PROJET DE PROGRAMMATION

2023-2024

SLEVI502

CAHIER DES CHARGES

Un biologiste vous fournit un tableau excel avec une liste de 30 génomes de la bactérie *Escherichia coli*. Il souhaite visualiser la synténie de différentes protéines dans ces génomes. Pour cela, il vous demande de réaliser un programme informatique avec une interface dynamique. Il vous donne l'exemple de l'outils de visualisation SynTView (Figure 1).

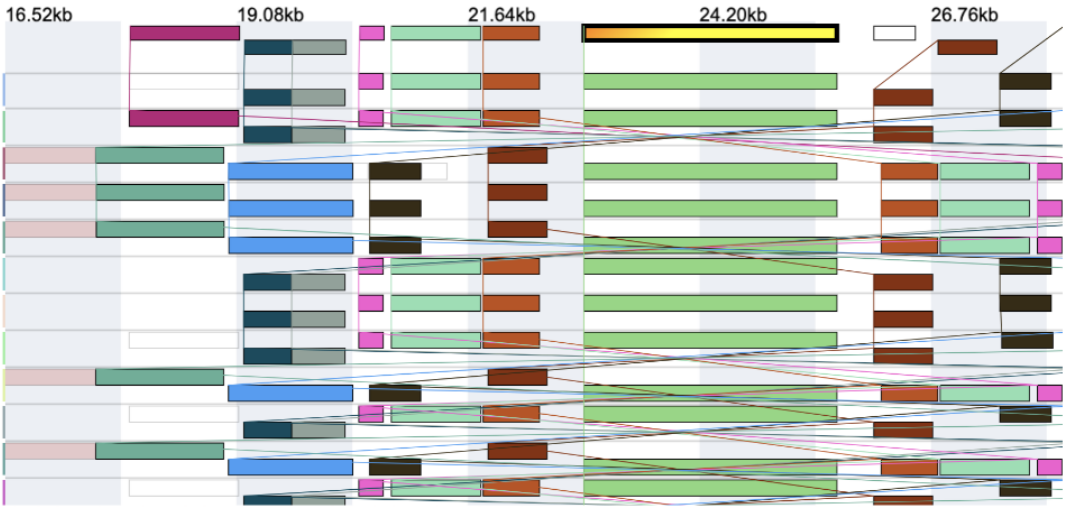

FIGURE 1 - EXEMPLE DE FIGURE DE SYNTENIE POUR 10 GENES SUR QUELQUES GENOMES. (SYNTVIEW)

En concertation avec le biologiste vous définissez le cahier des charges de ce programme. Il devra pouvoir :

- Lancer un BlastP de la protéine sur un autre génome *d'Escherichia coli*.
- A l'aide du tableau des protéines annotées, trouver la protéine en amont et en aval du génome.
- Lancer un BlastP des protéines en amont et aval.
- Faire une figure ou tableau de synténie montrant la conservation des 3 protéines

Le fichier excel (*data/Ecoli_genomes_refseq.xlsx*) donné par le biologiste inclut toutes les informations nécessaires pour accéder aux données des 30 génomes d'*Escherichia coli*.

- **Assembly Accession** est l'identifiant unique du génome
- **Refseq_id** est l'identifiant unique des chromosomes et plasmides de la bactérie.

Exemple :
Escherichia coli K-12 a pour assembly GCF_009832885.1 -
https://www.ncbi.nlm.nih.gov/assembly/GCF_009832885.1/
avec
un chromosome d'accession Refseq NZ_CP047127.1
https://www.ncbi.nlm.nih.gov/nuccore/NZ_CP047127.1
et un plasmide d'accession Refseq NZ_CP047128.1
https://www.ncbi.nlm.nih.gov/nuccore/NZ_CP047128.1

Le biologiste vous fournit aussi un dossier data/ contenant tous les fichiers utiles. Pour chaque souche il y a :

- Un tableau d'annotation avec la liste des gènes codants : annotation_{assembly_accession}.tsv
- Un fichier fasta avec toutes les séquences des protéines : protein.faa
- Les fichiers correspondant à la base de données BlastP du genome : fichiers .pot, .psq, .phr, etc.
- Un fichier d'annotation gtf avec tous les gènes annotés dans la souche : genomic.gtf

MODALITES

Vous effectuerez ce projet en binôme.
Votre projet se composera d'une archive (.gz, .zip, …) se composant de :

- le/les fichiers sources python (.py), le jupyter notebook (.ipynb) et les fichiers de données associés en n'oubliant pas de mettre des commentaires sur les principales lignes de votre programme.
- Un document court (max 5 pages) permettra d'expliquer votre projet, le contexte biologique, votre stratégie, vos remarques sur ce projet ou sur les résultats. Il devra expliquer l'utilité de chaque fonction en spécifiant le créateur de cette fonction.

Le projet devra être rendu pour le **vendredi 1 décembre (23h59)** en n'oubliant pas d'indiquer votre nom (dans le nom de l'archive et dans les fichiers sources). Votre programme devra être robuste vis à vis de l'utilisateur, des données récupérées, des résultats obtenus et devra gérer au mieux les erreurs possibles sans entrainer d'arrêt non souhaité du programme.

