Info123 - Projet pluridisciplinaire en Informatique

Dans ce projet, nous allons implémenter différents algorithmes pour comparer des séquences d'ADN ou découvrir un gène dans une séquence. Les 4 bases de l'ADN sont A, C, T, G, une séquence d'ADN sera donc représentée en mémoire par un tableau contenant des caractères. Les tableaux peuvent être définis à l'avance et de taille suffisante, disons 1000 mais une gestion dynamique de la mémoire sera appréciée. Les fonctions à implémenter sont décrites dans la suite du document, mais vous pouvez en ajouter d'autres qui vous semble intéressantes (et ce sera apprécié !). Il sera tenu compte de la lisibilité du code, de la diversité et de l'efficacité de vos programmes.

1 Préliminaires

1.1 Stucture du programme

Votre fonction *main* affichera des choix multiples:

1. Saisir la première séquence.
2. Saisir la seconde séquence.
3. Afficher la première séquence.
4. Afficher la seconde séquence.
5. Comparer les deux séquences.
6. Chercher la première séquence dans la seconde.
7. Quitter.

Vous pouvez bien sûr ajouter des options supplémentaires. Ensuite le programme va demander une entrée utilisateur pour choisir quelle action réaliser. Une fois celle-ci accomplie, on réaffiche les choix multiples.

1.2 Entrées-sorties

Il faut avoir des données pour tester vos algorithmes. Dans un premier temps, vous pouvez coder en dur des exemples de séquence. Pour que le projet soit complet, il faudra implémenter au moins une des procédures suivantes:

1. Une procédure qui lit au clavier une suite de caractères et renvoie un tableau contenant une séquence d'ADN ou une erreur.

2. Une procédure qui lit sur le disque un fichier texte contenant une séquence d'ADN et renvoie un tableau contenant cette séquence ou une erreur.

3. Une fonction qui génère une séquence de bases aléatoires de taille passée en argument.

1.3 Fonctions de base

Une fonction *retourne* qui prend une séquence en entrée et renvoie la même séquence dans l'ordre inverse. Si votre tableau m a n cases, le tableau t que vous devez renvoyer vérifie $t[i] = m[n-1-i]$.

Une fonction *simplecompare* qui compare deux chaînes de la *même* taille. Elle calcule la distance entre chaîne de la manière suivante: La première séquence s'écrit $a_1.w_1$ ou a_1 est une base et w_1 le reste de la séquence. La deuxième séquence s'écrit $a_2.w_2$. Si $a_1 = a_2$ alors $d(a_1.w_1, a_2.w_2) = d(w_1, w_2)$ sinon $d(a_1.w_1, a_2.w_2) = 1 + d(w_1, w_2)$.

2 Recherche de gène

Écrire une fonction *contain* qui prend en entrée deux séquences d'ADN w_1 et w_2 et renvoie vrai si w_1 est exactement contenue dans w_2.

Pour ce faire il y a plusieurs méthodes. La plus simple est de tester pour chaque position de w_2 si on y trouve le mot w_1 à partir de cette position. Vous pouvez également construire automatiquement automate qui reconnaît w_1 et le faire tourner sur w_2 ou utiliser une méthode encore plus efficace comme l'algorithme de Knuth-Morris-Pratt Voir par exemple le chapitre 32 d'introduction à l'algorithmique de Cormen, Leiserson, Rivest et Stein ou les pages wikipedia correspondantes.

Parfois les gênes se retournent: tester également si la séquence w_1 une fois retournée est dans w_2.

3 Similitude entre chaînes

On ne veut pas nécessairement déterminer si deux chaînes sont égales mais si elles se ressemblent. Voici quelques mesures possibles, à vous de les implémenter.

Toutes les substitutions n'arrivent pas avec la même probabilité. On donne ici la fonction d qui donne la distance entre les bases, par exemple $d(C, G) = 2$.

d	A	C	G	T
A	0	2	1	2
C	2	0	2	1
G	1	2	0	2
T	2	1	2	0

Soit $v = v_1.v_2 \dots v_l$ et $w = w_1.w_2 \dots w_l$, la distance entre les deux chaînes est définie par $d(v, w) = \sum_{1 \leq i \leq l} d(v_i, w_i)$. Implémenter une fonction qui calcule cette distance et proposer une manière de gérer deux chaînes de tailles différentes.

Plutôt que de comparer deux séquences qui peuvent être énormes, on préfère généralement comparer des grandeurs caractéristiques de ces séquences. Soit w_1 une séquence de taille k, la statistique de w_1 dans w_2 est le nombre d'occurences de w_1 dans w_2 divisé par la taille de w_2. Le vecteur des k-statistiques de w_2 contient les statistiques de tous les mots w_1 de taille k dans w_2. Ecrire une fonction qui prend en entrée une séquence d'ADN w et un entier k et qui renvoie le vecteur des k-statistiques de w. Vous pouvez vous contenter d'implémenter cette fonction pour $k = 2$, c'est ce qu'on appelle calcul des *bigrammes*.

La fonction qui suit doit être implémentée en suivant une méthode appelée **programmation dynamique**. Cette méthode plus complexe sera expliquée en classe.

On veut comparer deux chaînes de taille potentiellement différente et évaluer si elles sont proches à mutation génétique prêt. Écrire une fonction qui prend en entrée deux séquences d'ADN w_1 et w_2 et retourne le nombre de mutations (insertion et délétion de bases) pour transformer w_1 en w_2.