyse et présente sous forme d'images les résultats.

II Travail réalisé

II.1 Matériel

II.1.1 Outils

Notre script a été réalisé en Python et en R. Pour **Python**, nous avons utilisé les bibliothèques : *os, gzip, shutil, subprocess, hashlib, matplotlib.pyplot, numpy et ete3* ; pour le langage **R**, les bibliothèques : *lattice, VennDiagram, SNPRelate et ggplot2.*

Afin de réaliser ce projet, nous avons eu besoin d'installer et d'utiliser de nombreux outils. Nous précisons ici leurs noms, leurs versions et les commandes utilisées ainsi que quelques explications.

Pour l'alignement des séquences, nous avons téléchargé **BWA**, version 0.7.17-r1198-dirty. Nous avons implémenté les commandes :

Index	Crée les séquences de base de données au format FASTA. En effet, Bwa doit créer au départ la séquence avec laquelle les échantillons seront alignés. La commande s'utilise sur le génome de référence.
Mem	Aligne les séquences selon la séquence de base en fonction de un ou deux reads.

Dans la continuité, nous avons utilisé **SAMTOOLS**, version 1.11 :

View	Convertit un fichier sam en un fichier bam.
Sort	Permet de « trier » le fichier bam pour accélérer les commandes suivantes.
Faidx	Crée un fichier nécessaire pour l'appel des variants, cette commande s'appelle sur le génome de référence.
Flagstat	Compte le nombre d'alignement pour chaque type de flag, renvoie un fichier avec des statistiques sur l'alignement de l'échantillon.

BEDTOOLS, version 2.27.1, fut aussi installé.

Gemocov	Crée un fichier de statistique permettant par la suite de réaliser des analyses sur la couverture des échantillons.

BCFTOOLS, version 1.11 :

Query	Permet d'extraire d'un fichier vcf un autre fichier (ici texte) contenant seulement les colonnes utiles pour l'analyse des résultats et une meilleure compréhension de ces derniers.
View	Permet de filtrer un fichier selon un ensemble de filtres.

L'outil le plus utilisé dans notre pipeline est **GATK**, version 4.1.9.0. Les sept commandes que nous avons implémentées pour ce module sont :

MarkDuplicatesSpark	Marque les duplicats pour ne pas les prendre en compte par la suite.
CreateSequenceDictionary	Crée une séquence de dictionnaire pour la séquence de référence.
Haplotycaller	Extrait les variants pour chaque échantillon.

GenomicsDBImport	Crée une base de données pour ensuite appelé la commande suivante :
GenotypeGVCF	Crée une table résumée de format vcf contenant les informations de tous nos échantillons.
SelectVariant	Extrait d'un fichier vcf les variants que l'on souhaite étudier (SNP, …).
VariantFiltration	Ressort un fichier avec le résultat des filtres que nous avons appliqué.

Lors de l'installation des outils, nous avons effectué quelques choix d'implémentation et rencontré quelques problèmes. Nous avons décidé d'isoler BWA dans un dossier à part. Nous avons aussi rencontré un problème en téléchargeant GATK. En effet, GATK ne fonctionne qu'en Java 8. Nous avons donc installé Java 8 puis nous avons codé un script qui permet d'utiliser Java 8 pour exécuter GATK. Le script se finit en remettant le Java de la machine c'est-à-dire Java 11.

