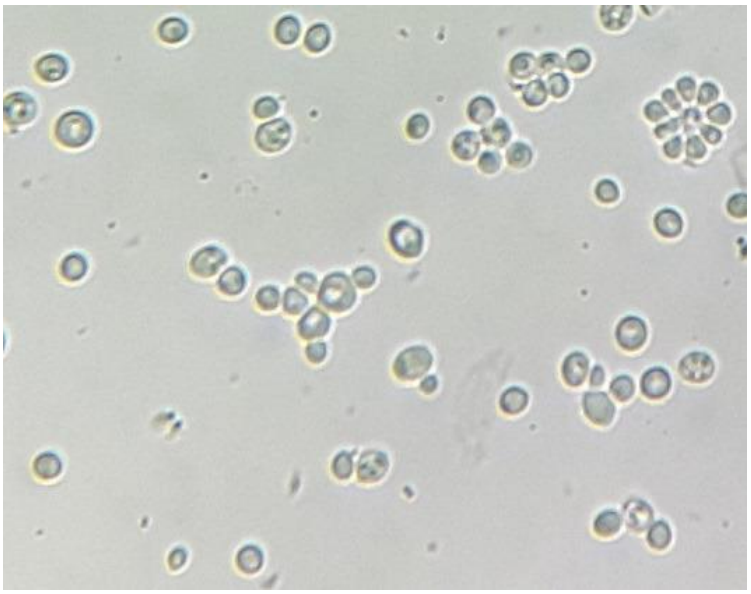
Rapport du projet intégré de Bio-informatique

Histoire génétique de la levure de Cachaça (S. cerevisiae)



Aspect microscopique des levures S. cerevisiae, après culture (grossissement x 400) (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7250219/figure/f0002/)

Plan de la présentation :

I) Introduction - Objectif informatique, contexte biologique, objectif du projet.

Nous allons construire un pipeline (succession de traitements automatiques du signal au travers d'une chaîne de logiciels et d'algorithmes) de préparation de données génomiques.
L'objectif informatique est d'écrire des scripts d'analyse de données pour illustrer les traitements appliqués et rendre compte de l'avancement à une équipe de recherche.

Le projet de cette UE consiste en la préparation de données , dans le but d'analyser la génétique des populations des levures,*(Barbosa, Pontes et al. (2018)) - déposé sous le code d'accession PRJEB24932)* , et en particulier l'histoire de la **levure de Cachaça (S. cerevisiae)** (Il y a 26 souches séquencées).

L'objectif global est de fournir à l'équipe une table de génotype c'est-à-dire une table qui indique l'ensemble des variations génétiques entre les individus de levures séquencées.

II) Matériel, méthode et résultats -
Notre solution consiste en un fichier bash principal pipeline.sh faisant appel à une multitude d'autres scripts spécifiques dans un but de lisibilité et de partage des tâches. Ce dernier supporte une demi-douzaine d'options, (parallélisme, fast-forward, reprise sur crash...) le rendant robuste et versatile. L'explication exhaustive de ces options se trouve dans le README du pipeline
Il fonctionne en 4 étapes principales que voici.

A) Collecte des données

Comment collectons-nous les données avec notre pipeline?

La version du génome utilisée est la suivante: S288C_reference_genome_R64-2-1_20150113 (16.8 MB) version du 13/01/2015.

On appelle le script dl.sh dans notre pipeline qui, à partir du fichier tsv obtenu dans une banque de données génétiques contenant des colonnes spécifiques, va télécharger les données nécessaires à l'analyse.
Nous avons sélectionné les colonnes: *sample_alias* (qui correspond aux noms des 26 souches sélectionnées), *fastq_md5* et *fastq_ftp* sur le TSV. Il peut y avoir un ou deux fichiers fastq pour une souche (simple ou double sens de lecture). La troisième colonne est l'adresse du fichier que va utiliser wget pour le téléchargement et la seconde un hash md5 servant de référence et permettant de confirmer que le fichier a bien été téléchargé, wget n'étant pas un programme effectuant ce genre de tâche.
Si on utilise l'option -d en appelant le pipeline, nous utiliserons juste cette partie du pipeline. Donc nous ferons uniquement un téléchargement simple des données fastq sur le site de l'ENA sans faire de traitement des données après cela.

Un compte-rendu des téléchargements est disponible dans dl_log.out

B) Nettoyage des données

Pour cette partie, nous allons utiliser un sous-script, cleandata.sh qui va réaliser en grande partie cette étape.
Cette étape du pipeline s'exécute en fonctionnement normal ou fast-forward du pipeline. L'option -k du pipeline va nous permettre de conserver si nécessaire les nombreux fichiers temporaires créés lors de cette étape.

