

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.

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.