

Nom, Prénom : Boukharouba Boumediene

Matricule : 20032279

Nom, prénom: Kassis Charbel

Matricule: 0976458

---

**Repliement d'ARNs en tige-boucle (45pts)****1- À quoi correspond les cellules de M sur l'anti-diagonale :**

Les cellules de M sur l'anti-diagonal correspondent au score d'un certain alignement de nucléotide, on déduit le score maximal de similarité à partir de cette même anti-diagonale. Il n'est pas nécessaire de remplir toute la table mais seulement une moitié.

**2- Donnez les équations d'initialiser et de récurrence pour remplir la table M**

\*les valeurs de la première lignes et la première colonne est égale à 0

**Cas initial:**  $\forall i, j \quad V(i,0) = V(j,0) = 0$

$$V(i,j) = \max \left\{ \begin{array}{l} v(i-1,j) \\ v(i,j-1) \\ \{ v(i-1, j-1) + 1 \text{ si match} \\ v(i-1,j-1) \text{ si mismatch} \end{array} \right.$$

**3- Décrivez comment on peut retrouver un repliement en tige-boucle maximisant les appariements de nucléotides.**

On remplit la table via les équations de récurrence ci-dessus et on va chercher le score le plus haut se trouvant sur l'anti-diagonale, ensuite on remonte ( backTracking ) jusqu'à  $M(0,0)$ .

#### **4- Dans le fichier Question4Repliment.png**

**5- Pour être stable, la boucle de la tige-boucle doit contenir au moins 3 nucléotides. Décrivez une façon de modifier votre algorithme afin d'avoir des tige-boucles avec au moins 3 nucléotides :**

Pour avoir au moins trois nucléotides dans la boucle, on doit ne pas prendre en compte une anti diagonale en comptant celle déjà ignoré. En effet, ces dernières représentent des appariements des nucléotides avec eux même. Et lors de la recherche de la valeur maximal on va plutôt ajuster le résultat en choisissons valeur maximal  $-1$ .

**Décrivez une modification de votre algorithme qui permet les appariement (G-U)**

Il suffit d'ajouter une modification qui met en évidence que le cas de match est plus generale en acceptant G-U \* c'est-à-dire C-U ou G-A

$$V(i,j) = \max \left\{ \begin{array}{l} v(i-1,j) \\ v(i,j-1) \\ \{ v(i-1, j-1) + 1 \text{ si match} \\ v(i-1, j-1) + 1 \text{ si C-U ou G-A} \\ v(i-1,j-1) \text{ si mismatch} \end{array} \right.$$

**7- Dans cette section, on désire améliorer l'algorithme précédent en considérant des scores d'empilement d'appariement, ainsi que des pénalités pour les nucléotides non-apparié**

Il faut à chaque match voir si c'est un empilement G-C ou A-U, Puis voir l'empilement precedent si c'Est A-U ou G-C..

Cas initial :  $\forall i,j \quad V(i,0) = V(j,0) = 0$

$V(i,j) = \max \{ \quad v(i-1,j)$

$v(i,j-1)$

$v(i-1, j-1) + p(S[i],S[j])$

Avec  $p(S[i],S[j]) = 2$  si  $(i,j) == G,G \mid \mid C,C \ \&\& \ (i-1,j-1) == G,G \mid \mid C,C$

$p(S[i],S[j]) = 1$  si  $(i,j) == G,G \mid \mid C,C \ \&\& \ (i-1,j-1) == A,A \mid \mid U,U$

$p(S[i],S[j]) = 1$  si  $(i,j) == A,A \mid \mid U,U \ \&\& \ (i-1,j-1) == G,G \mid \mid C,C$

Sinon 0

**(b)- Quelle est la complexité de votre algorithme en temps et en espace ?**

**En temps :  $O(n^2)$**

**En espace:  $O(n^2)$**

**C) Score : 10**

### **Alignement multiple de séquences (55pts)**

Les questions 1-5 ont été accomplies par programmation en python.

Le fichier QuestionsAlignementMultiple.py contient le code qui imprimera les réponses de chaque question sur la console.

A noté :

- 1) S'assurer que tous les fichiers fournis restent dans le même répertoire
- 2) Le programme principal (QuestionsAlignementMultiple.py) va automatiquement parser sequences.fasta (Sequence.py) et BLOSUM62.txt (MatrixScore.py) et les stocker en mémoire. Il appellera la fonction alignementGlobal (AlignementGlobal.py) sur chaque pair de séquences.
- 3) Pour lancer le programme (le programme a été testé sur python version 3):

python3 QuestionAlignementMultiple.py

- 4) La dernière question génère un fichier (frequences.txt) qui contient la table de fréquences des caractères. Un petit morceau sera affiché dans la console.

### Outils bioinformatique (15pts)

1)

s1 : [hemoglobin 2 \[Arabidopsis thaliana\]](#)

s2 : [cytoglobin \[Bos taurus\]](#)

s3 : [hemoglobin subunit alpha-A \[Gallus gallus\]](#)

s4 : [myoglobin \[synthetic construct\]](#)

s5 : [neuroglobin \[Rattus norvegicus\]](#)

2)