Afin d'optimiser la rapidité d'exécution, on se permet de paralléliser l'exécution de cleandata.sh car l'ordre d'exécution n'est pas important pour ce script-là.D'autant plus que le gain de temps est plus intéressant à avoir ici que pour les autres étapes du pipeline. En effet, les algorithmes appelés par cleandata.sh sont très longs et avec l'option -p 4 (4 shells en parallèles) on a un gain de plus de deux heures!

Nous appelons l'algorithme BWA pour faire les premiers pas de notre nettoyage.

D'abord, nous utilisons l'option index pour avoir des fichiers d'index du génome de référence sauvegardés, on obtient alors 5 fichiers supplémentaires en plus du fichier fsa du génome de référence.

BWA-MEM - Nous utilisons l'option "mem" pour faire le "mapping". *BWA-MEM* permet d'aligner les séquences sur le génome de référence.
Des séquences de 70 pb à 1M pb peuvent être traitées avec cette option qui est plus rapide et précise que BWA Backtrack (pour des reads Illumina de 70 à 100pb). Il s'agit d'un alignement fractionné avec une lecture longue (requêtes de haute qualité).
L'algorithme Burrows-Wheeler cherche pour chaque read, la plus longue chaîne exacte dans le génome, et cette séquence est transformée en arbre pour accélérer la recherche.
BWA-MEM permet d'obtenir des fichiers *.bam* (binaires).

Les fichiers .bam sont obtenus après conversion des fichiers .sam grâce à l'algorithme Samtools et son option *view*. Cette option va afficher les alignements demandés en faisant cette conversion.
Ensuite, nous utilisons l'option sort de Samtools qui va trier les alignements d'un fichier BAM et produit un BAM trié (*sorted.bam*). C'est ce fichier .bam que nous allons utiliser pour la suite du pipeline.

Bedtools genomecov permet de calculer la couverture moyenne: position chrm début - chrm fin (nb de fois moyen qu'une position est lue dans le génome: comparaison en fonction de l'échantillon et corrélation avec le flagstat de SamTools). Ces deux fonctions sont utilisées respectivement avec cov.sh et flagstat.sh.
flagstat.sh et cov.sh sont des étapes rapides et on préfère garder l'ordre des échantillons du tsv pour avoir une meilleure visibilité dans nos résultats.

Avec ce fichier bam trié nous allons appeler un nouvel algorithme, MarkDuplicatesSpark de GATK , cela va permettre le marquage des duplicatas sans les éliminer, en ajoutant un flag là où il y a un duplicata.

	Minimum	Maximum	Moyenne
Reads mappés	50.03%	98.71%	88.02%
Couverture des échantillons	7.6	143.7	52.4

Résultat (% des données qui mappent le génome de référence et couverture des échantillons): grâce à flagstat et genomecov

C) *Identification*

Pour chaque échantillon, on appelle les variants à l'aide de la fonction Haplotype Caller de gatk.

HaplotypeCaller permet l'appel des variants sur les régions cibles. Le format de sortie souhaité est GVCF, Génomique Variant Call Format. Ce fichier contient les informations pour chaque site du génome, indépendamment de la présence d'un variant. Il y en a un par échantillon.
Ensuite, on regroupe les informations des variants au sein des 26 échantillons dans un VCF (Variant Call Format) via un appel joint.
Pour cela, on utilise la fonction GenomicsDBImport de gatk qui va créer une base de données décrivant les relations entités/attributs des échantillons et de chaque site. (grâce à Database.sh)
Puis, on utilise GenotypeGVCFs qui va créer la table de variants résumés, cela correspond donc à notre première table de VCF.
Un VCF ne contient des infos que pour les sites variants chez au moins un échantillon, c'est donc un format beaucoup plus léger que le GVCF dans notre cas.

La table finale de VCF contient deux types de variants:
-Les SNP (Single Nucleotide Polymorphism)
-Les INDEL (INsertion or DELetion polymorphism)

Sur un total de 288 314 sites variants identifiés 260 940 sont des SNP et 27 374 des INDEL soit un ratio 90.5% / 9.5%. On peut expliquer cette différence dans un premier temps, car le changement d'un nucléotide dans un triplet ne code pas systématiquement pour un acide aminé différent (22 acides aminés pour $4^3 = 64$ possibilités), contrairement à un ajout ou une suppression dont l'impact est beaucoup moins muet.
De plus, les SNP se produisent pendant la réplication de l'ADN, qui est un processus beaucoup plus commun dans la vie d'un organisme que les attaques/contaminations extérieures dont proviennent généralement les INDELS.