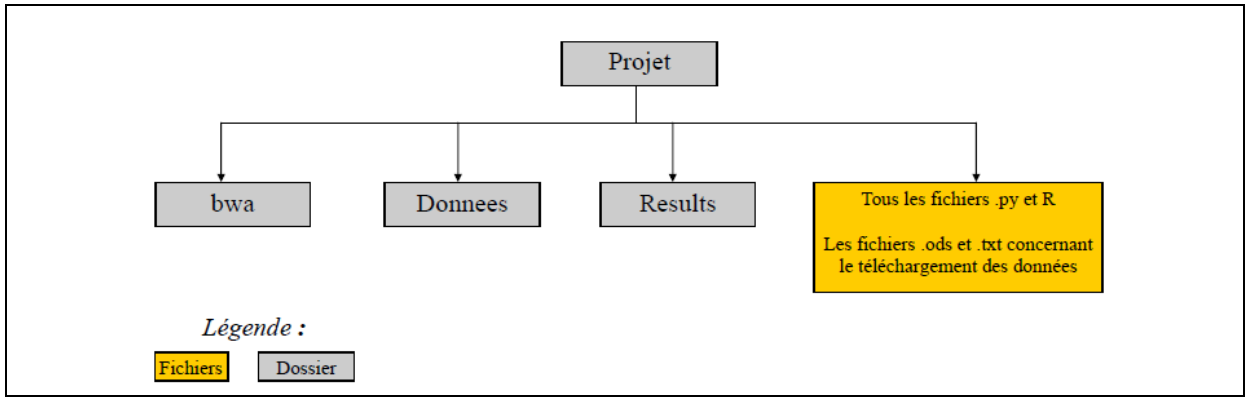
II.1.2 Architecture

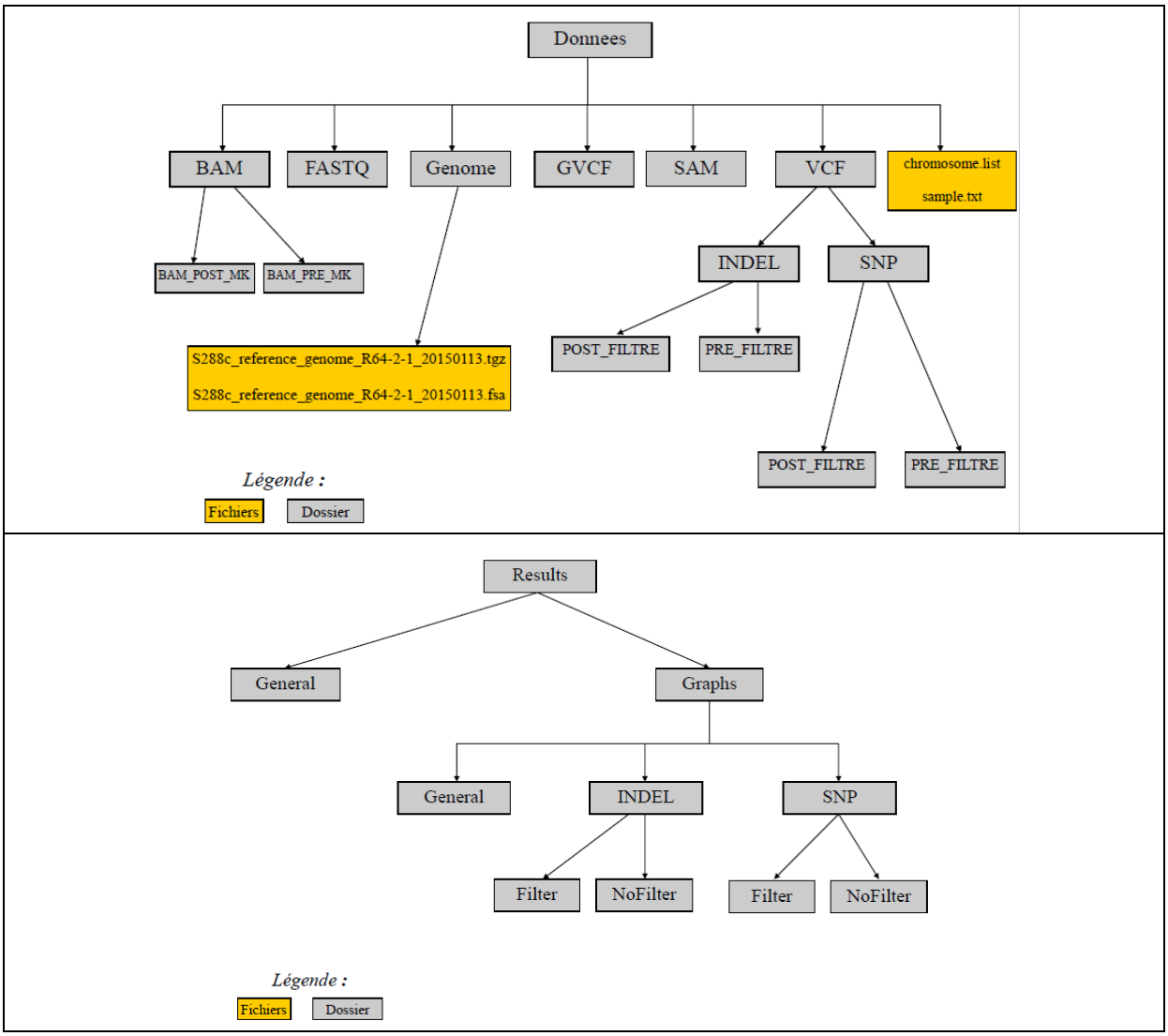
Nous avons fait tourner notre script dans un environnement Linux. Il faut un espace disque d'au moins 30 Go pour pouvoir télécharger les fichiers de données et stocker nos résultats. Pour que notre script fonctionne sur votre machine, quelques étapes préalables sont nécessaires : il vous faut notre organisation des dossiers et fichiers - disponible dans notre Github dans le fichier « architecture.zip » (https://github.com/George-Marchment/Projet-Bioinformatique-L3.git). Avant de lancer le script, il vous faut créer un « champ » utilisateur dans le fichier « variable.py » et indiquer quels sont les pass de vos dossiers sur votre machine.

