IFT3295 - Bio-Informatique

Professeur: Nadia El-Mabrouk

Travail Pratique #1

27 septembre 2016

Nom: Paul CHAFFANET
Code Permanent: CHAP23049307
Matricule: 1009543

Exercice 1

Question 1

Comme nous cherchons une similarité maximale lors d'un chevauchement, la solution optimale se trouve soit dans la dernière colonne j soit dans la dernière ligne i qui marque la fin d'une séquence Y et X respectivement.

Dans le case de l'alignement local, nous cherchons une ou des sous-séquences de X et Y ayant la similarité la plus maximale. Ainsi, nous n'avons pas la contrainte de devoir atteindre l'une des extrémité de X ou de Y : le score de similarité maximale dans l'alignement local se trouve dans la cellule (i, j) ayant la valeur maximale dans la table de programmation dynamique.

On ne pénalise pas l'alignement avant le début du chevauchage. On donne donc des coups gratuits aux cases $(i,0)$ et $(0,j)$ qui permettent d'accèder à chacune des base des séquences X et Y (coût = 0) comme dans l'alignement local. Commencer à la cellule $(i,0)$ marquera immanquablement le début de de la séquence Y (donc le début d'un chevauchement), et commencer à la cellule $(0,j)$ celui de la séquence X.

En revanche, la case (i, j) ne peut pas être mise à 0 lorsque le score de similarité est négatif (contrairement à l'alignement local qui vaut 0 si le score de similarité est négatif). En effet, ce score doit tenir compte du meilleur chevauchement entre les séquences, on ne peut pas commencer notre alignement de n'importe où comme dans le cas de l'alignement local.

Question 2

$(V(0,j), \forall j) = 0$
$(V(i,0), \forall i) = 0$

En effet, nous voulons étudier seulement le meilleur chevauchement des séquences X et Y. Pour cela, nous voulons accéder gratuitement à n'importe quelle base de X ou de Y gratuitement de manière verticale et horizontale respectivement.

Ainsi, nous pouvons ignorer toute une sous-séquence de X ou de X qui ne se chevauchent pas et ainsi nous pouvons économiser les pénalités liées au score d'indel avec un alignement de caractère. Les coût 0 permettant alors d'ignorer les coûts d'indels lorsqu'une séquence X ou Y n'a pas encore commencé à se chevaucher.

Ensuite, commencer à la cellule $(i,0)$ marquera immanquablement le début de de la séquence Y (donc le début d'un chevauchement), et commencer à la cellule $(0,j)$ celui de la séquence X. Ainsi, c'est à partir du moment où X chevauche Y que nous comptabiliserons son score de similarité de chevauchement.

Question 3

$$(V(0,j),\forall j)=0$$
$$(V(i,0),\forall i)=0$$

$$V(i,j)=\max\begin{cases}V(i-1,j)+\delta(x_i,-)\\V(i,j-1)+\delta(-,y_j)\\V(i-1,j-1)+\delta(x_i,y_j)\end{cases}$$

Question 4

Voir le code fourni. Il suffit de le compiler pour voir s'afficher dans la console la réponse à la plupart des questions posées dans ce devoir. Les chaînes x et y ainsi que les scores sont modifiables dans la méthode main() directement dans le code.

Question 5

Voir le code fourni.

		A	C	G	C	A	T	T
	0	0	0	0	0	0	0	0
G	0	−1	−1	1	−1	−1	−1	−1
A	0	1	−1	−1	0	0	−2	−2
T	0	−1	0	−2	−2	−1	1	−1
A	0	1	−1	−1	−3	−1	−1	0
C	0	−1	2	0	0	−2	−2	−2
G	0	−1	0	3	1	−1	−3	−3
T	0	−1	−2	1	2	0	0	−2
C	0	−1	0	−1	2	1	−1	−1
A	0	1	−1	−1	0	3	1	−1
C	0	−1	2	0	0	1	2	0
G	0	−1	0	3	1	−1	0	1
T	0	−1	−2	1	2	0	0	1
G	0	−1	−2	−1	0	1	−1	−1
C	0	−1	0	−2	0	−1	0	−2
A	0	1	−1	−1	−2	1	−1	−1
C	0	−1	2	0	0	−1	0	−2
G	0	−1	0	3	1	−1	−2	−1
G	0	−1	−2	1	2	0	−2	−3

```
X: GATACGTCACGTGCACGG---
Y: --------------ACGCATT
Score: 2
```

Question 6

Utiliser des score positifs pour les mismatchs n'est pas une bonne idée. En effet, l'algorithme privilégiera tout ce qui peut faire augmenter son score de similarité, et donc préfèrera ajouter des mismatchs à la séquence de chevauchement afin d'augmenter le score de similarité.

Ainsi, un chevauchement de séquence X et Y ne contenant que des mismatch peut être considéré comme de similarité maximale, ce qui n'est pas un chevauchement qui nous intéresse vraiment afin de faire un assemblage de séquences. Les séquences risquent également d'être carrément aligné le plus entièrement possible.

