

BIN702 - Série d'exercices sur les structures secondaires d'ARN

Manuel Lafond

Note : ici j'utilise $V[i, j]$ pour dénoter les tables de programmation dynamique (car j'utilise M pour dénoter une matrice de scores). La récurrence de Nussinov calcule $V[i, j]$ pour tout i, j , où $V[i, j]$ est le nombre maximum d'appariements pour $S[i..j]$. On a $V[i, j] = 0$ pour $i \geq j$ et sinon,

$$V[i, j] = \max \begin{cases} V[i + 1, j] \\ \max_{k=i+1..j} (V[i + 1, k - 1] + V[k + 1, j] + \delta_{ik}) \end{cases}$$

où $\delta_{ik} = 1$ si $S[i]$ s'apparie avec $S[k]$, et $\delta_{ik} = -\infty$ sinon.

Exercice 1: En classe, nous avons présenté une version récursive de l'algorithme de Nussinov. Écrivez le pseudo-code d'une version itérative de l'algorithme. C'est-à-dire, calculez les valeurs de la table de programmation dynamique avec des boucles, sans avoir recours à des appels récursifs.

Exercice 2: La récurrence de Nussinov calcule un nombre maximum d'appariements. Supposons plutôt que nous avons une matrice M qui donne un score d'appariement $M[a, b]$ pour chaque paire de nucléotides a, b . Donnez une récurrence qui permet de maximiser la somme des scores des appariements, toujours avec condition d'emboîtement des appariements.

Exercice 3: Considérez la séquence d'ARN ci-dessous avec ses paires de bases inférées. Dessinez l'arbre correspondant à ces appariements, puis dessinez un repliement en 2D qui illustre les sous-structures principales de la molécule.



Exercice 4: Supposons que l'on veut interdire les boucles multiples dans le contexte de maximisation du nombre d'appariements. On suppose que les seuls appariements possibles sont $A - U$ et $C - G$. Donnez un algorithme en temps $O(n^2)$ qui trouve ce maximum.

Indice : sans multi-boucle, un appariement ne peut pas avoir plus d'un appariement se trouvant directement à l'intérieur.

Exercice 5: La récurrence de l'algorithme de Nussinov pour $V[i, j]$ était basée sur deux cas possibles : soit i n'est pas apparié, ou soit i est apparié avec un certain $k \in \{i + 1, \dots, j\}$.

Il existe une autre version (équivalente) de la récurrence de Nussinov. Celle-ci est basée sur une autre séparation des cas possibles : soit i est apparié avec j , ou bien i n'est pas apparié avec j . Dans ce deuxième cas, il doit exister un $k \in \{i, \dots, j - 1\}$ tel que aucun appariement ne traverse k , c'est-à-dire que tous les appariements sont entre i et k , ou entre $k + 1$ et j .

Donnez une récurrence de programmation dynamique avec cette séparation en deux cas.

Exercice 6: Considérez l'algorithme qui minimise l'énergie libre des sous-structures d'un repliement 2D. On avait une récurrence pour $V[i, j]$ avec quatre cas possibles, un pour chaque sous-structure T, E, B et M . Les multi-boucles (M) étaient le cas le plus complexe à gérer, alors supposons qu'on s'en débarrasse et qu'on ne permette que les cas T, E et B . Décrivez la structure des arbres d'appariements qui sont possibles avec cette contrainte, puis

décrivez informellement la forme des molécules d'ARN qui peuvent résulter de l'algorithme avec cette contrainte.

Exercice 7: Supposons que l'on veut maximiser le nombre d'empilements, c'est-à-dire le nombre d'appariements (i, j) tels que $(i + 1, j - 1)$ sont aussi appariés. Donnez un algorithme qui maximise le nombre d'empilements.