Pour créer ce nouveau champ, vous pouvez vous inspirez des deux modèles déjà présents dans ce fichier :

Variante George	Si vous avez une clé USB pour télécharger les données. Certains pass doivent se trouver sur la machine (par exemple, la création d'un dossier n'est pas autorisée par toutes les clés USB)
Variante Clémence	Si vous avez suffisamment de place sur votre machine. Il suffit de remplir « l'adresse générale » (où se trouve votre dossier) et les pass suivants en découlent

Voici l'architecture de notre projet :





II.2 Méthode - Présentation Pipeline

Dans cette partie, nous allons expliquer en quelques phrases les différentes étapes de notre script et allons développer plus précisément la méthode BWA.

II.2.1 Téléchargement des données

Nous avons choisi d'automatiser le téléchargement des données et de fournir, dans l'architecture déjà présente, les fichiers relatifs au génome de référence. Le génome de référence date de 2015 et est celui du Saccharomyces cerevisiae, un micro-organisme, une levure particulière utilisée depuis l'Antiquité.

Pour le téléchargement des données nous sommes allés sur le site de l'European Nucleotide Archive (ENA) et avons téléchargé un premier fichier contenant les noms des échantillons (sample_name, run_accession, sample_alias), les liens de téléchargement et les md5 pour vérifier que le téléchargement s'est bien passé. Ce fichier s'appelle « donnees.txt ». Puis nous avons codé le téléchargement en utilisant WGET et en faisant des vérifications.

II.2.2 BWA – Approfondissement

BWA est un logiciel permettant l'alignement des séquences par rapport à un génome de référence, tel que le génome humain ou celui de levure dans notre cas. Il est composé de trois algorithmes: BWA-backtrack, BWA-SW et BWA-MEM.

Le BWA-backtrack est conçu pour des séquences Illumina de taille ne dépassant pas les 100 pb (Paire de bases) tandis que les deux autres algorithmes sont conçus pour des séquences plus longues variant de 70 pb à 1 Mpb. BWA-MEM et BWA-SW partagent des fonctionnalités similaires telles que la prise en charge de long-read et de split alignement. Un split alignement correspond à un alignement de séquences dans lesquelles différentes parties s'alignent sur des régions disjointes dans la séquence de référence.

BWA-MEM est généralement recommandé pour des séquences d'entrée de haute qualité car il est plus rapide et plus précis. BWA-MEM a également de meilleures performances pour des lectures Illumina de taille 70 à 100 pb. Nos séquences ont une longueur moyenne de 150 pb, nous avons donc choisi d'utiliser BWA-MEM pour l'alignement des séquences issues de nos échantillons.

Qu'est ce qu'un alignement ? L'objectif de l'alignement est d'organiser les séquences d'ADN, d'ARN ou de protéine pour identifier des régions de similitude entre deux (ou plusieurs) séquences. Des « trous » sont insérés à certaines positions dans les séquences, de manière à aligner les caractères communs sur des colonnes successives. Ces « trous » correspondent à des insertions ou des délétions de nucléotides ou d'acides aminés dans les séquences, on les appelle INDELs.

Explication de BWA-MEM : le BWA-MEM fonctionne en « seeding » et prolonge ensuite les « seeds » avec l'algorithme de Smith-Waterman.

« Seeding »	Les alignements avec « maximal exact matches » (MEMs) consistent à trouver des régions exactes d'une partie de la séquence de référence qui correspond à une partie de la séquence donnée en paramètres.
Smith-Waterman	Algorithme optimal donnant un alignement correspondant au meilleur score possible de correspondance entre les acides aminés ou les nucléotides des deux séquences. Le calcul de ce score repose sur l'utilisation de matrices de similarité ou matrices de substitution.

<u>Présentation de l'algorithme Smith-Waterman:</u>

Soit $A = a_1 a_2 \dots a_n$ et $B = b_1 b_2 \dots b_m$ les deux séquences que nous voulons aligner, avec n et m les longueurs respectives des séquences.