Le calcul du nombre de SNP/INDEL se fait dans INDEL_SNP.sh, grâce au format du VCF: la taille du champ REF et ALT sont identiques dans le cas d'un VCF et différents pour un INDEL.
Dans la suite, on ne s'intéresse qu'aux SNP qui seront donc extraits avec SelectVariants de Gatk.
La lecture et séquençage de l'ADN n'étant pas un processus parfait, il est associé à chaque read différents facteurs de qualité. Dans le but de faire des analyses fiables, nous allons filtrer nos SNP pour ne garder que ceux dignes de confiance.
Un choix s'impose donc, une filtration exhaustive gardant un maximum de variant quitte à en prendre des faux ou une filtration conservative réduisant drastiquement le nombre de variant exploitables.
Dans le pipeline, nous utilisons 6 filtres différents:
Quality Depth , Mapping Quality ,Mapping Quality Rank Sum ,Fisher Strand ,
Read Position Rank Sum ,Symmetric Odds Ratio

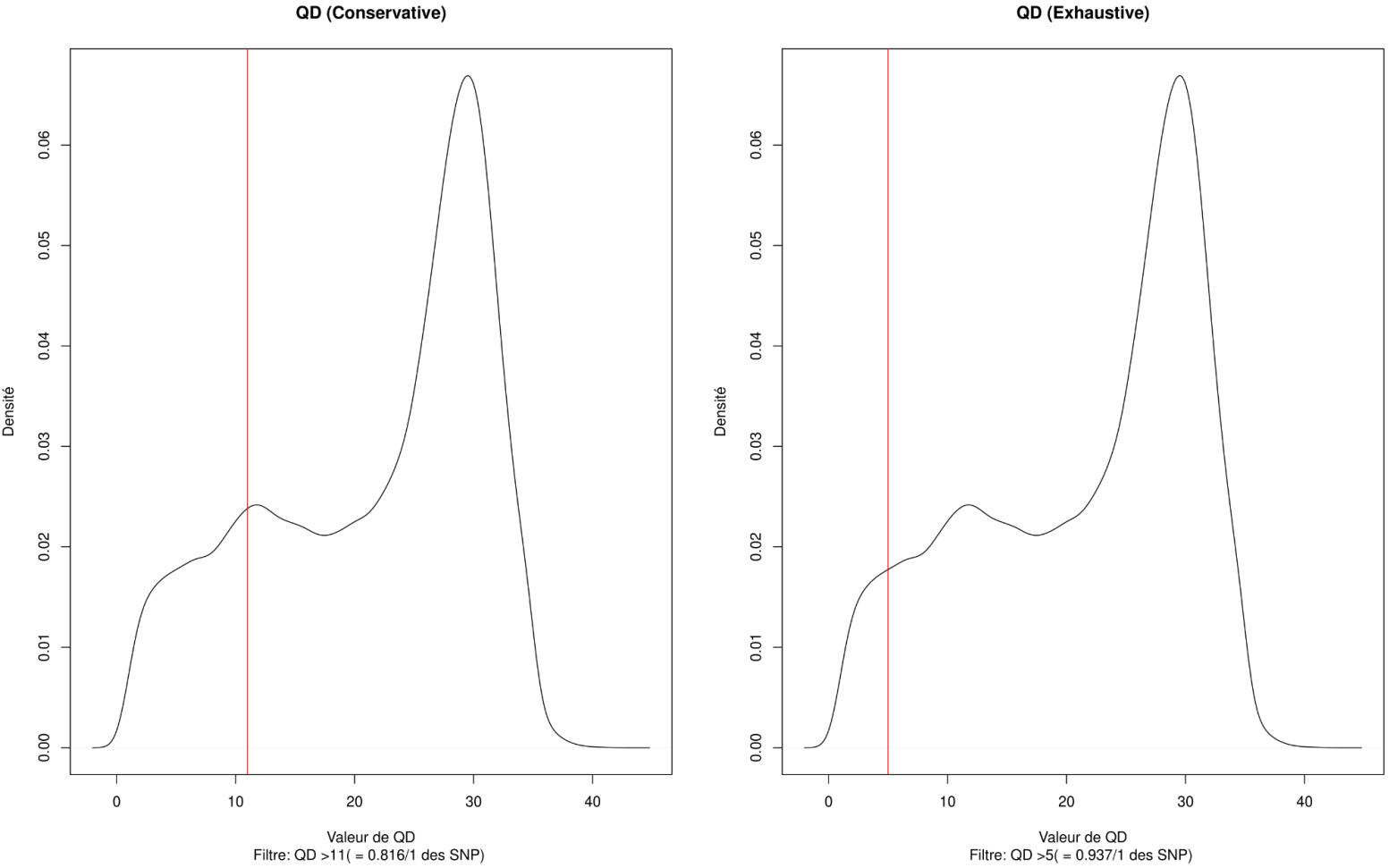
Les seuils choisis pour refuser un SNP sont les suivants pour respectivement l'approche:

