Agrégation externe de mathématiques, session 2021
Épreuve de modélisation, option informatique

(*D01*)

Mots clefs : programmation dynamique, arbre, grammaires

► *Il est rappelé que le jury n'exige pas une compréhension exhaustive du texte. Vous êtes laissé(e) libre d'organiser votre discussion comme vous l'entendez. Des suggestions de développement, largement indépendantes les unes des autres, vous sont proposées en fin de texte. Vous n'êtes pas tenu(e) de les suivre. Il vous est conseillé de mettre en lumière vos connaissances à partir du fil conducteur constitué par le texte. Le jury attend une illustration sur ordinateur comportant une partie significative de programmation ; la première des suggestions du jury en fin de texte permet aux candidats de se faire une idée des attentes minimales du jury en la matière.*

1. Motivation

L'information génétique est contenue dans les génomes que l'on peut voir formellement comme des textes linéaires sur l'alphabet des quatre nucléotides $\{A, C, G, T\}$. Le produit "final" que sont les protéines se replie dans l'espace en une structure dite tertiaire. Nous proposons d'étudier des propriétés formelles d'une structure intermédiaire dite *secondaire* lorsqu'elle est planaire.

2. Notations et représentation

Définition 1. *Soit $\mathcal{V}$ un alphabet et ϕ une bijection de $\mathcal{V}$ dans $\mathcal{V}$ satisfaisant $\phi^2 = Id$. Etant donné un mot w de longueur m sur l'alphabet $\mathcal{V}$, le mot $\tilde{w}$ de taille m satisfaisant*

$$\tilde{w}_i = \phi(w_{m-i+1}), 1 \leq i \leq m \tag{1}$$

est appelé image palindromique *de w. Un* palindrome *est un couple $(w, \tilde{w})$. Un mot est dit* palindromique *si $w = \tilde{w}$.*

Exemple 1. *Soit ϕ l'application définie sur l'alphabet $\mathcal{V} = \{A, C, G, T\}$ par*

$$\phi(A) = T, \phi(T) = A, \phi(C) = G, \phi(G) = C \ .$$

L'image palindromique de $ACGGT$ est $ACCGT$.

Dans les notations ci-dessous, $T[i \cdot\cdot j]$, $i < j$, représente la sous-séquence d'un texte T comprise entre les positions i et j incluses. $T[i]$ représente le caractère en position i.

Définition 2. *Dans un texte T, deux palindromes $(w_1, \tilde{w}_1) = (t[i_1 \cdots j_1], t[\tilde{i}_1 \cdots \tilde{j}_1])$ et $(w_2, \tilde{w}_2) = (t[i_2 \cdots j_2], t[\tilde{i}_2 \cdots \tilde{j}_2])$ sont dits* non-chevauchants *si ils satisfont*

$$([i_1, j_1] \cup [\tilde{i}_1, \tilde{j}_1]) \cap ([i_2, j_2] \cup [\tilde{i}_2, \tilde{j}_2]) = \emptyset \ .$$

Deux palindromes non-chevauchants sont dits
— disjoints *si ils apparaissent dans l'ordre $w_1, \tilde{w}_1, w_2, \tilde{w}_2$ ou $w_2, \tilde{w}_2, w_1, \tilde{w}_1$*
— imbriqués *si ils apparaissent dans l'ordre $w_1, w_2, \tilde{w}_2, \tilde{w}_1$, ou $w_2, w_1, \tilde{w}_1, \tilde{w}_2$ On dit alors que $(w_2, \tilde{w}_2)$ est imbriqué dans $(w_1, \tilde{w}_1)$ ou que $(w_1, \tilde{w}_1)$ est imbriqué dans $(w_2, \tilde{w}_2)$.*
— intercalés *si ils apparaissent dans l'ordre $w_1, w_2, \tilde{w}_1, \tilde{w}_2$, ou $w_2, w_1, \tilde{w}_2, \tilde{w}_1$.*
Deux palindromes sont compatibles *si ils sont soit disjoints soit imbriqués.*

Définition 3. *Etant donnée une séquence de longueur n, une* structure secondaire *sur cette séquence est un ensemble de palindromes, aussi appelés* hélices*, de cette séquence compatibles deux à deux.*

Soit $(w_1, \tilde{w}_1)$ un palindrome. Si aucun palindrome $(w_2, \tilde{w}_2)$ n'est imbriqué dans $(w_1, \tilde{w}_1)$, la sous-séquence séparant l'extremité finale de w_1 de l'extrémité initiale de $\tilde{w}_1$ est appelée une boucle.

