BIN702 - Série d'exercices sur le séquençage

Manuel Lafond

Exercice 1: Soit l'ensemble des 3-mers
$\{AGT, AAA, ACT, AAC, CTT, GTA, TTT, TAA\}$.

— Construisez le graphe de "overlap" et trouvez la plus courte superséquence à l'aide d'un chemin Hamiltonien (i.e. un chemin qui passe exactement une fois par sommet).

— Construisez le graphe de De Bruijn pour ces 3-mers. Est-ce qu'il existe un chemin Eulérien ? Si oui, comparez la superséquence résultante avec celle obtenue à la question précédente.

Exercice 2: Considérez l'algorithme glouton pour résoudre le problème du commis voyageur sur le graphe d'overlap.

— On trouve l'arête (u, v) de poids maximum et on initialise un chemin $P = (u, v)$;

— tant que la taille de P est plus petite que n :

  — soit w_1 le *premier* sommet de P. On trouve le voisin entrant z_1 de w_1 tel que z_1 n'est pas déjà dans P, et tel que (z_1, w_1) est de poids maximum.

  — soit w_p le *dernier* sommet de P. On trouve le voisin sortant z_p de w_p tel que z_p n'est pas déjà dans P, et tel que (w_p, z_p) est de poids maximum ;

  — si le poids de (z_1, w_1) est plus grand que celui de (w_p, z_p), on ajoute z_1 en début de P. Sinon, on ajoute z_p en fin de P.

Montrez que cet algorithme peut retourner une solution sous-optimale. Il suffit de donner un exemple d'instance.

Exercice 3: Donnez un algorithme qui reconstruit un chemin Eulérien dans un graphe orienté, s'il en existe un.

Exercice 4: Soit $S = ACATACACAG$. Construisez la transformée Burrows-Wheeler de S.

Exercice 5: Supposez que vous avez déjà l'arbre de suffixes d'une chaîne S. Montrez comment, de façon efficage, on peut construire la transformée de Burrows-Wheeler (i.e. la dernière colonne des rotations) à partir de l'arbre.

Suggestion. On peut visiter les suffixes en ordre lexicographique (i.e. alphabétique).

Exercice 6: On a mentionné que si tous les k-mers d'une séquence ne sont pas représentés dans l'entrée, alors la technique du chemin Eulérien ne fonctionne pas. Dans ce cas, on peut essayer de trouver le minimum de chemins tels que chaque arête apparaît exactement une fois sur un des chemins.
Donnez un algorithme qui trouve ce minimum de chemins.
Version facile : faites deux suppositions : (1) supposez que s'il y a un cycle C, alors il existe une solution optimale (i.e. un ensemble de chemins) qui contient C (i.e. il y a un chemin qui suit toutes les arêtes de C de façon consécutive) ; (2) supposez qu'il existe un algorithme boîte-noire qui trouve, en temps polynomial, une solution optimale si vous avez un graphe orienté acyclique. Avec ces deux suppositions, vous devriez être en mesure de concevoir un algorithme en temps polynomial.
Version difficile : démontrez ou infirmez les deux suppositions ci-haut. Sont-elles vraies ou fausses ?