Exercice 2

Question 1

Voir code fourni.

```
S1: AACTCTCTACTGCTTTCCCC------
S2: ------CTACTGCTTTCCCCGCCGGA
Score: 14

S1: AACTCTCTACTGCTTTCCCC----------
S3: ----------CTTTCCCCGCCGGAACCTTCAC
Score: 8

S1: --------AACTCTCTACTGCTTTCCCC
S4: TAAATTACAACTCTCTACTA--------
Score: 10

S2: CTACTGCTTTCCCCGCCGGA--------
S3: ------CTTTCCCCGCCGGAACCTTCAC
Score: 14

S2: ----------CTACTGCTTTCCCCGCCGGA
S4: TAAATTACAACTCTCTACTA----------
Score: 4

S3: ----------------CTTTCCCCGCCGGAACCTTCAC
S4: TAAATTACAACTCTCTACTA----------------
Score: 1
```

Question 2

```
Assemblage:

S1: --------AACTCTCTACTGCTTTCCCC----------
S2: ----------CTACTGCTTTCCCCGCCGGA--------
S3: ----------------CTTTCCCCGCCGGAACCTTCAC
S4: TAAATTACAACTCTCTACTA----------------
S:  TAAATTACAACTCTCTACTGCTTTCCCCGCCGGAACCTTCAC
```

Le réponse est 42 pour la longueur du contig final.

Exercice 3 (voir code fourni.)

Question 1

Frame 1:
LSYRA <STOP> ITTLYCFPRRNLHTSHTYVSEQPGAGGRLSLLQAFPERIQLESV <STOP>
<STOP> CRRE <STOP> D <STOP> GPRRN <STOP> SREGDPQCHQWQPILAPGG
<STOP> PGREWSHWPQQQFGCAGGDSHGSSEASAERGRR <STOP> V <STOP>
TAVPESVQ <STOP> SNIPASHNPRDRVSEL <STOP> AGSE <STOP>
TLSGWSKLGSHRGLFLLWRGTVRGKRRQGDAGIGESDCKLDGHLSE <STOP>
PPRALDPGERRLGHFCGSLREQCSSREPERPGALQPLVPDGHDCGWCGSAGLTLQSEVGRWYGM
SVSKKKKKKKK

Séquence protéinique candidate: MSVSKKKKKKK

Frame 2:
SPTELKLQLSTAFPAGTFTPVTRMSQSNRELVVDFLSYKLSQKGYSWSQFSDVEENRTEAPEET
EAERETPSAINGNPSWHLADSPAVNGATGHSSSLDAREVIPMAAVKQALREAGDEFELRYRRAF
SDLTSQLHITPGTAYQSFEQVVNELFRDGVNWGRIVAFFSFGGALCVESVDKEMQVLVSRIASW
MATYLNDHLEPWIQENGGWDTFVDLYGNNAAAESRKGQERFNRWFLTGMTVAGVVLLGSLFSRK
<STOP> VVGMV <STOP> V <STOP> VRKKKKKKK

Séquence protéinique candidate:
MSQSNRELVVDFLSYKLSQKGYSWSQFSDVEENRTEAPEETEAERETPSAINGNPSWHLADSPA
VNGATGHSSSLDAREVIPMAAVKQALREAGDEFELRYRRAFSDLTSQLHITPGTAYQSFEQVVN
ELFRDGVNWGRIVAFFSFGGALCVESVDKEMQVLVSRIASWMATYLNDHLEPWIQENGGWDTFV
DLYGNNAAAESRKGQERFNRWFLTGMTVAGVVLLGSLFSRK

Frame 3:
LLQSLNYNSLLLSPPEPSHQSHVCLRATGSWWSTFSPTSFPRKDTAGVSLVMSKRIGLRPQKKL
KQRGRPPVPSMATHPGTWRIARP <STOP> MEPLATAAVWMRGR <STOP> FPWQQ
<STOP> SKR <STOP> ERQAMSLNCGTGERSVI <STOP> HPSFT <STOP>
PQGPRIRALSR <STOP> <STOP> MNSFGME <STOP> TGVASWPFSPLAGHCAWKA
<STOP> TRRCRYW <STOP> VGLQVGWPPI <STOP> MTT <STOP>
SLGSRRTAAGTLLWISTGTMQQPRAGKARSASTAGS <STOP> RA <STOP>
LWLVWFCWAHSSVGSRSLVWYECK <STOP> EKKKKKK

Séquence protéinique candidate:
MSKRIGLRPQKKLKQRGRPPVPSMATHPGTWRIARP

Séquence protéinique probable:
MSQSNRELVVDFLSYKLSQKGYSWSQFSDVEENRTEAPEETEAERETPSAINGNPSWHLADSPA
VNGATGHSSSLDAREVIPMAAVKQALREAGDEFELRYRRAFSDLTSQLHITPGTAYQSFEQVVN
ELFRDGVNWGRIVAFFSFGGALCVESVDKEMQVLVSRIASWMATYLNDHLEPWIQENGGWDTFV
DLYGNNAAAESRKGQERFNRWFLTGMTVAGVVLLGSLFSRK

