# BIN702 - Série d'exercices sur l'alignement multiple

Manuel Lafond

**Exercice 1:** Soient les séquences

$$S_1 = AGTA$$
$$S_2 = AGCTA$$
$$S_3 = TCA$$
$$S_4 = TGA$$

a. Considérez l'algorithme de centre-étoile vu en classe. On suppose qu'on choisit la séquence centrale qui minimise la somme des distances Levenshtein avec les autres, et on utilise la version minimisation du coût SP. On suppose un coût de 1 pour toute mutation et tout caractère aligné avec un gap (un gap avec un gap a un coût de 0). Quelle serait une séquence centrale possible ? Donnez le résultat de l'algorithme selon la séquence centrale choisie.

b. Considérez l'algorithme de Clustal pour maximiser le score SP, avec un score de $+1$ pour un match, $-1$ pour une mutation ou un caractère aligné avec un gap, et 0 pour un gap avec un gap. Supposez que vous avez l'arbre guide dans lequel $S_1$ et $S_2$ ont un parent commun, et $S_3$ et $S_4$ ont un autre parent commun (et ces deux parents sont joints par la racine).

   Donnez l'alignement multiple retourné par Clustal suite au parcours de l'arbre guide. Pour bien faire cet exercice, vous devriez construire la table de programmation dynamique pour combiner les deux alignements du 2ème niveau de l'arbre guide.

**Exercice 2:** Écrivez les conditions initiales et la récurrence servant à calculer l'alignement multiple optimal entre 3 séquences $S_1, S_2$ et $S_3$.

**Exercice 3:** Quelle est la complexité de l'algorithme exact vu en classe qui calcule l'alignement exact optimal entre $m$ séquences ? Vous pouvez supposer que toutes les séquences ont une longueur de $n$.
Notez qu'en classe, j'ai donné une borne de $n^m 2^m$. C'est presque ça, mais il manquait un petit détail dans cette analyse.

**Exercice 4:** Dans l'algorithme de Clustal, nos arbres guide étaient toujours binaires. Supposons que l'arbre est plutôt ternaire, i.e. chaque noeud a trois enfants. Comment peut-on modifier Clustal pour supporter un arbre guide ternaire ?

**Exercice 5** (Difficile)**:** Dans cette question, nous allons montrer que l'approche centre-étoile est une 2-approximation. Cette question est complèment optionnelle et ne vous préparera pas réellement à l'examen final.
Soit $S_1, \ldots, S_m$ un ensemble de $m$ séquences. On veut minimiser le coût SP, qui est la somme des distances Levenshtein sur toutes les paires de séquences alignées. Soit $A$ l'alignement multiple retourné par l'algorithme centre-étoile, et soit $A^*$ l'alignement multiple optimal. Montrez que

$$cout(A) \leq 2cout(A^*)$$

où $cout(A)$ est le coût SP de l'alignement $A$ (même chose pour $cout(A^*)$).
Notez que la distance Levenshtein satisfait l'inégalité triangulaire. C'est-à-dire, si $D(S_i, S_j)$ représente la distance entre $S_i$ et $S_j$, alors pour $S_i, S_j, S_k$ on a $D(S_i, S_k) \leq D(S_i, S_k) + D(S_k, S_i)$.

Plus précisément, si $S_c$ est la séquence centrale, vous pouvez montrer que et que

$$cout(A) \leq m \sum_{i=1}^{m} D(S_c, S_i) \qquad \text{et} \qquad cout(A^*) \geq \frac{m}{2} \sum_{i=1}^{m} D(S_c, S_i)$$

Pour la première inégalité, vous pouvez utiliser le fait que grâce à l'inégalité triangulaire, on a $dist_A(S_i, S_j) \leq dist_A(S_i, S_c) + dist_A(S_c, S_j) = D(S_c, S_i) + D(S_c, S_j)$.