

Application de la programmation dynamique à l'alignement de séquences

Exercice 1 : (Alignement global (version récursive))

Implémentez en Python une version récursive pour calculer l'alignement global de deux séquences. Pour cela,

1. Récupérez d'abord une matrice de score de mutation, par exemple à l'adresse : <http://www.i3s.unice.fr/~comet/SUPPORTS/>.
2. Ecrivez une fonction qui lit un tel fichier et stocke dans un dictionnaire les valeurs lues.
3. Ecrivez l'algorithme récursif qui met en oeuvre la récurrence décrite en cours pour l'alignement de séquences global à pénalité linéaire des gaps.
4. Testez votre fonction sur des séquences de plus en plus grandes.

Exercice 2 : (Alignement global (version itérative))

La complexité excessive de l'algorithme précédent ne permet pas d'envisager une recherche de similarité pour des séquences de longueur supérieure à 15. La version itérative de l'algorithme se base sur le remplissage systématique d'un tableau permettant ainsi une mémorisation des scores intermédiaires.

1. Quelle structure utiliser pour stocker les scores intermédiaires ?
2. Comment initialiser les premières ligne et colonne de ce tableau ?
3. Implémentez l'algorithme de Needleman & Wunsch.
4. Développez une fonction qui prend en entrée le tableau complet des scores intermédiaires et qui retourne 3 chaînes de caractères représentant l'alignement. Par exemple, pour l'alignement (blosum62, g=5)

```
-KKR-SDNEDKRWKF
  ||:   |::||:
  KKKKPKKEEQKWKW
```

les 3 chaînes de caractères seront respectivement : “-KKR-SDNEDKRWKF”, “ ||: |::||:” et “KKKKPKKEEQKWKW”.

Exercice 3 : (Alignement global avec pénalité affine)

Modifiez l'algorithme de l'exercice précédent pour développer l'algorithme de Needleman & Wunsch avec pénalité affine des gaps.

Exercice 4 : (Alignement local avec pénalité affine)

Modifiez l'algorithme de l'exercice précédent pour développer l'algorithme de Smith & Waterman avec pénalité affine des gaps.