Université de Nantes

XMS2IE700 Rapport de Projet : Analyse et Visualisation de Séquences Protéiques en Bioinformatique

Mawuéna AHONDO & Ahamed TCHATAKOURA

14 Mars 2024

Sommaire

1 Introduction

Ce rapport présente les travaux pratiques réalisés dans le cadre du cours M1 BIOINFO XMS2IE700 : language de sccript 2 à Nantes Université pour l'année universitaire 2023/2024. Ce projet implique le développement d'un programme en Python pour l'analyse et la visualisation des séquences protéiques et a pour objectif de nous apprendre à d'automatiser le processus d'analyse de données biologiques pour permettre une interprétation rapide et visuelle des séquences protéiques, essentielle dans le domaine de la bioinformatique pour diverses applications, notamment l'étude des fonctions protéiques et la recherche biomédicale.

2 Sujet du projet

Le projet consiste à développer un outil de programmation en Python capable de télécharger, de décompresser si nécessaire un fichier GenBank, d'en extraire la première séquence protéique, de comptabiliser les occurrences des acides aminés et de générer un barplot des résultats (des acides aminés) dans un fichier PDF. Le programme devra être robuste et gérer les fichiers compressés en .zip et .gz.

3 Structures de données

Un dictionnaire est utilisé pour stocker les correspondances entre les codons et les acides aminés. Chaque clé du dictionnaire représente un acide aminé et sa valeur est une liste de codons correspondants. Par exemple, la clé "Leu" a pour valeur une liste de codons correspondant à l'acide aminé Leucine.Nous avons utulisé une dictionnaire parceque :

- c'est une structures de donnée efficace pour la recherche de correspondances entre des clés et des valeurs
- Il offre un accès rapide aux données en fonction de la clé
- Il permet de stocker des données de manière organisée et structurée

Une Liste pour stocker les acides aminés extraits de la séquence traduite. Chaque acide aminé est ajouté à cette liste à partir de la séquence de protéine traduite.Une liste à été utulisé pour de diverses raisonsd à savoir :

- c'est une structures de donnée efficace simple et flexible pour stocker une séquence ordonnée d'éléments
- Elle permet d'itérer facilement sur les éléments de la liste et de réaliser des opérations telles que le comptage d'éléments.
- adaptée pour stocker des données séquentielles sans clé.

4 Algorithmes utilisés

Dans le cadre de notre projet de bioinformatique, nous avons utilisé une série d'algorithmes et de scripts pour accomplir les tâches requises. Voici un résumé des composants clés de notre solution :

- **Script Python pour extraire et traduire les séquences protéiques (bases2prot.py):**
  - **Fonction extract_first_sequence:**

    parcourt un fichier GenBank pour extraire la première séquence de nucléotides après l'entête "ORIGIN" et jusqu'à la terminaison "//". La séquence est nettoyée des espaces et des chiffres pour ne conserver que les nucléotides.

  - **Fonctions transcription et traduction:**

    convertissent la séquence d'ADN en ARN, puis en séquence peptidique, en utilisant le code génétique fourni. Des fonctions auxiliaires comme **codon_to_aa** et **get_stat_aa** sont utilisées pour convertir chaque codon en acide aminé et compter les occurrences.

  - **Fonction affichage_acides_aminés_statistiques:**

    imprime sur la sortie standard les acides aminés et leurs occurrences formatées pour une utilisation ultérieure par barplot.py.

- **Script Python pour analyser les données et générer un barplot (barplot.py):**
  - **Fonction parse_input:**

    extrait les données de occurrences des acides aminés à partir d'un fichier donné, en vérifiant que le fichier est bien formaté et en stockant les données dans un dictionnaire pour un accès rapide et facile.

  - **Fonction generate_barplot:**

    utilise la bibliothèque matplotlib pour créer un barplot des données d'occurrences des acides aminés. Le graphique est ensuite sauvegardé dans un fichier PDF. Cette approche a été choisie pour sa simplicité et son efficacité pour produire des visualisations de données.

- **Script Bash pour le téléchargement et la décompression de fichiers (occurrences.sh):**

fait usage des commandes wget ou curl pour télécharger le fichier depuis une URL donnée. Utilise un script de décompression (**decompression.sh**) personnalisé pour gérer les fichiers .gz et .zip. Renomme et prépare le fichier décompressé pour l'analyse.

- **Script Bash decompression:**

La fonction **decompress** identifie l'extension du fichier et applique la commande de décompression appropriée (gunzip pour .gz, unzip pour .zip). Retourne le nom du fichier décompressé pour un traitement ultérieur ou un code d'erreur en cas d'échec.

5 Difficultés rencontrées

- Optimisation des temps de traitement pour les fichiers de grande taille et gestion des exceptions lors de la génération des PDF:
- Comment afficher les occurences et leurs nombres sous formes d'un tableau dans la fonction Pour trouver une solution nous noous sommes basés sur l'intelligence artificiele dans la fonction qui affiche les acides aminés
- Filtrer les caractères non standard dans la séquence lorsque le fichier genebank contient des caractères non standart
- Comment gerer les codons incomplets ou non standard dans la séquence d'ARN dans la fonction Traduction, Nous avons la possibilé de choisir d'ignorer, remplacer ou traiter autrement ces codons . Nous avons donc choisit d'ignorer le codon s'il n'est pas valide
- utulisation de l'intélligence artificiell pour decompresser le fichier si necessaire

6 Manuel d'utilisation

- **Pour exécuter le programme, utilisez la commande:**

  `python3 bases2prot.py <nom_du_fichier_genbank>`

- **Pour générer le barplot, utilisez:**

  `python3 barplot.py <nom_du_fichier_de_séquences> <nom_du_fichier_pdf>`

- **pour automatiser le processus complet:**

  `/.occurrences.sh <lien de téléchargement du fichier Genbank>`

7 Conclusion

En conclusion, ce projet de bioinformatique a été une opportunité riche et éducative pour développer un outil d'analyse et de visualisation de séquences protéiques. Grâce à ce travail, nous avons pu concevoir un programme fonctionnel qui répond aux besoins spécifiques du domaine de la bioinformatique, en particulier dans le traitement et l'interprétation des séquences de protéines. Les défis rencontrés au cours du projet, qu'ils soient anticipés ou imprévus, nous ont permis d'affiner nos compétences en programmation et en résolution de problèmes. Bien que certaines restrictions demeurent, comme la gestion des séquences très longues ou des annotations complexes, les résultats obtenus sont prometteurs et ouvrent la voie à de futures améliorations et extensions du programme. Les compétences acquises en matière de structures de données, d'algorithmes de décompression et de parsing, ainsi qu'en visualisation de données avec matplotlib, sont des atouts qui nous seront bénéfiques bien au-delà de ce projet.Pour la réalisation du rapport du projet nous avons utulisé un language de balisage :**latex**

Nous espérons que ce programme servira d'outil utile à la communauté de la bioinformatique et inspirera de futures recherches et développements dans ce domaine dynamique et en constante évolution. En définitive, ce projet n'est pas seulement une fin en soi, mais une étape importante dans notre parcours éducatif et professionnel.