DÉPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPÉRATIONNELLE

SIGLE DU COURS : IFT3295 (Aut2018)

NOM DU PROFESSEUR : Nadia El-Mabrouk

TITRE DU COURS : Bio-Informatique

EXAMEN FINAL

Date : Mardi 18 décembre 2018
Heure : 13h30 - 16h20
Salle : 1360 Pav. A.Aisenstadt

DIRECTIVES PÉDAGOGIQUES :

- **Aucune documentation** n'est permise.
- **Inscrivez tout de suite votre nom et matricule** en bas de cette page.
- Répondez **sur le questionnaire** en utilisant l'espace libre qui suit chaque question.
- Lisez attentivement toutes les questions avant de commencer.
- Les exercices sont indépendants les uns des autres.

1. _____ /20
2. _____ /15
3. _____ /27
4. _____ /18
5. _____ /20

Total : _____ /100

Nom : _____ **Matricule :** _____

Règlement sur le plagiat : (extrait du règlement disciplinaire sur le plagiat de l'Université de Montréal).
Constitue un plagiat :

1. Faire exécuter son travail par un autre ;
2. Utiliser, sans le mentionner, le travail d'autrui ;
3. Échanger des informations lors d'un examen ;
4. Falsifier des documents.

Le plagiat est passible de sanctions allant jusqu'à l'exclusion du programme.

Exercice 1 : (20 points)

Pour chacun des énoncés suivants indiquer s'il est vrai ou non en encerclant OUI ou NON.

1.	Un alignement multiple donnant lieu à un score induit minimal pour chaque paire de séquences est nécessairement un alignement multiple de score SP minimal.	OUI	NON
2.	La complexité en temps de l'alignement local est la même que celle de l'alignement global	OUI	NON
3.	Le problème de retrouver le plus long facteur commun de deux séquences S de taille n et T de taille m peut se résoudre en temps $O(m+n)$	OUI	NON
4.	Une pression de sélection négative est inférée pour un gène lorsque sa séquence codante révèle une distance non-synonyme (d_N) significativement inférieure à la distance synomyne (d_S) avec les séquences homologues.	OUI	NON
5.	Pour une matrice de distance quelconque, l'algorithme UPGMA donne lieu à un arbre unique.	OUI	NON
6.	Une matrice de distance ne peut pas être à la fois ultramétrique et additive.	OUI	NON
7.	Pour un arbre donné et pour un score unitaire (coût de 1 pour une substitution), l'algorithme de Fitch et l'algorithme de Sankoff peuvent donner lieu à un nombre de substitutions différent.	OUI	NON
8.	Un arbre non-raciné de n feuilles est déterminé par $n-3$ bi-partitions non-triviales.	OUI	NON
9.	Une inversion augmente d'au plus un le nombre d'adjacences d'une permutation (par rapport à la permutation identité).	OUI	NON
10.	Étant donnée une permutation signée, il existe toujours une inversion qui diminue d'au moins 1 le nombre de points de cassures (breakpoints) de la permutation.	OUI	NON

Exercice 2 : Questions diverses. (**15 points**)

1. Justifier votre réponse pour un parmi les 10 énoncés de l'exercice précédent.
   Justification de l'énoncé numéro :

2. Expliciter la condition des trois points permettant de vérifier si une distance D entre un ensemble de feuilles Σ est ultramétrique.

3. Expliciter la condition des quatre points permettant de vérifier si une distance D entre un ensemble de feuilles Σ est additive.

Exercice 3 : Repliement d'ARN et arbres des suffixes (27 points)

Soit $S[1, n]$ une séquence d'ARN de taille n. On veut trouver le repliement de S en une seule tige-boucle, ne contenant que des appariements Watson-Crick ($G-C$, $C-G$, $A-U$, $U-A$), autrement dit sans boucles multiples, ni boucles internes ni renflements, qui maximise le nombre d'appariements. De plus il n'y a aucune contrainte sur la taille de la boucle.

1. **Recherche par programmation dynamique :**

   (a) Soit $D(i, j)$ le nombre maximum d'appariements contenus dans un repliement du facteur $S[i, j]$ de S. Expliciter (1) **la relation de récurrence** et (2) **les conditions initiales** permettant de calculer la table D, et expliquer (3) **comment trouver la VALEUR du repliement maximal**, et (4) **comment retrouver ce repliement**.

   (b) Remplir la partie supérieure-droite de la table de programmation dynamique suivante, diagonale par diagonale, dans l'objectif de trouver un repliement optimal en une seule tige-boucle de la séquence $S = CCCTCAGG$. **Indiquer la case et les flèches permettant de retrouver le repliement optimal**.

$D(i,j)$	C	C	T	C	G	G	A
C							
C							
C							
T							
C							
A							
G							
G							

(c) **Expliciter le repliement** obtenu.

(d) Donner la complexité en temps et en espace de l'algorithme.

2. **Recherche par arbres des suffixes :** Soit S^c la séquence inverse et complémentaire de $S[1, n]$, autrement dit la séquence obtenue en lisant S en sens inverse (de n à 1), et en remplaçant C par G, G par C, A par T et T par A. Par exemple, dans le cas de la séquence $S = CCCTCAGG$, on a $S^c = CCTGAGGG$.

   Le problème précédent (celui de la recherche de la plus longue tige-boucle) peut se ramener à celui de la recherche du plus long facteur commun, disposé d'une certaine façon, dans la séquence S et la séquence inverse et complémentaire S^c. Pour celà, on peut utiliser l'arbre des suffixes généralisé pour S et S^c.