-Conservative : "QD < 11.0 || FS > 5.0 || MQ < 58.0 || SOR > 1.3 || MQRankSum < -0.5 || MQRankSum > 0.5 || ReadPosRankSum < -0.5 || ReadPosRankSum > 0.5"

-Exhaustive : "QD < 5.0 || FS > 15.0 || MQ < 54.0 || SOR > 2.5 || MQRankSum < -2.5 || MQRankSum > 2.5 || ReadPosRankSum < -1.3 || ReadPosRankSum > 1.3"

D) *Analyse*

Analyse des résultats des figures obtenues



Ici, on voit les deux approches possibles dans le cas de QD, le détail du travail de filtrations est résumé dans Résultat/PRJEB24932_snp filtres Filtres et Diagrammes de Venn.pdf produit par le pipeline.

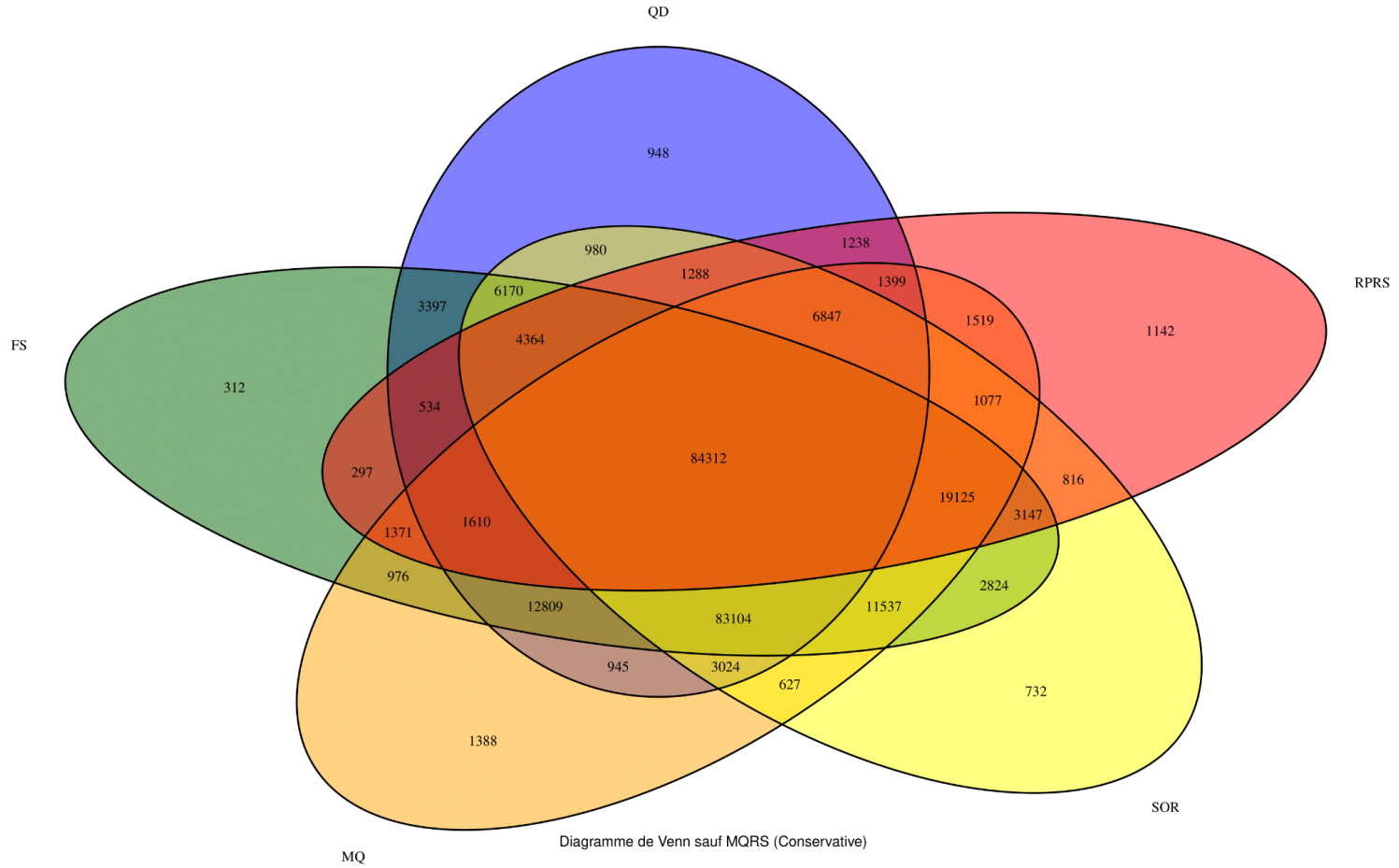


Diagramme de Venn sauf MQRS (Conservative)