1. On commence par déterminer les matrices de substitution :
   - $s(a, b)$ càd le score de similarité entre les deux éléments des séquences
   - W_k càd la pénalité d'un « trou » (indel) de longueur k
2. Ensuite on construit la matrice de score $H \in M_{n+1,m+1}$ en initialisant la première ligne et la première colonne (on utilise l'indice 0 en référence à la première ligne et à la première colonne).

$$H_{k,0} = H_{0,l} = 0 \; avec \; 0 \leq k \leq n \; et \; 0 \leq l \leq m$$

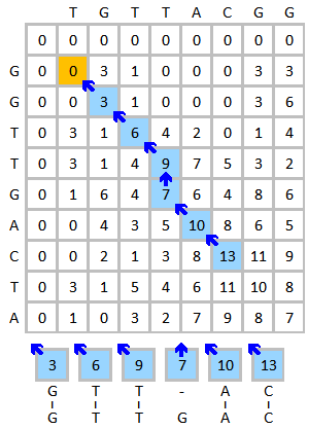
3. On remplit la matrice H en utilisant l'équation suivante :

$$H_{i,j} = \begin{cases} H_{i-1,j-1} + s(a_i, b_j), \\ max_{k \geq 1}\{H_{i-k,j} - W_k\}, \\ max_{l \geq 1}\{H_{i,j-l} - W_l\}, \\ 0 \end{cases} \qquad (1 \leq i \leq n, 1 \leq j \leq m)$$

avec :

- $H_{i-1,j-1} + s(a_i, b_j)$ qui correspond au score d'alignement entre a_i et b_j
- $max_{k \geq 1}\{H_{i-k,j} - W_k\}$ le score si a_i est à la fin d'un trou (indel) de longueur k
- $max_{l \geq 1}\{H_{i,j-l} - W_l\}$ le score si b_j est à la fin d'un trou (indel) de longueur l
- 0 signifiant qu'il n'y a pas de similitude entre a_i et b_j

4. On effectue un backtrack pour trouver les deux alignements. On commence par la case contenant le score le plus élevé dans la matrice H. A chaque étape, on doit choisir entre trois cases (soit la case de gauche, soit la case au dessus soit la case en diagonale haut gauche). On choisit entre ces trois cases celle qui contient un zéro (si elle existe) ou celle qui contient le score le plus élevé. La case zéro indique qu'on a terminé le backtrack.



Voici un exemple d'utilisation de cet algorithme :
Une animation se trouve à l'adresse [8].

II.2.3 GVCF

A l'étape précédente, nous avons créé les fichiers d'alignement. A l'aide de l'outil HaplotyCaller de GATK, nous avons « converti » ces fichiers. Les nouveaux fichiers contiennent pour chaque site du génome des informations à propos des variants (SNP, INDEL). Puis nous avons créé un unique fichier avec toutes les informations des vingt-six échantillons, sur lequel nous allons par la suite effectuer des filtres et des analyses.

II.2.4 Filtration

Nous avons créé des figures sur les données non filtrées pour observer les valeurs prises selon les différentes caractéristiques telles que la couverture moyenne, le scores de l'alignement, ... Leur étude nous a permis d'appliquer au mieux les filtres en trouvant les bonnes valeurs seuils et éliminer de faux INDEL ou SNP. Nous avons filtré environ 10% des SNP ou INDEL totaux. Lors de cette étape, nous avons aussi commencé à préparer les fichiers pour l'analyse des résultats en extrayant les informations nécessaires.

II.2.5 Analyse des SNP

Avec l'aide de R et de sa librairie SNPRelate, nous avons analysé les SNPs filtrés. Nous avons choisi de les représenter sous forme d'arbre et de clustering. Pour mieux comparer nos données, nous avons créé un fichier « sample.txt » contenant pour chaque échantillon le nom du groupe auquel il appartient (Bread, Wine, Cachaça1 …), selon les résultats de l'article « Genome Biology and Evolution ». Nous pouvons ensuite les analyser et comparer avec les résultats obtenus par le groupe de chercheurs.

III Résultats

III.1 Echantillons

Nous avons commencé notre projet en effectuant des images « globales de nos données » pour avoir un premier aperçu de celles-ci :

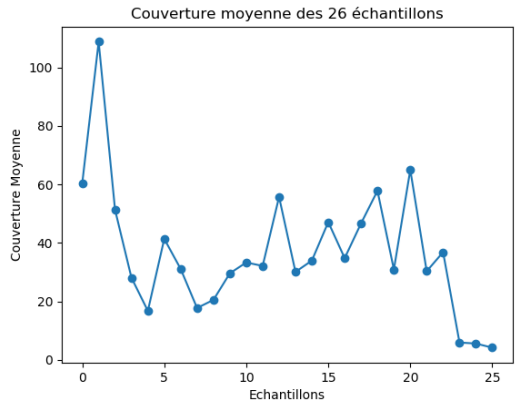


Figure 2

On observe sur ce premier graphique que les couvertures moyennes de chaque échantillon varient énormément. Ceci s'explique par le fait que certains échantillons ont été alignés selon deux reads tandis que les trois derniers échantillons ont été alignés sur un seul read.