Question 2

Longueur de la protéine X: 233

Question 3

- **Nom de la protéine**: BCL2-like 1
- **Espèce d'origine**: Mus musculus (Souris grise)
- **Fonction**: Membre de la famille Bcl-2 des régulateurs de l'apostose(mort cellulaire programmée) chez la souris. Cette protéine se lie à des canaux anioniques voltage-dépendants dans la membrane mitochondriale externe afin de pouvoir faciliter l'absorption des ions calcium.
- **Position génomique**: sur le chromosome 2 à la position 2 75.41 cM; 2 H1

Exercice 4 (voir code fourni.)

Question 1

La case $D(i, j)$ représente l'alignement de score maximal entre tous les sous-séquences de l'intervalle $X[x_1,...,x_i]$ avec $Y[y_1,...,y_j]$ en entier.

Question 2

$$D(0, j) = j * \delta(-, y_j), \forall j$$
$$D(i, 0) = 0, \forall i$$

On pénalise le fait de vouloir insérer une base X avant une base Y, car nous voulons aligner Y en entier. On pourrait dans l'absolue avoir coûts tels $D(0, j) = -\infty$ dans le cas de scores d'indels positifs par exemple.
On peut commencer l'alignement avec Y à partir de n'importe quelle base de X sans pénalité (coût gratuit) puisque que nous recherchons un facteur de X ayant l'alignement maximal avec Y.

Question 3

On doit regarder le score maximal dans la dernière colonne correspond à la fin de l'énumération complète de la séquence Y, car cela nous assure que nous avons aligné Y en entier de manière maximale. La dernière ligne ne nous intéresse pas car dans ce cas Y ne serait pas entièrement aligné.

Question 4

Soit (i', j) la cellule qui contient le score d'alignement maximal du chevauchement tel que $0 \leq i' \leq n$, $j = m$ dans une table de programmation dynamique de dimension $n * m$. A cette position, nous nous sommes assurés que la séquence Y de taille m a été énumérée complètement.

A partir de cette case, nous allons remonter jusqu'à une case $(i,0)$ correspondant au début de l'énumération de la séquence Y . Arrivé là, nous pouvons conclure que le facteur $X' : x_i ... x_{i'}$ a le plus haut score d'alignement avec la séquence complète Y .

Question 5

```
Table de programmation dynamique de X et Y pour le meilleur
alignement entre un facteur de X et Y:

0   -2   -4   -6   -8  -10  -12  -14
0   -1   -3   -3   -5   -7   -9  -11
0    1   -1   -3   -4   -4   -6   -8
0   -1    0   -2   -4   -5   -3   -5
0    1   -1   -1   -3   -3   -5   -4
0   -1    2    0    0   -2   -4   -6
0   -1    0    3    1   -1   -3   -5
0   -1   -2    1    2    0    0   -2
0   -1    0   -1    2    1   -1   -1
0    1   -1   -1    0    3    1   -1
0   -1    2    0    0    1    2    0
0   -1    0    3    1   -1    0    1
0   -1   -2    1    2    0    0    1
0   -1   -2   -1    0    1   -1   -1
0   -1    0   -2    0   -1    0   -2
0    1   -1   -1   -2    1   -1   -1
0   -1    2    0    0   -1    0   -2
0   -1    0    3    1   -1   -2   -1
0   -1   -2    1    2    0   -2   -3

Alignement:
X: GATACGTCACGTGCACGG
Y: ---ACG-CATT-------
Score: 1
```

Question 6

Calculer la valeur $D(i, j)$ prend un temps constant $\mathrm{O}(1)$;
Calculer chaque valeur $D(i, j)$ de la matrice de dimension $n * m$ prend un temps $\mathrm{O}(nm)$.
Donc l'algorithme est en $\mathrm{O}(nm)$ dans notre cas.

L'algorithme consistant à reveler le score d'alignement maximal entre chaque facteur de X' et Y prend plus de temps. Soit X un mot de taille n et Y un mot de taille m : on compte le nombre de séquences possible dans X , et on lui associe aussi sa taille.
Ainsi, pour $n = 10$, 1 facteur de longueur 10, deux facteurs différents de longueurs 9, 3 facteurs de longueurs 8 différents etc.
Pour chaque facteur, afin de trouver son score d'alignement maximal, cela a un coût de *(longueur du facteur X')* m* , où

On obtient donc la somme suivante qui correspond à la complexité de l'algorithme naïf:

$$m \times \sum_{i=1}^{n} (n-i) \times i \quad , \forall n \geq 2$$

$$= m \times \frac{1}{6} n(n^2 - 1) = \mathrm{O}(n^3 m)$$

Ainsi on voit bien que l'algo naïf consistant à énumérer tous les facteurs de X et les aligner avec Y a une plus grande complexité par rapport à notre algorithme.