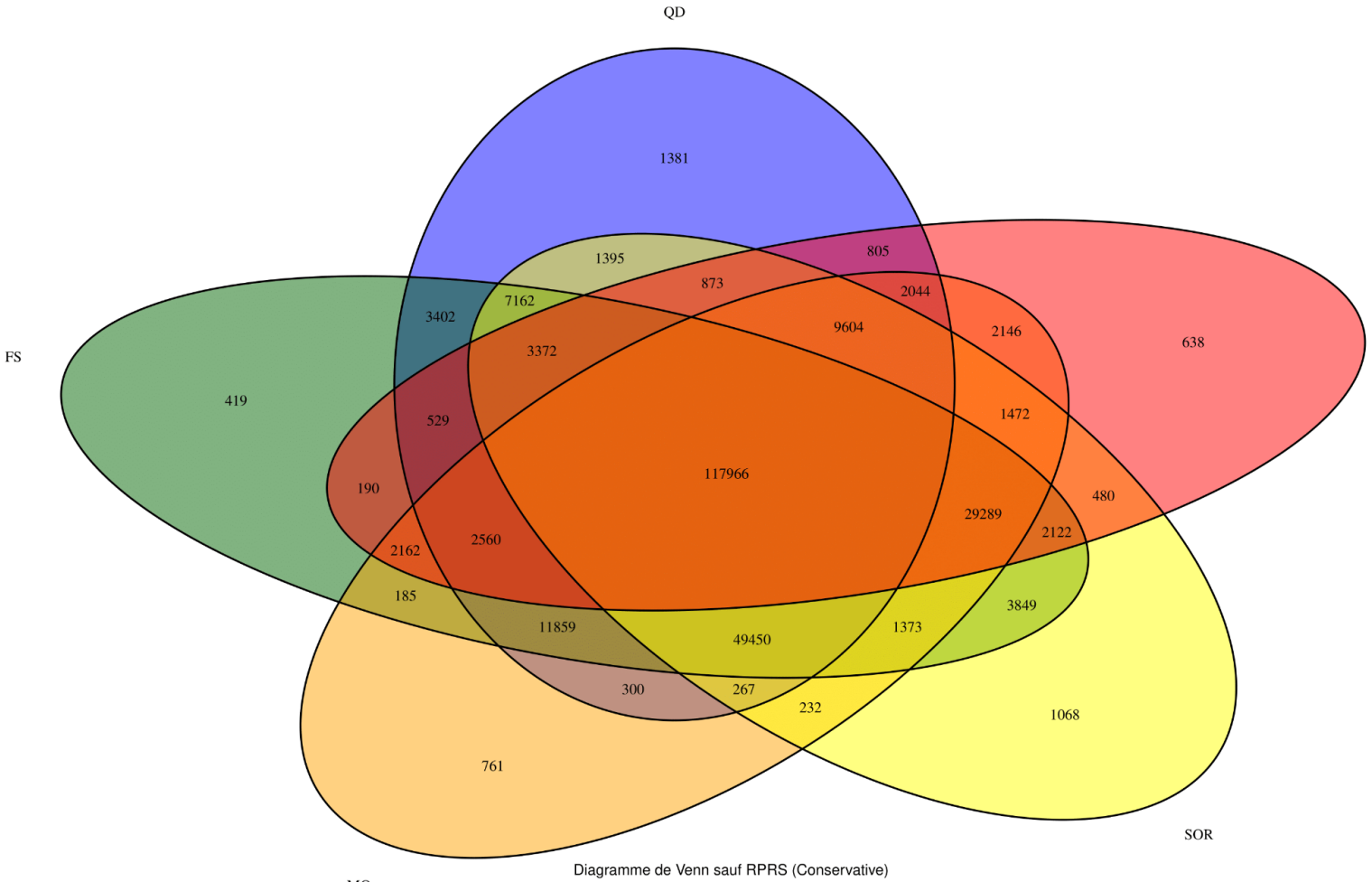


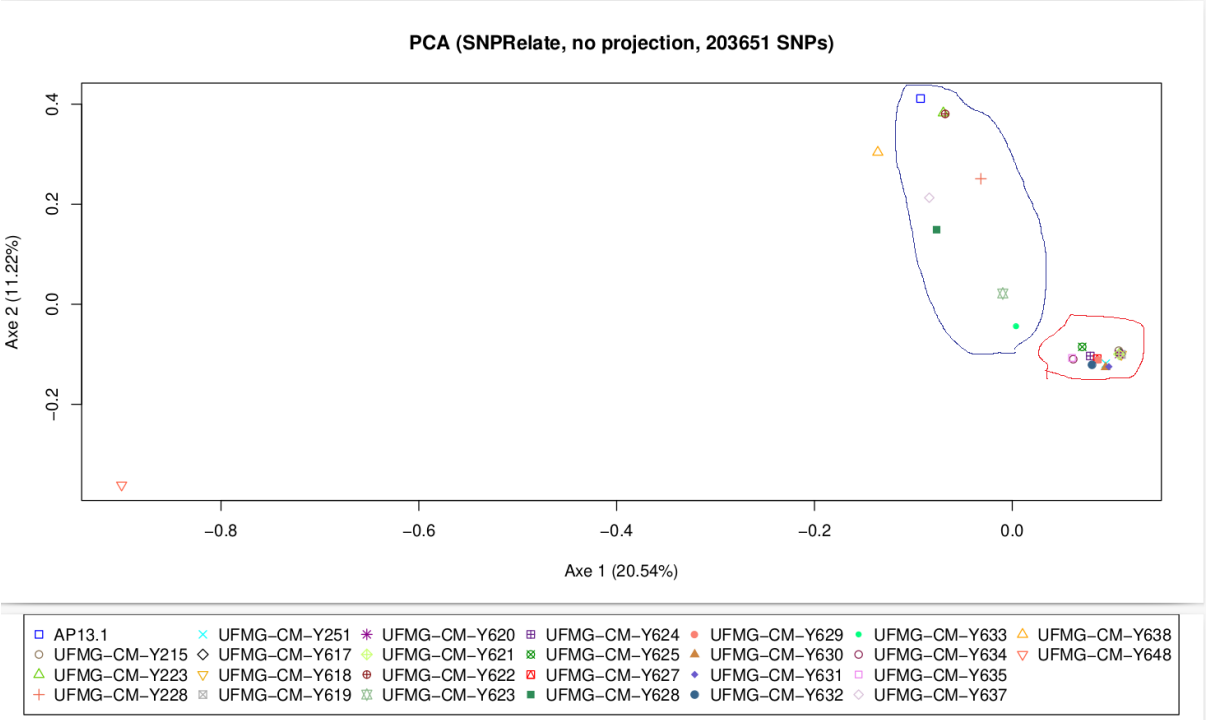
Diagramme de Venn sauf RPRS (Conservative)

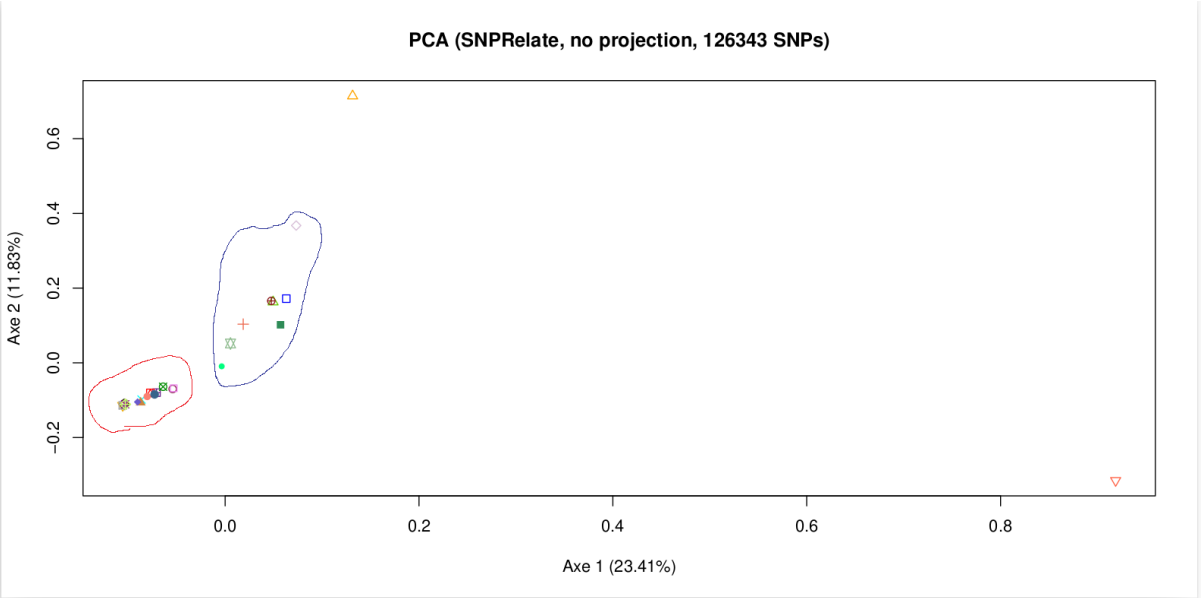
Ces deux diagrammes de Venn montrent les interactions entre les 6 différents filtres sur nos variants, dans le cas d'une approche conservative. VennDiagram de R ne supportant qu'un maximum de 5 filtres, deux figures sont nécessaires, au minimum, pour avoir une idée globale de la filtration. Dans ce cas, j'ai retiré arbitrairement MQRS et RPRS de la filtration.