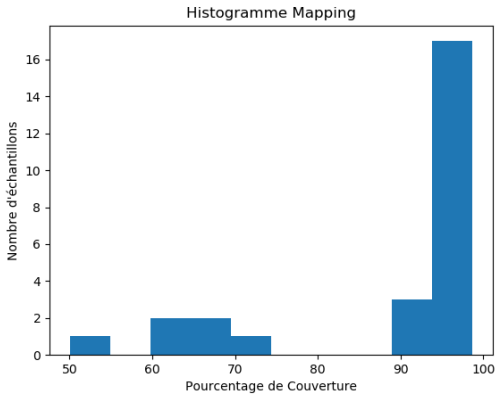


Figure 3

Cette figure a été obtenue après avoir lancé BWA et obtenu l'alignement. On observe que nous obtenons de très bons résultats : tous les échantillons ont un pourcentage d'alignement avec le génome de référence supérieur à 50%. Comme nos séquences ont une longueur moyenne de 150 pb, ces résultats sont très satisfaisants.

III.2 SNP

Exemple Valeurs sans filtres :

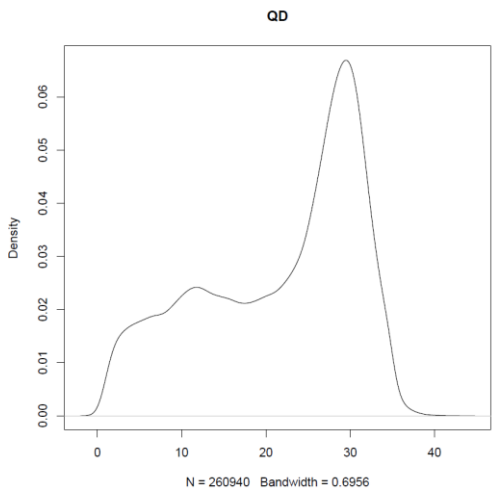


Figure 4

On observe sur cette courbe représentant la Quality Depth que les « vrais » SNP prennent des valeurs supérieures à vingt (il y a une forte densité) et qu'il y a de nombreux « faux » SNP aux alentours de dix. Nous allons appliquer un filtre : on ne souhaite garder seulement les SNP ayant une valeur supérieure ou égale à dix.

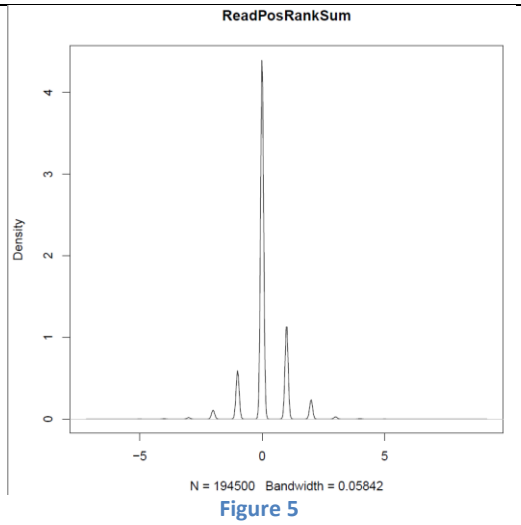


Figure 5

Cette caractéristique mesure si l'allèle alternatif a une position médiane dans les reads différente de la référence. Le calcul se base sur une approximation normale du test de Wilcoxon-Mann-Whitney. Le graphique nous montre qu'il y a un pic à zéro et un nombre élevé de valeurs supérieures à zéro. Ceci est le résultat que nous voulions : l'allèle alternatif est vu au milieu ou à n'importe quelle position. On a choisi un filtre bilatéral : on ne souhaite garder que les SNP ayant des valeurs comprises entre [-2 ; 2]

