UNIVERSITÉ DE SHERBROOKE
DÉPARTEMENT D'INFORMATIQUE

IFT 870 BIN 710 - Forage de données

TP#4 : Fonctions descriptives

Hiver 2023

Le but de ce devoir est de pratiquer la comparaison et la sélection de méthodes de partitionnement sur des données séquentielles.

Ce devoir est à faire individuellement ou en équipe de deux. Il devra être complété avant le vendredi 14 avril 2023 à 23h59. Vous devez remettre, sur `turnin.dinf.usherbrooke.ca`, un fichier Ipython notebook (nommé `tp4.ipynb`) contenant votre rapport et vos scripts Python pour ce devoir.

Description des tâches à réaliser : Clustering d'un ensemble de séquences d'acides ribonucléiques (ARN)

Contexte : Les acides ribonucléiques (ARN) sont des molécules produites à partir des gènes et qui jouent d'importants rôles au sein des êtres vivants. Un ARN peut être représenté comme une séquence sur un alphabet de 4 lettres $\{A, C, G, U\}$. Les ARN non-codants (ARN-nc) sont des ARN qui ne servent pas à produire des protéines et qui sont directement fonctionnels en tant qu'ARN. En fonction de leurs rôles, les ARN-nc sont regroupés dans différents groupes (familles) fonctionnels. Les ARN-nc d'un même groupe peuvent avoir des séquences similaires, et partager des sous-chaînes (motifs) communes dont la présence est importante pour la réalisation de leurs fonctions. En pratique, un ensemble de séquences d'ARN-nc peut être obtenu suite à l'analyse de données de séquençage d'ARN, et pour regrouper ces ARN-nc en familles, on procède au partitionnement des données.

Tâches à réaliser : On vous fournit un ensemble de données de séquences d'ARN-nc stockées dans un fichier au format `.csv` (`TP4_data.csv`). Chaque donnée est représentée sur 2 attributs de type Texte : `id` est l'identifiant de la séquence et `sequence` est la séquence de l'ARN. Vous devez trouver une bonne représentation pour les données et un bon modèle de partitionnement pour calculer les familles d'ARN contenues dans cet ensemble de données.

1. Segmentation des données suivant la longueur des séquences
   (a) Visualisez la distribution des longueurs des séquences dans l'ensemble des données à l'aide d'un histogramme.
   (b) Proposez et justifiez un premier partitionnement des données en groupes de séquences de longueurs similaires. Par la suite, un modèle de partitionnement devra être fourni pour chacun de ces groupes.

2. Partitionnement à partir d'une représentation vectorielle des données
   (a) Pour chaque groupe de la question 1, générez une représentation vectorielle des séquences basée sur les 2-mer, 3-mer et 4-mer.
   (b) *Un motif (sous-chaîne) est fréquent si son score (i.e. la proportion de séquences qui le contient) est supérieur ou égal à* 1/3.
   Pour chaque groupe de la question 1, supprimez de la matrice de données tous les attributs correspondant à des motifs non-fréquents. Expliquez pourquoi il est important de retirer les motifs les moins fréquents de la représentation.
   (c) Pour chaque groupe de la question 1, trouvez la meilleure valeur de k (nombres de clusters) en utilisant la méthode de partitionnement en k-moyennes (k-means), la représentation obtenue à la question précédente, et le coefficient de Silhouette.
3. Partitionnement à partir d'une matrice de distances
   (a) *Pour cette question, vous devez installer la bibliothèque Python* `biopython` *pour avoir accès à la fonction bio.pairwise2.align.globalxx(). Étant donnée deux séquences S1 et S2, la distance d'édition simple entre S1 et S2 peut être calculée comme suit :*
   alignment = pairwise2.align.globalxx(S1,S2)
   distance = alignment[0][4]-alignment[0][2]
   Pour chaque groupe de la question 1, générez une matrice carrée des distances d'édition simple entre les séquences.
   (b) *La méthode de partitionnement en k-médoïdes est similaire à celle des k-moyennes, mais sa principale différence est qu'à chaque itération, le centre d'un cluster n'est pas le centroïde (moyenne des données du cluster), mais plutôt une donnée du cluster dont la distance moyenne aux autres données du cluster est minimum (médoïde).*
   Fournissez une implémentation de la fonction **KMedoid(n_clusters,X)** qui prend en paramètre un nombre de clusters `n_clusters` et une matrice carrée $n \times n$ de distances `X`. La fonction retourne un vecteur `y` de dimension n contenant les indices de cluster pour chaque séquence (indices = entiers dans l'intervalle [0,n_clusters-1]).
   (c) Pour chaque groupe de la question 1, trouvez la meilleure valeur de k (nombres de clusters) en utilisant la méthode de partitionnement en k-médoïdes, la matrice des distances entre les séquences, et le coefficient de Silhouette.
4. Analyse de la signification sémantique
   (a) En vous basant sur vos résultats pour la question 2 (Partitionnement à partir d'une représentation vectorielle des données), analysez les clusters obtenus pour proposer des caractéristiques communes et spécifiques aux données de chaque cluster.
   (b) En vous basant sur vos résultats pour la question 3 (Partitionnement à partir d'une matrice de distances), analysez les clusters obtenus pour proposer une séquence représentative des données de chaque cluster.
   (c) Laquelle des deux approches (question 2 ou question 3) permet une analyse plus appronfondie de la signification sémantique des clusters ? Justifiez votre réponse.

Remise du travail

Pour soumettre votre travail, connectez-vous, dans un fureteur, au serveur `http://turnin.dinf.usherbrooke.ca` en utilisant votre CIP, puis choisissez le cours `IFT870 (BIN710)` et le projet `TP4`. Chargez votre fichier `tp4.ipynb` et soumettez-le. Les noms de votre fichier de remise doit être exactement `tp4.ipynb`. Si le travail est réalisé en équipe, indiquez bien les noms des deux membres de l'équipe dans le fichier `tp4.ipynb`. Ne faites qu'une seule soumission par équipe. Ne remettez pas d'autre fichier.