Partant d'un VCF contenant 260940 SNP, 212 076 sont conservés (81%) pour le filtrage exhaustifs et 131 822 (50%) pour le conservatif. Le choix de la méthode de filtrages est paramétré par l'option -m "mode", par défaut conservative.
Si ces chiffres sont plus bien haut que ceux indiqués en théorie au centre des diagrammes de Venn, c'est, car une grande quantité de SNP ne possédant pas d'information pour MQRS et RPRS car ces filtres n'ont pas de sens pour les variants de type haploïde.

Ils sont considérés comme refusés dans les diagrammes de Venn, mais en pratique on ne prend pas en compte cette absence qui n'implique en rien une non-fiabilité du variant.

Résultat final:





Avec les SNP identifiés par le pipeline , nous avons pu faire une représentation en deux dimensions des similarités de nos 26 échantillons. Les deux graphiques sont respectivement pour l'approche exhaustive et conservative.
Il est un peu ardu de les comparer, car les échelles des deux graphiques ne sont pas les mêmes, nonobstant, on trouve bien des similarités entre les deux malgré la grande différence de SNP conservés, 203651 contre 126343.
-UFMG−CM−Y638 est stable et isolé.
-On identifie deux groupes stables parmis les 26 échantillons bien qu'ils ne soientt pas aux même positions d'un graphique à l'autre: AP13.1 , UFMG−CM−Y637 , UFMG−CM−Y622 , UFMG−CM−Y223 ,UFMG−CM−Y228 , UFMG−CM−Y628 , UFMG−CM−Y633 légèrement dispersé et le reste fortement groupé.

Cela montre bien la viabilité des deux approches de filtration, car si les résultats ne sont pas identiques, les tendances sont conservées.

III) Approfondissement - Biologie

Les résultats principaux issus de l'analyse phylogénétique (cf. Source A.)(basée sur l'étude de génomique des populations des souches de cachaça) sont:

Elles ne se regroupent pas en un seul clade (deux clades C1, C2) et ne sont donc pas monophylétiques (contrairement aux groupes domestiqués (sauf Beer 1))

Groupe monophylétique = clade : regroupe un ancêtre et l'ensemble de ses descendants.

Elles sont distinctes des principaux groupes de souches domestiquées (vin, bière 1, bière 2, pain et saké).

Elles sont **un mélange de souches diploïdes et tétraploïdes** alors que les levures de vin sont **diploïdes**

Les levures Cachaça partagent la majorité de leurs ancêtres avec les levures de vin. Elles apparaissent comme des levures de vin modifiées (domestiquées) : Hypothèse la plus forte

Domestication secondaire = présence du gène RTM1 dans les souches cachaça -> adaptation au jus de canne à sucre. (Probabilité d'acquisition par les levures de bière (clade Beer 1)).
La réacquisition de la prototrophie de la biotine par cooptation de BIO1/BIO6 à partir de S. paradoxus (ou de populations sauvages brésiliennes qui ont initialement acquis ces gènes à partir de S. paradoxus)

Multiples autres contributions de populations domestiquées, de lignées sauvages indigènes, de S. paradoxus

Introgressions (*) de 8 ORF en provenance de S. paradoxus dans les souches de Cachaça (mais pas dans celles de vin). de huit ORF provenant de la popula- tion nord-américaine de S. paradoxus

« La phase ouverte (**ORF**, Open Reading Frame) est la région de l'ADN qui sépare deux codons STOP. Dans celle-ci, une séquence codante (CDS, CoDing Sequence, région traduite en protéine) commence par un **codon START**, se termine par le **codon STOP** et est précédée d'un site de liaison aux ribosomes (RBS). » *(http://www.edu.upmc.fr/sdv/masselot_05001/genes_et_genomes/genomique.html)*

Les signatures de la domestication primaire = présence des régions B et C et certains des types de mutations inactivatrices des gènes d'aquaporine chez les levures de vin et de cachaça

(*) Introgressions en provenance de S-paradoxus:

Méthode: recherche à partir d'autres espèces de Saccharomyces, en mappant les reads à une référence combinée qui comprend toutes les séquences codantes annotées disponibles des espèces de Saccharomyces (grâce à BWA v0.6.2 (Li et Durbin 2009) avec les paramètres par défaut, mais en fixant le seuil de qualité à 10 (-q 10)) + SAMtools v1.1852 (Li et al. 2009) pour la manipulation des fichiers BAM résultants.