Une épingle à cheveux *est une structure secondaire admettant une seule boucle.*

Exemple 2. *Soit T la séquence* **ACC**$TCCGGCGGC\overline{TT}G\overline{AA}T$**GGT**. *Les trois palindromes (ACC, GGT), (CCG, CGG), (TT, AA) forment une structure secondaire où (CCG, CGG), (TT, AA) sont disjoints et tous deux imbriqués dans (ACC, GGT).*

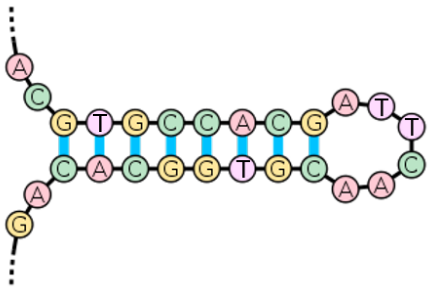

FIGURE 1. Une épingle à cheveux

Les contraintes de stabilité physique et thermodynamique imposent une taille minimale pour les boucles et les hélices. Ces tailles minimales sont notées respectivement b et h.

Une structure secondaire peut être interprétée comme une expression (bien) parenthésée et se modéliser par un arbre. Les énumérations se basent sur l'existence d'une bijection entre les structures secondaires et les chemins de Motzkin.

Définition 4. *Un* chemin de Motzkin *est un chemin dans $\mathbb{N} \times \mathbb{N}$ partant du point $(0,0)$ et terminant en un point $(n,0)$ en utilisant les déplacements :*
Montée : $(i, j) \to (i+1, j+1)$ noté M;
Descente : $(i, j) \to (i+1, j-1)$ noté D;

Plateau : $(i,j) \to (i+1,j)$ *noté* P;

Proposition 1. *Sans contrainte sur b, une structure secondaire se représente par un mot de Motzkin.*

Exemple 3. *Le mot de Motzkin associé à l'épingle à cheveux de la figure 1 est* $PP(M)^8P^6(D)^8PP$.

3. Algorithmique des structures secondaires

Une séquence donnée sur l'alphabet $\mathcal{V} = \{A, C, G, T\}$ admet formellement plusieurs repliements. Cependant, dans la nature, un seul repliement (ou un nombre très limité) est observé. Les caractéristiques de ce repliement restent mal connues des biologistes. Différentes hypothèses ont été émises, qui conduisent à différents algorithmes.

3.1. *Maximisation du nombre de paires*

Un premier algorithme cherche à *maximiser le nombre d'appariements* pris en compte dans la structure. Le calcul se fait par programmation dynamique, en calculant la meilleure structure pour les petites sous-séquences et en étendant à de plus grandes structures. On suppose que $b = 1$ et que $h = 1$.

Le nombre maximal de paires de bases que l'on peut former sur $T[i \cdots j]$ est noté $\theta_{i,j}$. On pose :

$$\delta_{i,j} = \begin{cases} 1 & \text{si } \phi(T[i]) = T[j] \ , \\ 0 & \text{sinon} \ . \end{cases} \tag{2}$$

L'algorithme se base sur l'observation suivante.

Remarque 1. *La meilleure structure de la séquence* $T[i \cdots j]$ *se déduit des meilleures structures des sous-séquences incluses de* 4 *manières possibles :*

(i) la position i est ajoutée, non-appariée*, à la meilleure structure pour la sous-séquence* $T[i+1 \cdots j]$ *;*

(ii) la position j est ajoutée, non-appariée*, à la meilleure structure pour la sous-séquence* $T[i \cdots j-1]$ *;*

(iii) la paire (i,j) *est ajoutée à la meilleure structure pour la sous-séquence* $T[i+1 \cdots j-1]$ *;*

(iv) $T[i \cdots j]$ *est la* combinaison *de deux sous-structures optimales sur* $T[i \cdots k]$ *et* $T[k+1 \cdots j]$ *;*