Vous aurez un oral le **vendredi ??? décembre**. Vous devrez faire une démonstration de votre programme en 5 minutes. Nous vous poserons une série de questions pendant 10 minutes sur les différentes fonctions et les choix fait. La note finale sera **une moyenne pondérée entre la qualité du code final, la finition du projet, l'oral et la participation dans le projet de chaque membre du binôme.**

EVALUATION :

RAPPORT ECRIT / 10 :

Intro - 3, Référence/illustration - 1, Description code (input, output, difficultés) - 4, recul sur le sujet + robustesse - 1, Compréhension de chaque partie - 1

CODE / 10 :

Fonctionnel – 3, Commentaires – 1, Gestion erreur – 1, Aspect général (facilité d'exécution) – 1, Jupyter (Partie bonus) – 4

ORAL (DEMONSTRATION) / 10

ORAL (REPONSE AUX QUESTIONS) / 10

STRUCTURE DU PROGRAMME

Pour vous aider à réaliser ce programme, nous l'avons décomposé en 6 parties. Chaque partie correspond à une ligne du cahier des charges. Les 5 premières parties doivent se regrouper dans un fichier projet.py. Celui-ci sera lancé en ligne de commande avec les données tests. La dernière partie importera projet.py dans un Jupyter notebook pour permettre à l'utilisateur de rechercher dynamiquement la synténie de ses protéines d'intérêt dans *Escherichia coli*.

Modules recommandés : Pandas et biopython

PARTIE 1 : RECUPERER LA LISTE DES PROTEINES ANNOTEES DANS UNE SOUCHE D'*ESCHERICHIA COLI*

Écrire une (ou plusieurs) fonctions permettant de lire les tableaux donnés par le biologiste (annotation_{assembly_accession}.tsv) pour chaque génome donné par son **Assembly Accession**, récupérer la liste des protéines, et accéder aux informations disponibles pour chacune des protéines.

Exemple : Vous devrez être capable de récupérer les informations pour la protéine WP_000004024.1 dans la souche GCF_001900435.1.

PARTIE 2 : RECUPERER LA SEQUENCE D'UNE PROTEINE DONNEE

Écrire une (ou plusieurs) fonctions permettant à partir de l'identifiant d'une protéine et l'assembly

accession de la souche de récupérer sa séquence en utilisant le fichier protein.faa.

Exemple : Vous devrez être capable de récupérer les informations pour la protéine WP_000004024.1 dans la souche GCF_001900435.1.

PARTIE 3 : LANCER UN BLASTP D'UNE PROTEINE SUR UN GENOME.

Écrire une (ou plusieurs) fonctions permettant d'exécuter un BlastP d'une protéine donnée sur un génome donnée et de récupérer le meilleur hit.

Le logiciel BlastP sera exécuté en ligne de commande en utilisant le logiciel blast+ que je vous fournis dans le dossier data (**data/Blast+**). Celui-ci dépend de votre système d'exploitation, veuillez ajouter un paramètre my_os_type afin de permettre à l'utilisateur d'indiquer son OS. Le programme a exécuté est :

- Blast+/linux/bin/blastp
- Blast+/macosx/bin/blastp
- Blast+/win/bin/blastp.exe

Afin d'exécuter une ligne de commande en python, il faut utiliser le module subprocess. Par exemple pour effectuer un « ls -l » vous devez écrire le code python :

```
import subprocess
cmd_line = ["ls", "-l"]
subprocessu.run(cmd_line)
```

Exemple : Vous devrez être capable d'exécuter un BlastP de la protéine WP_000004024.1 (provenant de la souche GCF_001900435.1) sur la souche GCF_001901165.1.
La ligne de commande a exécuté serait :

```
blastp -query GCF_001900435.1.fasta -db GCF_001901165.1/ GCF_001901165.1 -evalue
0.001 -out result_blast.xml -outfmt 5
```

Le résultat du Blastp sera sauvegardé en format xml (voir cours Biopython)

PARTIE 4 - TROUVER LA PROTEINE EN AMONT ET EN AVAL

Écrire une (ou plusieurs) fonctions permettant de trouver les protéines en amont et en aval d'une protéine donnée. Vous utiliserez pour cela le tableau d'annotation en sortie des fonctions de la partie 1.

Exemple : Vous devrez être capable de récupérer les protéines en amont et aval de la protéine WP_000004024.1 dans la souche GCF_001900435.1.

PARTIE 5 - LANCER UN BLASTP DES PROTEINES EN AMONT ET AVAL.

Écrire une (ou plusieurs) fonctions permettant de récupérer la séquence des protéines en amont et en aval pour lancer un BlastP sur un génome donné. Récupérer ensuite les 2 meilleurs hits, leur identifiant, leur pourcentage de similarité, et la position sur le génome cible.

Exemple : BlastP de la protéine en amont et en aval de WP_000059111.1 du génome GCF_009832885.1 (souche de référence *Escherichia coli K-12*) sur le génome GCF_001901165.1 (souche de référence *Escherichia coli M15 avec le taxid=1007065*).

PARTIE 6 – INTERFACE UTILISATEUR

Écrire un jupyter notebook dans lequel sera importé sous forme de module le fichier **projet.py**. Utiliser ensuite ce module pour permettre au biologiste d'étudier la synténie de n'importe quelle protéine dans n'importe quel génome (parmi les 30). Vous devez réaliser une interface complète permettant de sélectionner les protéines, les génomes, de lancer les BlastP, et d'afficher les résultats de synténie sous la forme que vous déciderez.

Exemple :