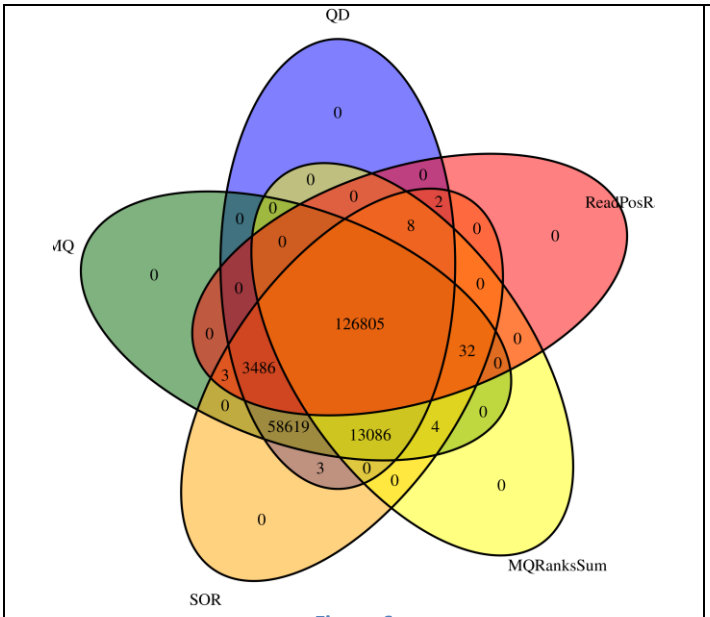


Figure 6

Après filtre :

Après avoir filtré nos données nous obtenons 202048 SNP. Nous avons filtré environ 22.57% des SNP totaux.

Le diagramme de Venn obtenu nous montre comment les SNPs ont été filtrés selon les filtres.

On observe que certains SNP ne possèdent pas toutes les caractéristiques. Ainsi, 126805 possèdent toutes les caractéristiques et 58619 SNP n'ont pas de valeurs pour MQRankSum et ReadPosRankSum.

A cette étape là, nous pouvons représenter les SNP sous la forme de clusters et d'un arbre phylogénique.

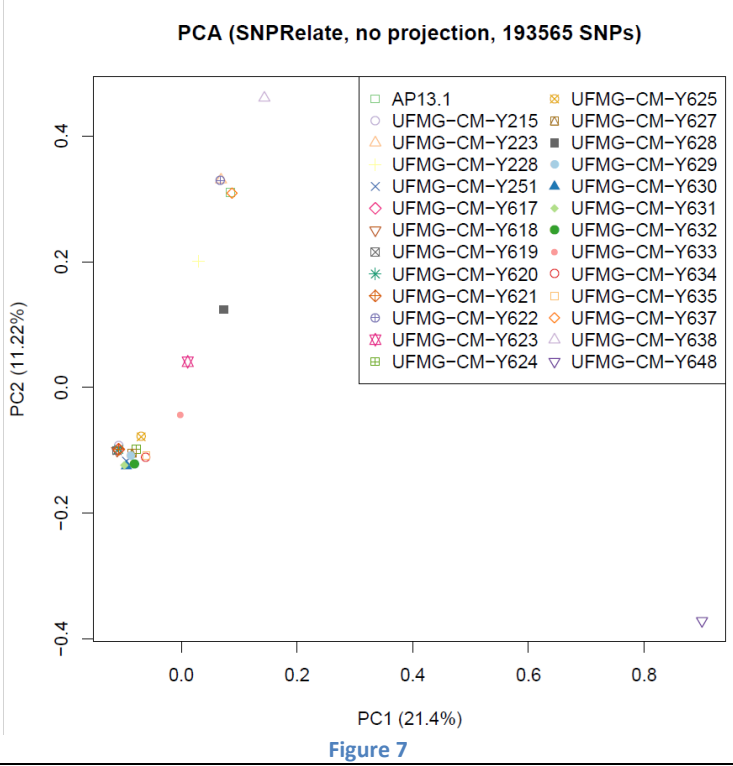


Figure 7

Graphique représentant les vingt-six échantillons après une PCA. Chaque échantillon est représenté par une couleur différente et un symbole (ce dernier permet aux daltoniens d'interpréter le graphique)

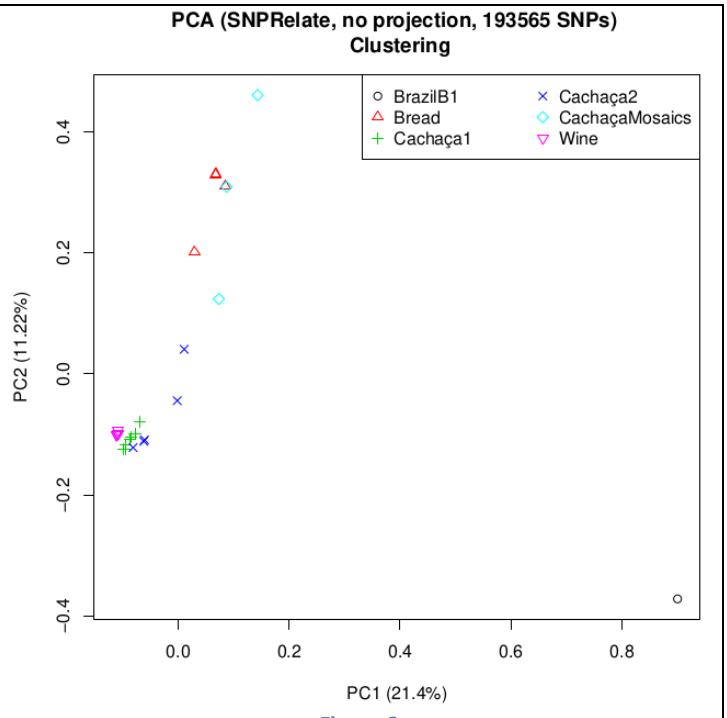


Figure 8

Graphique représentant les vingt-six échantillons selon leur groupe d'appartenance

