

LV 229 – Introduction à l’algorithmique et à la programmation II

TD 5

Les séquences d’ADN sont représentées par des chaînes de caractères ne comprenant que quatre symboles : A, C, G ou T. La recherche de signaux dans des séquences dépassant souvent le millier de caractères (taille approximative d’un gène bactérien) et pouvant parfois atteindre des milliards de caractères (taille approximative d’un gros chromosome du génome humain) illustre le caractère indispensable de l’outil informatique pour traiter des chaînes de caractères en biologie.

Nous recherchons ici à développer un outil simple d’annotation permettant de trouver les Régions Potentiellement Codantes (RPC) dans de grandes séquences ADN.

Le modèle est le suivant :

- Une RPC débute par un codon start
- Une RPC se termine par un codon stop en phase avec le start
- Les RPC peuvent être situées sur la séquence étudiée ou sur sa séquence complémentaire (il y a donc 6 phases possibles).

Le fichier `Esco.fst` (disponible sur le site) contient la séquence d’un fragment de 10 kilobases du génome de la bactérie *Escherichia coli* en format fasta :

```
>Esco
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAGAGTGTCT
TGATAGCAGCTTCTGAACCTGGTTACCTGCCGTGAGTAAATTTAAATTTTATTGACTTAGG
TCACTAAATACTTTAACCAATATAGGCATAGCGCACAGACAGATAAAAAATT
```

Exercice 1 (TD/TP) – Etapes préalables

Les fonctions `substr`, `lit_fasta`, `complement` et `rev_comp` des TD3 et TD4 pourront être ré-utilisées pour ce TD.

Exercice 1 (TD/TP) – Codon(s) START.

Trouver l’algorithme d’une fonction `start()` qui cherche la position du premier codon ATG (codon START) à partir de la position p dans une séquence et retourne la position du ‘A’ du ATG sur la séquence en question. Ne pas oublier de traiter le cas où aucun ATG n’est trouvé (la fonction retournera -1).

Exercice 2 (TD/TP) – Codons STOP.

Trouver l’algorithme d’une fonction `stop()` qui cherche dans une séquence, à partir de la position p , la position dans la séquence de la première base du premier codon STOP (TGA, TAG ou TAA) en phase avec cette position p . La fonction `stop()` devra retourner la position trouvée. Ne pas oublier de traiter le cas où aucun stop en phase n’est trouvé (la fonction retournera -1).

Exercice 3 (TD/TP) – Annotation des phases ouvertes de lecture.

a) En s’appuyant sur les fonctions `start()` et `stop()` déjà définies, écrire l’algorithme d’une fonction `RPC()` qui cherche dans une séquence (chaîne de caractères), à partir de la position p , la première RPC et qui stocke dans un tableau de longueur 2 passé en argument la position du début du start et de la fin stop (si pas de start ou pas de stop, elle stockera [-1,-1]).

b) Ecrire la fonction `toutes_RPC()` qui permet d’imprimer à l’écran en format fasta, pour toutes les RPC, un identifiant (le nom de la séquence et un numéro), la position de début et de fin (la région couverte par la RPC), ainsi que le brin et la séquence de la RPC :

```
>Esco_1|30|98|D
ATGTCTCTGTGTGGATTAAAAAAGAGTGTCTGATAGCAGCTTCTGAACCTGGTTACCTGCCGTGAGTAA
>Esco_2|190|255|D...
```