Un ORF est considéré comme d'origine étrangère à S. cerevisiae si sa couverture est supérieure ou égale à 1/4 de la couverture médiane du génome entier pour la souche analysée.

La couverture de l'**ORF** = *produit du nombre total de reads mappées aux ORF orthologues (descendent d'une séquence unique présente dans le dernier ancêtre commun aux deux espèces) par la taille des reads, divisé par la somme de la longueur de chaque ORF, en ne considérant que celles dont le nombre de reads mappées est supérieur à 25 % (par rapport à l'ORF orthologue ayant le plus grand nombre de reads) : pour un contrôle des alignements faux. Ce seuil de couverture a permis une certaine hétérogénéité dans les comptes de lecture et la présence éventuelle d'un ORF étranger avec l'ORF natif de S. cerevisiae.*

La **divergence par paires** entre S. paradoxus (souche YPS 138) et S. cerevisiae (souche S288c) a été utilisée pour rechercher des preuves d'introgressions de segments d'ADN de S. paradoxus dans les génomes des souches de S. cerevisiae. La divergence par site, (correction de *Jukes- Cantor* (**)) a été calculée en utilisant une fenêtre coulissante non chevauchante de 10 000 sites, en utilisant Variscan v2.0 (Hutter et al. 2006). En utilisant des assemblages de génomes de novo, une base de données BLAST locale a été produite pour chaque génome : cela permet de retrouver les gènes introgressés.

Les ORFs introgressés ont été recherchés par BLASTN (utilisation des séquences ORF de S. cerevisiae corrélatives disponibles comme requêtes).

(**): Introduction à la phylogénie (Théophile Sanchez & Sarah Cohen Boulakia): Dans le cas de séquences très proches, on estime que la distance évolutive réelle entre les séquences est proche de la p-distance qui est simplement le nombre de substitution dans l'alignement sur le nombre total de nucléotide. On applique ensuite la correction de **Jukes-Cantor** afin de prendre en compte le phénomène de saturation (un même site peut muter plusieurs fois au cours du temps). Sa formule est : $-(\frac{3}{4})\ln(1-(\frac{4}{3})\times p\text{-}distance).$

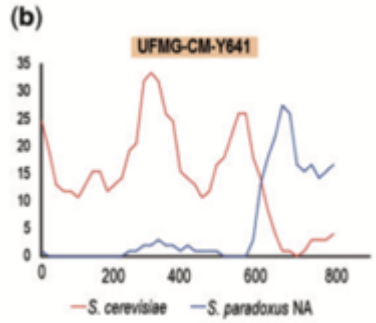


Fig.4 - (b) Divergence plot of recombinant (North American S. paradoxus X S. cerevisiae) FZF1 sequence of UFMG-CM-Y641

Source A: Multiple Rounds of Artificial Selection Promote Microbe Secondary Domestication—The Case of Cachaça Yeasts

*1,† 1,† 2 2 2 Raquel Barbosa , Ana Pontes , Renata O. Santos , Gabriela G. Montandon , Camila M. de Ponzzes-Gomes , 3 1 2 1, Paula B. Morais , Paula Goncalves , Carlos A. Rosa , and Jos□e Paulo Sampaio **

1UCIBIO-REQUIMTE, Departamento de Ciencias da Vida, Faculdade de Ciencias e Tecnologia, Universidade Nova de Lisboa, Caparica, Portugal 2Departamento de Microbiologia, ICB, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

3Laborato□rio de Microbiologia Ambiental e Biotecnologia, Universidade Federal de Tocantins, Palmas, Brazil - †These authors contributed equally to this work.

**Corresponding author: E-mail: jss@fct.unl.pt.* *- Accepted: June 29, 2018 - Data deposition: This project has been deposited at the European Nucleotide Archive (ENA) under the accession code PRJEB24932.*

IV) Conclusion et perspectives

Jérémie et Mathieu ont travaillé ensemble sur les scripts. Jérémie a beaucoup œuvré sur les très gros scripts, notamment sur le cœur du pipeline et les figures en R. Mathieu quant à lui a aidé à réaliser des plus petits scripts et les tests du pipeline pour voir si tout fonctionnait. Guillaume a aidé aux sous-parties A et B, à la rédaction du rapport et à notre approfondissement biologique. Il a rédigé le résumé de l'article ce qui nous permet de mieux comprendre le but de l'étude.

Nous avons, à l'issue de ce projet, réalisé un pipeline efficace qui pourrait servir de base à une multitude de problèmes biologiques ou médicaux. Une amélioration importante pour la portabilité de ce pipeline serait d'ajouter de permettre à l'utilisateur de choisir les filtres utilisés et les seuils de ces derniers.