Algorithme 1. *Initialisation :*

$$\theta_{i,i-1} = 0 \quad \textit{pour } 2 \le i \le n \ , \tag{3}$$

$$\theta_{i,i} = 0 \quad \textit{pour } 1 \le i \le n \ . \tag{4}$$

Récursion :

$$\theta_{i,j} = \max\left(\theta_{i+1,j}, \theta_{i,j-1}, \theta_{i+1,j-1} + \delta_{i,j}, \max_{i<k<j}[\theta_{i,k} + \theta_{k+1,j}]\right) \tag{5}$$

Cet algorithme remplit une matrice $n \times n$ de gauche à droite et de bas en haut. L'angle supérieur droit, $\theta_{1,n}$ contient le nombre d'appariements de la (d'une) structure la mieux appariée. Un exemple de matrice finale est donné dans la Table 1.

-	C	C	C	T	T	T	A	G	G
C	0	0	0	0	0	0	1		
C	0	0	0	0	0	0	1	2	
C	-	0	0	0	0	0	1	2	2
T	-	-	0	0	0	0	1	1	1
T	-	-	-	0	0	0	1	1	1
T	-	-	-	-	0	0	1	1	1
A	-	-	-	-	-	0	0	0	0
G	-	-	-	-	-	-	0	0	0
G	-	-	-	-	-	-	-	0	0

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

TABLE 1. Table intermédiaire avec deux sous-séquences optimales et table finale pour la séquence CCCTTTAGG

Cet algorithme se modifie facilement en associant un score $s(x,y)$ à l'appariement de deux résidus, pour tenir compte d'une affinité physico-chimique accroissant la stabilité thermodynamique. Par exemple, on peut prendre $s(G,C) = 3$, $s(A,T) = 2$ et $s(x,y) = 0$ pour les autres couples.

Le score maximisé par ces algorithmes représentant une heuristique, il n'est pas certain que cette structure calculée soit la structure réelle dans la "nature". Lorsque l'on dispose de plusieurs séquences similaires -organismes ou gènes voisins- on recherche une structure commune, a priori sous-optimale. L'idée de base consiste à trouver des palindromes communs à l'ensemble des séquences pour les rechercher ultérieurement dans les autres séquences.

3.2. *Grammaires*

On peut engendrer une structure secondaire en utilisant une *grammaire*. La grammaire ci-dessous a 14 règles de production et un seul état terminal.

$$
\begin{array}{lllll}
 & S & \rightarrow & AS|CS|GS|TS & i \text{ non apparié} \\
 & S & \rightarrow & SA|SC|SG|ST & j \text{ non apparié} \\
(6) & S & \rightarrow & AST|CSG|GSC|TSA & i,j \text{ appariés} \\
 & S & \rightarrow & SS & \text{bifurcation} \\
 & S & \rightarrow & \epsilon & \text{terminaison}
\end{array}
$$

On associe des probabilités à chaque règle. On utilise les notations $P(xS), P(Sx), P(xSy)$ et $P(SS)$ et on utilise comme scores les $logarithmes$ de ces probabilités. On donne une forte probabilité aux productions appariant les bases et on cherche la dérivation de probabilité maximale en modifiant l'algorithme ci-dessus.

Etant donnée une structure (ou sous-structure) commune à une famille, il est intéressant de rechercher ce repliement pour une autre séquence de la même famille. Par exemple, les trois séquences ci-dessous admettent une structure commune en épingle à cheveux.

$$\begin{aligned} seq_1 &= \mathbf{CAG}GAAA\mathbf{CTG} \\ seq_2 &= \mathbf{GCT}GCAA\mathbf{AGC} \\ seq_3 &= \mathbf{GCT}GCAA\mathbf{AGC} \end{aligned}$$

La grammaire modélisant les épingles à trois appariements et une boucle GCAA ou GAAA, qui ne contient pas de bifurcation, sera :

$$\begin{aligned} S &\rightarrow AW_1T|CW_1G|GW_1C|TW_1A\ , \\ W_1 &\rightarrow AW_2T|CW_2G|GW_2C|TW_2A\ , \\ W_2 &\rightarrow AW_3T|CW_3G|GW_3C|TW_3A\ , \\ W_3 &\rightarrow GAAA|GCAA \end{aligned}$$

4. Comptage des structures secondaires

L'ensemble des appariements associés à une structure secondaire (sans tenir compte des lettres) est appelé un *repliement*. On note S_n le nombre de structures secondaires différentes sur une séquence de taille n.

