Compte-rendu

Algorithmes de programmation dynamique pour l'alignement de séquences

EZ-ZEJJARI Imad

BENLHABIB Youssef

AIT BAHESSOU Haytham

TAGHY Soufiane

I. Algorithme de Smith-Waterman

On a implémenté cet algorithme comme demandé dans l'énoncé, en utilisant une seule matrice contenant les cellules avec leurs scores et précédents.
On traite les séquences d'ADN avec un modèle de coût linéaire pour les indels. On construit puis on affiche l'un des alignements optimaux.

II. Algorithme de Altschul & Erickson

Dans cet algorithme, on a choisi d'utiliser une seule matrice où on stocke dans une même matrice les trois scores D/H/V.

Pour afficher l'alignement, on a utilisé ultimatePrev qui prend trois états 1/2/4 et montre la direction où on a obtenu le résultat précédent. Cette méthode est codée dans `aliPrintBestAlisOld` dans aliOutAE.

III. Application: épissage

- Exemple_splicing

1. Interprétation des résultats (chromosome 10 vs 17):

Le chromosome 10 donne 1 résultat de petite taille et de score 80 dans le modèle affine, alors que dans le modèle linéaire, on obtient un score de 149. Ceci peut être interprété en valeur absolue par la faiblesse du coût des indels linéaires qui offrent une possibilité de trouver des matchs, en comparaison avec le modèle affine où on a un coût plus grand pour indelOpen de 100, et dont il est moins probable de faire -100.

Pour le chromosome 17, le score est plus grand dans le modèle affine que dans le modèle linéaire, ceci est expliqué par le fait que la plupart des gaps viennent de indelExtend qui ne diminue que 0.05, alors que dans le chromosome 10.

La plupart des gaps viennent de indelopen. Pour le modèle linéaire malgré qu'on obtient tous des matchs, le nombre des matchs n'est pas élevé pour avoir un score plus grand que dans le modèle affine.

On obtient plusieurs résultats dans le modèle affine pour 17 que dans le modèle linéaire, et on remarque que cet effet est inversé pour le 10.

2. Le modèle de coût le mieux adapté à ce type de problème:

Le modèle affine est plus adapté à ce type de problème.
Dans la plupart des cas, considérer que l'insertion d'un gap possède un coût constant ne correspond pas à un modèle réaliste. On préférera un modèle pour lequel un gap de longueur k est plus probable que k gaps de longueur 1.
Ce qui est le cas pour le chromosome 17 car on obtient des gaps continus constituant un seul gap, et dont le coût en valeur absolue n'est pas trop élevé pour chacun des gaps car on ne diminue que 0.05, alors que dans le modèle linéaire plusieurs gaps éloignés ont un coût plus grand.
Comme la complexité des deux modèles n'est pas la même: le modèle de gap affine n'augmente pas la complexité du problème d'alignement. Les deux ont $O(n^2)$.

On obtient un fait une vingtaine d'alignements optimaux pour l'alignement vs le chromosome 17. On sait par ailleurs que la plupart des introns commencent par GT et se terminent par AG. A la lumière de cette information:

3. Analyse détaillée de ces alignements:

La plupart des alignements commencent par TG et se terminent par AC, ce qui nous amène à remarquer que deux grandes parties des chaînes au début et à la fin des alignements sont similaires.
On comprend donc que la différence entre les alignements se situe dans le milieu des chaînes, où il y a de longues indels, mais vu qu'elles sont amorties, ceci revient au coût d'une seule indel, donnant ainsi plus de chance et de liberté pour trouver des matchs augmentant le score.

IV. Extension 1 : séquences peptidiques

- Exemple_Hemoglobin

1. Comparaison des résultats de alpha vs beta avec ceux de l'exemple:

Nos résultats contiennent tous les alignements possibles, conformément à l'exécutable fourni aliProtAffine_ntm qui confirme la validité de notre programme.

Par contre, les résultats de la page :
http://emboss.sourceforge.net/apps/release/6.6/emboss/apps/water.html
ne donnent qu'un seul alignement, ceci est peut être dû au fait que le programme s'arrête au premier score maximal rencontré.

2. Interprétation des résultats:

Concernant les alignements des sous-unités alpha/beta et beta/zeta, on constate qu'il y a plusieurs alignements qui donnent le meilleur score.
En effet, à un moment donné, le chemin se duplique sur une petite séquence, puis se fusionne en un seul alignement. Ceci peut être interprété par le phénomène de duplication.
De plus, à la fin de l'alignement, s'ajoute un mismatch (R/H) de coût nul, ce qui ne diminue pas le score final. On peut dire qu'une évolution a eu lieu.

Cependant, les séquences de alpha/zeta ne donnent qu'un seul alignement de score optimal élevé par rapport à celui des alignements de alpha/beta

et beta/zeta. Ainsi, la réduction du score maximal est, d'une manière ou d'une autre, liée à une duplication.

V. Extension 2

Cette extension est codée dans la fonction de aliOutAE **`aliPrintBestAlis.`** Pour afficher tous les alignements optimaux, on a utilisé la récursivité qui est utilisée à chaque fois qu'on trouve plusieurs possibilités. Malgré que le code paraît un peu long, cependant il donne les résultats voulus après plusieurs tests et débugs.