TP : Perl pour l'exploitation de fichiers biologiques

Le but de ce TP est d'écrire des scripts permettant la manipulation de fichiers biologiques et leur exploitation. Ces scripts devront être enrichis au fur et à mesure de plusieurs niveaux de complexité. C'est aussi réfléchir à la manière de procéder pour être le plus efficace possible en temps de calcul et utilisation des ressources.

Exercice 1 - Comptage du nombre de lectures lors d'analyses RNA-Seq

Le format BAM est un format de mapping de reads Illumina contre une référence.
Suite à des expériences RNA-Seq, les reads correspondant aux transcripts ont été mappés contre l'ensemble des gènes prédits d'un génome. Nous avons alors obtenus des fichiers BAM pour chaque échantillon.
A partir de ces échantillons que l'on veut comparer il est possible de réaliser un comptage du nombre de reads par gènes, reflétant l'expression des gènes.

Créer un script qui prend en entrée un nom de répertoire contenant un ensemble de fichiers bam indexés (répertoire /data/projects/tp-cluster/bams). Afficher dans la sortie standard, la base de nom de tous les fichiers (sans l'extension bam).
Lecture de répertoires, expression régulière

Réaliser pour chaque fichier bam le comptage du nombre de reads par gènes (avec la commande samtools idxstats) et rediriger la sortie vers un fichier dont l'extension sera « .idxstats »
Appel système sur un ensemble de fichiers

Récupérer les valeurs de comptage dans un hachage pour chaque gène, puis réécrire tous les résultats dans un tableau récapitulatif. Vous pouvez également réécrire ce procédé sans passer par un fichier intermédiaire.
Table de hachage

Exercice 2 - Exploitation de fichiers VCF

Le format VCF est un format de variants génétiques (SNP, insertions/délétions) obtenu suite à un alignement de lectures de séquençage NGS sur une séquence de référence.
Il s'agit d'un format tabulé qui peut être facilement parsé en Perl pour en extraire l'information et recueillir des statistiques. Chaque ligne correspond à un variant, tandis que l'information de génotypage est affichée en colonnes.

Créer un script permettant de lire le fichier VCF « /data/projects/tp-cluster/VCF/tp.vcf » et filtrer les variants dont la qualité est supérieure à 200 et les écrire dans un fichier « SNP_filtered.vcf ».
Attention, les premières lignes de métadonnées (commençant par #) doivent être conservées.
Ouverture de fichier en lecture et écriture, split, conditionnelle

En fin de traitement, afficher dans la sortie standard le nombre de variants conservés et éliminés.
Incrémentation

Ne garder que les variants ayant une profondeur totale (champ DP) supérieure à 30. Vérifier que le nombre de variants conservés a bien diminué ;
Expression régulière, conditionnelle combinée

Modifier ce script pour fournir en arguments le nom du fichier VCF d'entrée, le seuil de qualité à respecter puis le seuil de profondeur.
Gestion d'arguments

Nous souhaitons maintenant afficher le nombre de variants par chromosome. Afficher pour chaque chromosome (classé), le nombre de variants détectés.
Table de hachage, sort

Supprimer du fichier de sortie tous les variants pour lesquels tous les individus sont hétérozygotes (0/1). Attention, le traitement doit être générique pour tout fichier VCF (quelque-soit le nombre d'individus).
Foreach, comptage

Ne garder que les SNP présentant au moins un SNP hétérozygote avec un ratio des allèles supérieur à 40%.
Expression régulière

Séquence flanquantes des SNPs à partir du VCF et de la référence Fasta
Le script doit maintenant en plus ouvrir le fichier de référence fourni en 4e argument et écrire un 2e fichier « flanking.txt » contenant les séquences flanquantes (60pb de part et d'autre) de chaque variant précédemment filtrés. Pour économiser du temps de calcul, ce traitement peut être appliqué pour les 1000 premiers variants.
Réflexion d'un algorithme, table de hachage, substr, BioPerl

Annotation des variants.
Le script doit maintenant ouvrir le fichier d'annotation GFF fourni en 5e argument et sortir le nombre de variants qui sont localisés dans les gènes.