Exemple 4. *On énumère l'ensemble des structures de taille* 5 *pour* $b = 1$. *Soit*

- w_1 n'est pas apparié. *La structure* $w_1 \cdot S(w_2w_3w_4w_5)$ *se réécrit suivant une des 4 possibilités :*

  $$w_1 \cdot \mathbf{w_2} w_3 w_4 \mathbf{w_5},\ w_1 \cdot \mathbf{w_2} w_3 \mathbf{w_4} w_5\ ;\ w_1 w_2 \cdot \mathbf{w_3} w_4 \mathbf{w_5}\ ;\ w_1 w_2 w_3 w_4 w_5$$

- w_1 est apparié. *Il y a alors 3 possibilités :*
  - $\mathbf{w_1} w_2 \mathbf{w_3} \cdot w_4 w_5$;
  - $\mathbf{w_1} w_2 w_3 \mathbf{w_4} \cdot w_5$;
  - $\mathbf{w_1} S(w_2w_3w_4) \mathbf{w_5}$ *qui se réécrit :*
    - $\mathbf{w_1 w_2} w_3 \mathbf{w_4 w_5}$;
    - $\mathbf{w_1} w_2 w_3 w_4 \mathbf{w_5}$;

On a $S_5 = 8$ *et on observe que*

$$S_5 = S_4 + (S_1S_2 + S_2S_1) + S_3\ . \tag{7}$$

Plus généralement, on obtient la relation de récurrence,

Théorème 1. *Lorsque la taille minimale des hélices est* $h = 1$ *et celle des boucles est* $b = 1$, *on obtient*

$$S_0 = S_1 = 1, S_n = S_{n-1} - S_{n-2} + \sum_{k=0}^{n-2} S_k S_{n-2-k}\ \ \forall n \geq 2. \tag{8}$$

Cette récurrence peut être utilisée pour étudier expérimentalement l'asymptotique de S_n et d'observer que S_n croît exponentiellement ; une étude analytique très délicate permet plus

précisément d'obtenir l'équivalent

$$S_n \sim \sqrt{\frac{15+7\sqrt{5}}{8\pi}} \cdot n^{-\frac{3}{2}} \cdot \left(\frac{3+\sqrt{5}}{2}\right)^n .$$

Suggestions pour le développement

► *Soulignons qu'il s'agit d'un menu à la carte et que vous pouvez choisir d'étudier certains points, pas tous, pas nécessairement dans l'ordre, et de façon plus ou moins fouillée. Vous pouvez aussi vous poser d'autres questions que celles indiquées plus bas. Vos investigations doivent comporter une partie traitée sur ordinateur, dont une partie significative de programmation.*

— On pourra programmer une fonction effectuant la tâche suivante : étant donné un texte $\mathscr{T}$ vu comme un tableau de caractères sur l'alphabet $\{A, C, G, T\}$, une position i du texte et un entier m, on note w le mot de taille m commençant en i ; la fonction renvoie alors toutes les occurrences de $\tilde{w} = \phi(w)$ disjointes de w, où ϕ est l'application de l'exemple 1.
— Montrer que le nombre de chemins de Motzkin de longueur n satisfait la même équation de récurrence que les nombres de Catalan, qui dénombrent les arbres binaires. Interpréter ce résultat à la lumière de la representation arborescente des structures secondaires.
— Modifier l'algorithme 1 de façon que la taille minimale d'une hélice soit h. Donner les nouvelles équations.
— Préciser la structure de données associée à l'algorithme 1. Donner sa complexité en temps et en espace.
— Etendre l'algorithme 1 à la recherche d'une structure de score maximal.
— Donner un algorithme de recherche de palindromes communs à un ensemble de séquences.
— Ecrire une grammaire pour modéliser les épingles à cheveux.
— Justifier l'équation (7) et prouver l'équation (8). On pourra considérer l'ensemble $\mathscr{F}$ des structures dans lesquelles le premier élement de la séquence n'est pas apparié.
— Généraliser (8) pour b quelconque.
— On pourra valider expérimentalement l'équivalent final.