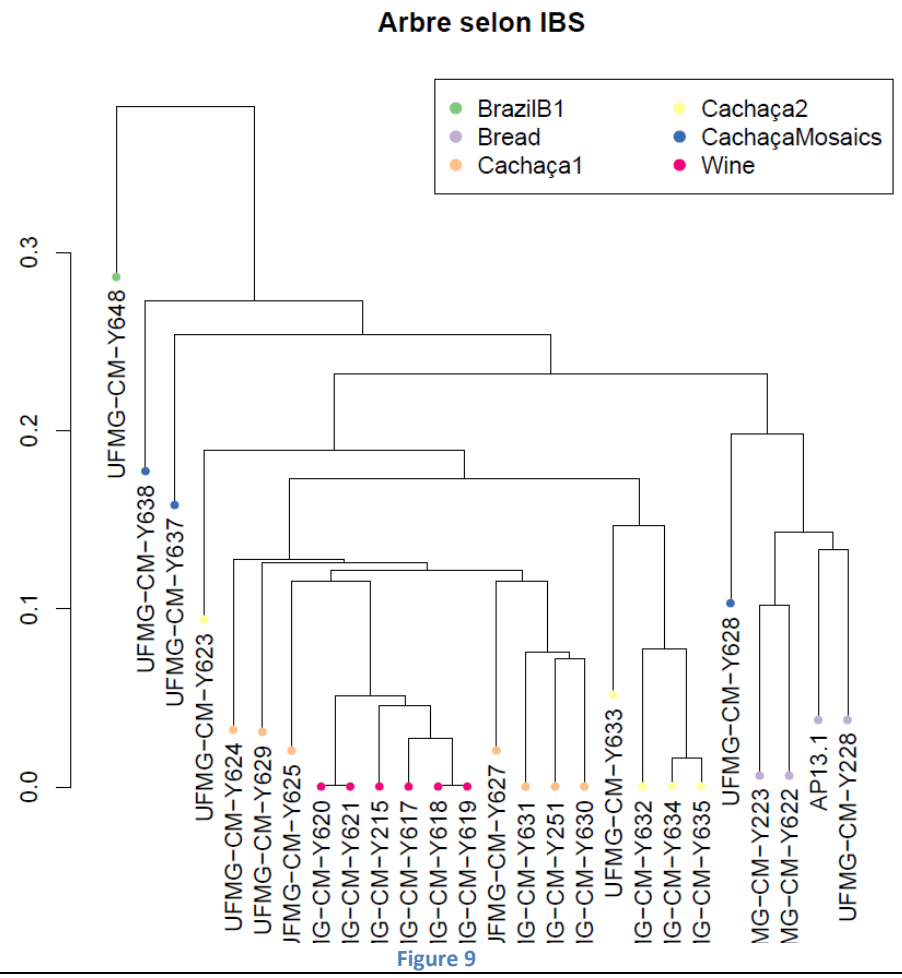


Figure 9

Arbre phylogénique de nos vingt-six échantillons

IV Conclusion et Perspectives

IV.1 Conclusion Générale

A partir des graphiques obtenus, nous pouvons émettre plusieurs **conclusions**.

La **figure 8** nous montre que les clusters obtenus respectent plutôt bien les groupes donnés dans l'article. Les levures appartenant aux groupes Wine et Cachaça1 sont toutes bien regroupées ensemble, elles forment deux clusters distincts mais très proches. Le groupe CachaçaMosaics « s'étale » un peu plus sur l'image et se mélange avec le groupe « Bread ». D'un point de vue général, on peut conclure que les données ont été analysées correctement car on retrouve bien les données d'un même groupe ensemble.

La **figure 9** représente l'arbre phylogénique de nos vingt-six échantillons. On observe au premier regard, que les différents groupes ont bien été respectés (chaque couleur est regroupée). Si nous devions retracer l'évolution de ces groupes, nous pourrions supposer que le plus ancien groupe est BrazilB1, que CachaçaMosaics s'est séparé ensuite et que son évolution a pris plusieurs années car Y638, Y637 et Y628 ne sont ni aux mêmes niveaux ni sur les mêmes branches. Seraient arrivées ensuite les levures les plus récentes, Cachaça1, Cachaça2 et Wine qui possèdent un ancêtre commun plus proche que Bread qui est lui aussi une levure récente.

