Projet de Programmation
M1 Bio-Info DLAD 2020-2021

Légende de la répartition :

Ghassan

Mégane

Ghassan et Mégane
(Voir le détail de la répartition en dessous)

Description de l'application

1- Ecrire un programme qui :
   a. lance une recherche BLAST d'un génome requête contre un autre génome cible (paramètres)
   b. lit le fichier résultat afin d'en récupérer la liste des meilleurs hits pour chaque protéine
   c. lance la recherche BLAST réciproque afin d'en déduire la liste des hits bidirectionnels.

Etape optionnelle/parallèle : écrire un programme qui interroge une source de génomes sur le web pour en proposer la liste/le choix à l'utilisateur, et les télécharger alors sur le disque.

⇨ Mise en place d'un programme fonctionnel par le terminal (pour un premier temps)

⇨ Adaptation du programme pour qu'il soit fonctionnel sur Tkinter

2- Ecrire un programme qui lancera le programme précédent pour un ensemble de génomes (>2), afin de les comparer tous les uns aux autres. Le programme regroupera les protéines ayant un hit bidirectionnel commun à travers ces différents génomes en *clusters* d'orthologues.

Etape optionnelle/parallèle 1 : développer une interface graphique pour sélectionner les génomes parmi ceux présent sur le disque et lancer le programme via cette interface. Une interface graphique de l'étape optionelle précédente peut être aussi envisagée.

Etape optionnelle/parallèle 2 : Créer une base de données composée de trois tables [GENOMES(id, name), PROTEIN(id, genome_id, cluster_id), BLAST(prot1_id, prot2_id, e-value, coverage] *pour stocker les données générées.*

Etape optionnelle/parallèle 3 : Proposer un graphique permettant de visualiser par exemple la distribution des e-valeurs à l'intérieur d'un cluster ou génome.

⇨ Mise en place d'une fonction qui génère un histogramme de la distribution des e-values (en format png)

⇨ Mise en place du choix et de la visualisation sur Tkinter

3- Ecrire un programme qui :

a. sélectionne les *clusters* n'ayant **pas plus** d'une protéine par organisme
b. génère un alignement multiple de chaque *cluster*
c. concatène les différents alignement en un super-alignement
d. lance la phylogénie des espèces à partir du super-alignement

Etape optionnelle/parallèle 1 : proposer une visualisation/analyse des clusters *pour ne sélectionner qu'une protéine par organisme (si plusieurs dans le* cluster*). Proposer un moyen de stocker cette information (sur le disque ou la base de données si créée à l'étape précédente). Inclure ces* clusters *additionnels dans la procédure (étapes b à d).*

Etape optionnelle/parallèle 2 : en supposant présentes sur le disque local les séquences ADN qui encodent ces protéines, écrire un programme pour générer le super-alignement nucléotidique correspondant.

Etape optionnelle/parallèle 3 : utiliser une librairie Python pour dessiner l'arbre phylogénétique résultat.

⇨ Mise en place du programme utilisant la librairie ete3 pour effectuer l'arbre phylogénétique

⇨ Ajout de la modélisation de l'arbre phylogénétique au programme dans Tkinter