Option Bioinformatique L3
TP2

2016

Cours disponible ici:

https://docs.google.com/presentation/d/1AswTLepd2WwFNf9nEwN3sHFmgplQiHcoWY-Nfpym5nM/edit

Preuve du Lemme sur le nombre de noeuds internes / feuilles:

http://goo.gl/QUPnfc

Le code source réalisé pendant ce TP2 sera ajouté au code source du TP1. A la fin de tous les TPs, le résultat final sera un seul outil, “bioseq”, comprenant les fonctionnalités implémentées dans chaque TP. Il est à rendre à la même date que le TP1, à l’adresse suivante:

rayan.chikhi@univ-lille1.fr

avec le sujet de mail suivant (remplacer Nom1, Nom2 par chaque nom de famille; monomes autorisés):

[BI] [TP] Nom1 Nom2

Il est demandé de, soit, créer une archive .zip ou .tar.gz se décompressant dans un repertoire “BI-Nom1-Nom2”, et d’attacher cette archive en pièce jointe à l’email; soit de créer un repository Github (au nom de votre programme principal, par exemple “bioseq”) et d’envoyer le lien par email en précisant les noms du binome/monome. Dans ce repertoire (ou dans le repository), un fichier script nommé **execute-tp2** exécutera (et auparavant, compilera si besoin) tout ce qui est demandé dans les sections de ce TP nommées “Pour le compte-rendu [..]”. De telle manière qu’exécuter:

./**execute-tp2**

dans le répertoire BI-Nom1-Nom2 suffira à l’évaluateur du TP pour tester tous vos programmes. **Ce script (ainsi que le code source de votre programme) fera office de compte-rendu du TP2.**

Note: ne pas rendre ce TP2 seul. L’ensemble des TPs sera à rendre, d’un seul bloc (car il s’agira d’un seul code source commun).

Les langages acceptés sont: **Java, C, C++, Python**

Partie 1: Construction de la BWT

Télécharger la séquence du génome complet du virus **Ebola** au format FASTA sur le site du NCBI: http://www.ncbi.nlm.nih.gov/nuccore/743615855?report=fasta
et sauvegardez la dans un fichier se terminant par l'extension ".fasta".

Ecrire une fonctionalité **bwt** s'éxécutant de la manière suivante:

```
./bioseq bwt <nom du génome g>.fasta
```

qui calcule la transformée de Burrows-Wheeler du génome g. Le programme fera en sorte que la séquence de g soit terminée par le caractère $, lexicographiquement plus petit que A,C,T,G.

Le programme affichera la chaîne sur une seule ligne.

Exemple: bwt appliqué à un fichier FASTA contenant la séquence "TCGA" affichera:

AGTC$

car les rotations de TCGA$, ainsi que leur positions respectives dans la chaîne originale, sont:

TCGA$
CGA$T
GA$TC
A$TCG
$TCGA

en triant les rotations par ordre lexicographique, on obtient la BWT en lisant de haut en bas la dernière colonne:

$TCGA
A$TCG
CGA$T
GA$TC
TCGA$

Pour le compte-rendu, appliquer bwt à la séquence "GGCCATCCTTCCTGACCATTTCCATCATTCCAGTCGAACT" ainsi qu'au virus Ebola.

Optionnel: afficher les temps d'exécution.

Partie 2: Inversion de la BWT

Ecrire une fonctionalité **unbwt** s'exécutant de la manière suivante:

`./bioseq unbwt` `<nom du fichier>.bwt`

qui implémente l'algorithme d'inversion de la BWT sur la chaîne de caractères stockée dans un fichier texte. Le caractère final "$" ne sera pas affiché. L'hypothèse est que, dans ce fichier, il n'y a qu'une seule ligne contenant la BWT d'un génome. Note: l'entrée n'est PAS un fichier FASTA.

bwt affichera la séquence sur une seule ligne.

Exemple: "`./bioseq unbwt`" appliqué à un fichier texte contenant la séquence "AGTC$" affichera:

TCGA

Pour le compte-rendu, appliquer unbwt aux deux sorties de la partie précédente (BWT(GGCCATCCTTCCTGACCATTTCCATCATTCCAGTCGAACT$) et BWT(génome Ebola)). Notez que la commande:

```
./bioseq bwt genome.fasta > genome.bwt
```

permet de rediriger la sortie vers un fichier.

Optionnel: afficher le temps d'exécution de chaque inversion.

Optionnel: pour aller plus loin, regardez pour le virus Ebola si la BWT permet de mieux compresser le virus. Pour cela, utiliser gzip sur le fichier FASTA d'Ebola puis sur le fichier BWT d'Ebola que vous aurez généré à la partie 1. Qu'observez-vous? Comparer avec le résultat de bzip (compresseur basé sur la BWT) sur le fichier FASTA.