

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

Professeur Tom Lenaerts
Assistant : Catharina Olsen

Information additionnelle sur :
http://www.ulb.ac.be/di/map/tlenaert/Home_Tom_Lenaerts/INFO-F-208.html

Partie. 1.1

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

1. un ADT `séquence` qui représente une séquence d'acides aminés et tous les opérations qu'on peut exécuter sur une séquence.
2. un ADT `score` qui représente une matrice de substitution et les opérations qu'on peut exécuter sur cette matrice.

Voici un lien vers un site qui peut créer des matrices de substitution:

- <http://blocks.fhcrc.org/blocks/uploads/blosum/>

Utilisez les séquences dans le fichier `SH4-sequences.fasta` et `maguk-sequences.fasta` sur le site web pour l'évaluation de votre *parser*.

Partie 1.2

En utilisant les ADT construits pendant l'étape précédente, vous implémentez en Python l'algorithme Needleman-Wunsch qui calcule l'alignement global en utilisant la pénalité affine. L'algorithme renvoie tous les meilleurs alignements.

Vérifiez si votre programme donne le même résultat que l'outil LALIGN
http://www.ch.embnet.org/software/LALIGN_form.html

Utilisez les séquences dans le fichier `SH3-sequences.fasta` sur le site web.

Partie 1.3

Modifiez le logiciel de la partie 1.2 de sorte que on peut faire un alignement local (Smith-Waterman).

Utilisez les séquences dans le fichier `maguk-sequences.fasta` sur le site web. Retrouvez les similarités entre les 4 séquences. Expliquez les similarités. Plus d'informations concernant les protéines dans le fichier `.fasta` peuvent être trouvées sur le site UniProt (<http://www.uniprot.org/>).

Vérifiez encore une fois vos résultats avec l'outil LALIGN.