   (a) Expliquer comment utiliser l'arbre des suffixes généralisé pour retrouver la plus longue tige-boucle (définition simplifiée comme à la question précédente) d'une séquence S. En particulier, **expliquer quel étiquettage est nécessaire pour les nœuds internes, et comment retrouver le résultat** à partir de cet étiquettage.

(b) Donner la complexité en temps et en espace de cet algorithme.

(c) Illustrer votre algorithme sur la séquence $S = CCCTCAGG$. Autrement dit, construire l'arbre des suffixes généralisé pour S et S^c, avec étiquettage, et indiquer où se trouve le résultat.

Exercice 4 : Phylogénie, méthodes de distance (18 points)

Soient quatre génomes identifiés par quatre ordres signés des gènes {a,b,c,d} :

$$G_1 : a\ b\ c\ d \qquad G_2 : a\ -c\ -b\ d$$
$$G_3 : a\ c\ -b\ d \qquad G_4 : a\ c\ -b\ -d$$

Les génomes sont considérés circulaires ce qui veut dire que les deux extrémités sont considérées adjacentes. Autrement dit, (d, a) est une adjacence dans G_1, G_2, G_3 et $(-d, a)$ est une adjacence dans G_4.

1. Remplir la matrice BP 4×4, où $BP(i, j)$ représente la distance de points de cassure (breakpoints) entre G_i et G_j.

BP	**G_1**	**G_2**	**G_3**	**G_4**
G_1				
G_2				
G_3				
G_4				

2. Est-ce que la distance BP calculée est **ultramétrique** ? Est-ce qu'elle est **additive** ? Justifier vos réponses.

3. Dérouler l'algorithme UPGMA sur les séquences ci-dessus i.e. expliciter (1) les sous-arbres créés à chaque étape avec la longueur des branches ; (2) les matrices de distances successives obtenus ; (3) l'arbre final obtenu avec ses longueur de branches.

Exercice 5 : Phylogénie, Méthode de parcimonie (20 points)

1. Expliquer, dans ses grandes lignes, le principe de la méthode de parcimonie pour la construction d'un arbre phylogénétique. En particulier ; quels sont les arbres à tester, qu'est-ce qui est optimisé ; comment calculer le score d'un arbre.

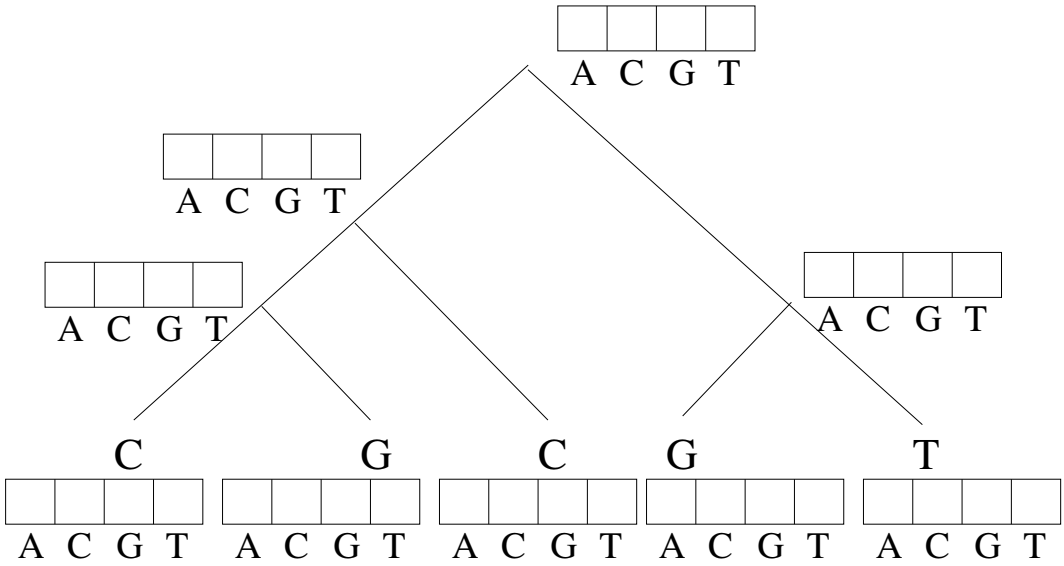
2. Soit $\mathcal{T}$ un arbre enraciné binaire de n feuilles, chaque feuille étant étiquettée par un nucléotide. On note par $S(a, b)$ le score de substitution d'un nucléotide a par un nucléotide b.

   (a) On demande dans cette question de rappeler l'algorithme de Sankoff pour trouver les nucléotides aux nœuds internes de $\mathcal{T}$ permettant de minimiser le score de l'arbre. Plus précisément, étant donné un nœud k de $\mathcal{T}$, on note $S_k(a)$ le score du sous-arbre de $\mathcal{T}$ de racine k sachant que k est étiquetté par le nucléotide a.

   Expliciter : (1) **les conditions initiales** et (2) **la formule de récurrence permettant de calculer $S_k(a)$ pour tout k et tout a**, et (3) expliquer **où trouver le score minimal de l'arbre** et (4) comment obtenir une **assignation optimale des nœuds internes**.

(b) Dans cette question, on demande d'appliquer l'algorithme de Sankoff sur l'arbre suivant, en considérant que toute substitution a un score de 1, autrement dit : $S(a, b) = 1$ pour tout $a \neq b$ et $S(a, a) = 0$.

Pour celà : (1) **Remplir les vecteurs** associés à chaque nœud de l'arbre, y compris les feuilles, et (2) **Déterminer le score minimal de l'arbre**.



(c) Expliciter une assignation optimale des nœuds internes **qui ne serait pas inférée par l'algorithme de Fitch**. Expliquer pourquoi.