En comparant les deux images, on retrouve bien ces résultats. Les groupes de Cachaça1 et Wine sont très proches sur les figures 7 et 8. De plus, on retrouve bien un élément de CachaçaMosaics parmi les Bread. BrazilB1 est bien éloigné des autres groupes. Pour finir Cachaça2 est lui aussi étalé mais reste proche et possède un ancêtre commun à Wine et Cachaça1.

Nos résultats sont en accord aves les résultats présentés dans l'article. Les chercheurs expliquent dans leur conclusion que la levure de Cachaça est un mélange entre Bread et Wine. On retrouve bien ce résultat sur la figure 8, on retrouve Cachaça2 qui fait le lien entre les groupes Wine et Bread. Les auteurs parlent aussi dans leur conclusion du temps de domestication de S. Cerevisiae, cette domestication ayant eu lieu en trois étapes. On retrouve bien ce résultat dans nos figures 7 et 8. BrazilB1 est séparé des autres levures dans le clustering ainsi que dans l'arbre (il se situe sur la branche extérieure). Les groupes de Cachaça sont des intermédiaires des groupes Wine et Bread.

IV.2 Méthodologie retenue

Pour chaque étape, nous avons choisi un échantillon pour mettre au point nos méthodes avant de généraliser à tous les échantillons. Cette technique de travail nous a permis de régler au mieux les paramètres de chaque commande et de débugger plus simplement. Cette méthode nous a aussi permis de ne pas avoir trop de difficultés en même temps et d'avancer dans le projet sereinement.

IV.3 Répartition du travail

Chaque mardi après-midi, nous nous retrouvions avec Fanny POUYET, notre référente sur ce projet, pour échanger sur nos avancées effectuées pendant la semaine précédente et lui poser nos questions. Les premières étapes ont été réalisées en binôme. Pour l'analyse des SNP, George s'est occupé de reprendre et d'adapter les méthodes d'affichage des arbres phylogéniques vues dans l'option « Introduction à la Bioinformatique » enseignée par Théophile SANCHEZ. Clémence s'est occupée pour sa part de la partie clustering et arbre en utilisant R. Pour finir le projet, George s'est occupé de la partie approfondissement (BWA) et de commenter le code, Clémence de rédiger le rapport.

IV.4 Nos impressions

Ce projet fut pour nous un premier vrai projet de recherche. En effet, il nous a permis de mettre en place un projet de A à Z, de réfléchir à la meilleure manière d'organiser un projet (dossiers et fichiers) mais aussi à son implémentation, à la séparation et à l'écriture des différents scripts pour qu'une personne extérieure puisse le comprendre le plus facilement : la communication dans un groupe est en effet un point très important pour une bonne organisation et le bon déroulé du projet. Ce travail nous a aussi permis de mieux comprendre nos machines informatiques : installation des outils et ajout d'un disque dur par exemple. Ce projet nous a poussé de plus à faire de nombreuses recherches pour approfondir, mieux comprendre les méthodes et le vocabulaire. A chaque étape, nous avions envie de découvrir la suite et hâte de connaitre les résultats finaux. L'analyse des données est surprenante : à partir de fichiers nombreux et volumineux, nous avons obtenu des images et fichiers simples et compréhensibles. Ce projet nous a conforté dans l'idée de continuer dans la voie Bioinformatique.

V Bibliographie

[1] GATK (broadinstitute.org)

[2] Burrows-Wheeler Aligner (sourceforge.net)

[3] Samtools(1) manual page (htslib.org)

[4] Bedtools: a powerful toolset for genome arithmetic — bedtools 2.30.0 documentation

[5] Bcftools (samtools.github.io)

[6] Génome de référence :

http://sgd-archive.yeastgenome.org/sequence/S288C_reference/genome_releases/

[7] ENA : https://www.ebi.ac.uk/ena/browser/view/PRJEB24932

[8]BWA animation : https://en.wikipedia.org/wiki/File:Smith-Waterman-Algorithm-Example-En.gif

[9]BWA vidéo : https://www.youtube.com/watch?v=4WRANhDiSHM&list=WL&index=1&t=3s

[10] BWA vidéo : https://www.youtube.com/watch?v=jA8RI4u_hd8&list=WL&index=2

[11] Test de Wilcoxon-Mann-Whitney — Wikipédia (wikipedia.org)