Mini projet 1 : L'alignement des séquences et la programmation dynamique

Professeur : Tom Lenaerts (tlenaert@ulb.ac.be)
Assistant : Charlotte Nachtegael (Charlotte.Nachtegael@ulb.ac.be)

Information liée au cours : http://www.ulb.ac.be/di/map/tlenaert/

Date limite : le 12 oct. 2018 à 12h

Pour ce projet, l'objectif est d'implémenter les algorithmes de *Needleman-Wunsch* et de *Smith-Waterman*, de les comparer au logiciel LALIGN[1] et présenter **un rapport scientifique** contenant votre code et vos résultats dans un Jupyter notebook[2].

Pour mieux comprendre comment faire l'alignement de séquences, nous rajoutons aussi l'article de S.R. eddy « *What is dynamic programming ?* », que vous pourriez lire.

Toutes les expériences cette année sont fait en utilisant les *Bromo* domaines et des protéines qui contiennent des *Bromo* domains. Des informations concernant ce domaine protéique sont disponible dans l'article « Filippakopoulos_et_al-2012-FEBS_Letters.pdf »

Ce projet est divisé en trois parties, expliquées plus en détail ci-dessous.

Exigences

1. Le Jupyter notebook que vous construisez est un rapport, ce qui signifie que vous devriez le structurer comme un rapport, même si le code est directement disponible.
2. Un rapport se compose d'une introduction du problème, d'une explication des méthodes (et leurs implémentations), d'une discussion sur les résultats et enfin d'une conclusion sur les résultats que vous avez obtenus.
3. Toutes les questions posées dans ce document doivent être clairement répondues et les résultats doivent être présentés afin qu'ils puissent être reproduits dans le Jupyter notebook (pas d'exécution dans un terminal)
4. Des captures d'écran de la sortie du terminal sont pas acceptable et vous ne pouvez pas faire du *copy-paste* des diapos du cours.
5. **Les explications dehors du code ne sont pas une documentation du code mais une description explicative d'algorithme : qu'est-ce que la fonction ou l'ensemble de**

[1] http://www.ch.embnet.org/software/LALIGN_form.html
[2] http://jupyter.org/install.html ou https://www.anaconda.com/download/

fonctions fait. De telles explications contiennent des exemples qui illustrent vos propos.

6. **Un rapport est un document formel. On utilise donc la première personne du pluriel, pas la première personne du singulier.**

Évaluation

L'évaluation sera basée sur les critères suivants :

1. La compréhension générale des instructions et exigences,
2. L'utilisation correcte du langage de programmation,
3. La structure du rapport et l'organisation des blocs de code dans le *Jupyter notebook,*
4. L'efficacité et l'exactitude de l'algorithme mis en œuvre,
5. La clarté et la pertinence des commentaires par bloc de code et en général,
6. La clarté des exemples utilisés pour l'illustration du fonctionnement de votre code,
7. La clarté de la comparaison faite avec d'autres outils,
8. Les illustrations graphiques.

Données

Les fichiers nécessaires pour tester votre code et l'évaluation de l'alignement peuvent être trouvés sur l'université virtuelle :

1. Les matrices de substitution blosum62, blosum80, pam60 et pam120.
2. Les séquences au format FASTA pour le *global alignment* : `BRD-sequence.fasta`
3. Les séquences au format FASTA pour le *local alignment* : `protein-sequences.fasta`

Partie 1, Abstract data types

Avant d'implémenter un algorithme qui calcule l'alignement entre deux séquences, implémentez deux ADT (Abstract Data Types) :

1. Un ADT séquence qui représente une séquence d'acides aminés et toutes les opérations qu'on peut exécuter sur une séquence qui sont nécessaires pour l'implémentation d'algorithme d'alignement (p.ex. retourner un acide aminé à une certaine position ou visualiser la séquence en format FASTA).
2. Un ADT matrice qui représente une matrice et les opérations qu'on peut exécuter sur cette matrice. Vous devez ensuite utiliser cet ADT pour créer des sous-classes matrice, pour la matrice de substitution et pour les matrices scores que vous utilisez lors de l'implémentation de l'algorithme d'alignement de séquences.

Construisez un *parser* qui peut lire :

1) Des fichiers avec des séquences (en format FASTA) et qui enregistre chaque séquence comme une instance de l'ADT séquence
2) Des fichiers qui représentent des matrices de substitution et qui enregistre chaque matrice comme une instance de l'ADT *substitution matrix*

Partie 2, Alignement global

En utilisant les ADT construits pendant l'étape précédente, implémentez en Python l'algorithme *Needleman-Wunsch* qui calcule l'alignement global en utilisant la pénalité affine.

L'alignement global doit retourner au maximum les *k* meilleurs alignements, *k* étant un paramètre dans l'appel de fonction que vous êtes en train d'implémenter.

Comparez vos résultats avec les résultats produit par LALIGN[1]. Montrez quelques résultats de LALIGN comme preuve et expliquez les similarités et différences.

Utilisez les séquences du fichier `BRD-sequence.fasta`. Lorsque vous alignez les séquences BRD, lesquelles sont les plus similaires ? Viennent-elles de la même protéine ou de protéines différentes ? Consultez ceci via le site Uniprot[3] et expliquez ceci dans votre Jupyter notebook.
Dans l'article « Filippakopoulos_et_al-2012-FEBS_Letters.pdf » vous trouverez de l'information additionnelle concernant les domaines que vous pouvez utiliser dans votre rapport.

Partie 3, Alignement local

Modifiez le logiciel de la partie 2 de sorte qu'on puisse faire un alignement local (*Smith-Waterman*).

L'alignement local doit retourner au maximum *l* alignements, *l* étant un paramètre dans l'appel de fonction que vous êtes en train d'implémenter.

Comparez vos résultats avec l'outil LALIGN[1]. Montrez quelques résultats de LALIGN comme preuve et expliquez les similarités et différences

Utilisez les séquences du fichier `protein-sequences.fasta` et détectez quelles sont les parties similaires. Expliquez les similarités en utilisant les informations sur les protéines venant du site UniProt[2].

Éthique

Le plagiat sera sévèrement sanctionné. Les cas de plagiat comprennent la réutilisation du matériel écrit ou tiré de quelqu'un d'autre[4], ou tout type de travail, sans devis ou référence explicite.

---

[3] http://uniprot.org

[4] http://www.bib.ulb.ac.be/fr/aide/eviter-le-plagiat/ et http://www.plagiarism.org/