Projet HAI724I – 2023-2024

Réplicats Biologiques

-

Anna-Sophie Fiston-Lavier

Contexte

En science, afin de tester la véracité de résultats de l'analyse d'une expérience, il est de mise de répliquer (au moins 3 fois) la même expérience. Dans le cas du projet d'innovation pédagogique BILL pour "Bioinfomatics Learning Lab", nous avons séquencé l'ADN du génome de l'Herpesvirus3 de trois échantillons (P15, P30, P50) avec 10 réplicats pour chaque échantillon. Toutes les données de séquençage obtenues (les séquences d'ADN appelées aussi des lectures) ont été alignées sur un génome de référence (séquence d'ADN) de la même espèce, qui nous sert de guide. Pour des génomes proches du génome de référence en terme de séquence, on s'attend à ce que la majorité des lectures s'aligne correctement sur le génome de référence. Cependant, des différences ponctuelles, au nucléotide près ou de plusieurs nucléotides, pourront être observées. Nous appellerons ces différences, des variants nucléotidiques (ou SNP) et variants structuraux (ou SV). A l'aide d'outils informatiques dédiés à l'appel de variants, les alignements ont été analysés afin de lister les SNP et SV. Les résultats sont stockés dans des fichiers tabulés appelés des fichiers VCF (https://fr.wikipedia.org/wiki/Variant_Call_Format). Les fichiers VCF pour trois réplicats pour deux échantillons sont mis à disposition sur moodle (P15.1, P15.2, P15.3, P30.1, P30.2, P30.3).

But du projet

Le but de ce projet consiste à implémenter un script principal ***main.sh*** pour comparer les fichiers VCF pour plusieurs échantillons et plusieurs réplicats de manière automatique. Ce script prendra en entrée le chemin des données et devra appelé deux scripts python :

1) A partir du chemin des différents données, un script ***parcourir.py*** devra parcourir les différents documents du chemin données en entrée. Si un document est un dossier, alors il devra parcourir le sous-dossier de manière à lister les fichiers VCF. Il faudra ensuite tester si les fichiers VCF sont bien des réplicats en comparant les noms de fichiers.

2) Un deuxième script ***compare.py*** aura pour but de stocker les données dans une structure de données de type dictionnaire. Les positions des variants serviront de clefs alors que les valeurs seront des listes de séquences. Il devra y avoir un dictionnaire par échantillon et les données seront ajoutées réplicat par réplicat (dicoP15 = P15.1, P15.2, P15.3/dicoP30 = P30.1,P30.2,P30.3). Dans une première version, seuls les variants au sein des réplicats avec la même position et la même séquence seront considérés comme communs (version 1). Si pour une position donnée, un variant du réplicat 1 a déjà été stocké dans le dictionnaire et que le même variant a été détecté pour le réplicat 2, alors il faudra ajouter à la valeur la séquence du variant du réplicat 2. A chaque ajout de données dans le dictionnaire, le script pourra compter le nombre de variants communs. Une autre possibilité sera de compter le nombre de variants communs après l'ajout de tous les variants des réplicats dans le dictionnaire. Dans une seconde version, ce script ne prendra en compte que la position avec +/- 10 nucléotides de différence (version 2). Le script devra donc retourner pour chaque échantillon, le nombre de variants communs.

Modalités du projet :

Le projet doit se faire en bash et python3 et n'utiliser aucun module hormis os, re, sys.
Le projet est facultatif (Contrôle Continu valant 30 % de l'UE).
Il peut être réalisé en monôme ou binôme (préférable).
Une démonstration sur machine devra être réalisée lors de la dernière séance de TP planifiée le mardi 5 décembre et les codes déposés sur le Moodle de l'UE