CSI5180. Topics in Artificial Intelligence Machine Learning for Bioinformatics Applications Fall2019

Assignment 1 Report

1. Encoding

Step 1: Downloading the files

Read through the human_skin_microbiome.csv and downloaded the .fna.gz files using *urllib.request* in python into a folder named dataset. This folder is created in the location where the program is run. Now we have unzipped the .fna.gz files into a .txt file using *zgip in python*.

Step 2: Dealing with the lines preceded by '>'(not considering these lines) and converting all letters in the sequence to uppercase letters and also collecting the names of genome files into an array named labels.

Step 3: Creating a function to find the pairs in "ACGT" for the given *l* value

Step 4: Creating a function to replace the alphabets other than A, C, G ,T in a string with ""

Step 5: Finding the frequency vector

Traversing the text files of each organism in folder dataset and reading the whole file as single string (considering the sequences between each ">" as a complete unit in each file)and calling function in step 4. Now calculating the total possible pairs of length *l* in the sequence, which is our denominator. Now calculating the count of pairs derived calling function in step 3, in the sequence, dividing this with the denominator and storing all these in a vector named *encoded_vector* (frequency vector of each genome). Now appending frequency vector of each genome to *frequency_vector_allGenomes* (frequency vector of all genomes). Converting this into a *pandas dataframe* in python. This is our final frequency vector.

2. Analysis

2.1. KMeans

1. **KMeans**

2. Using matplotlib, showing the inertia of the clusters for all possible values of k

Applying KMeans with clusters ranging from 1 to 27 and plotting the inertia of clusters

Example: for *l*=6 – Fig 1

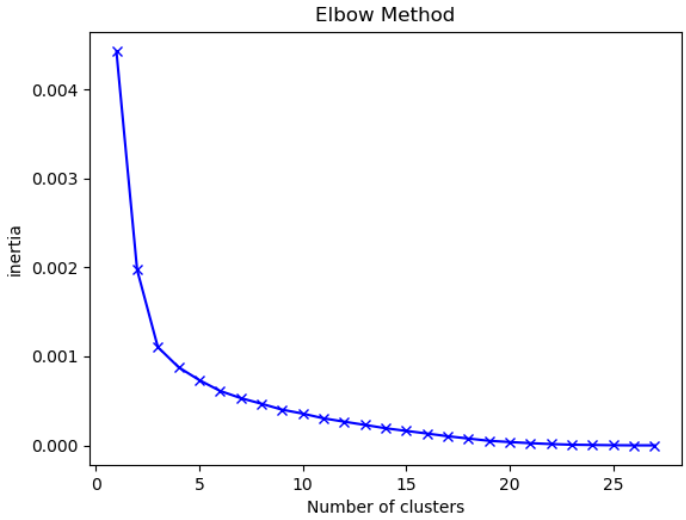

Fig 1:Inertia for l = 6

3. Using matplotlib, showing the silhouette score of the clusters for all possible values of k

Applying KMeans with clusters ranging from 2 to 26 and plotting the silhouette score of clusters (For silhouette score the range of length of clusters should from 2 to n-1)

Example: for *l*=6 – Fig 2

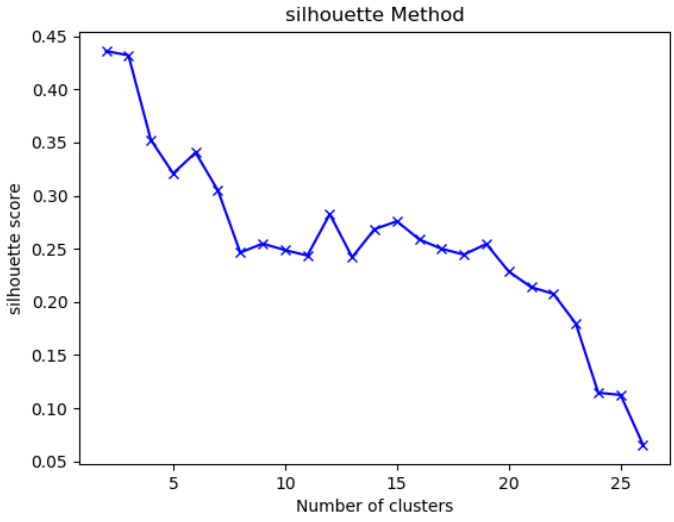

Fig 2: silhouette score for l = 6

a. optimal number of clusters

The number of clusters value for the highest silhouette score value is the optimal number of clusters. In the code, a python *dictionary* is created storing number of clusters as key and silhouette score as value for the key. We find the maximum of value in the dictionary and the corresponding key for it, which gives us the optimal k value.

Example: for *l*=6 – The optimal k value is 2-Fig 3

```
D:\study\uottawa\fall 2019\ML in bioinformatics\assignment1\submission>python a1.py 6 human_skin_microbiome.csv
optimal number of clusters : 2
```

Fig 3: Optimal value of k for l=6

2.2. Dendrogram

Using dendrogram and linkage from *scipy.cluster.hierarchy* we plot the result of a hierarchical cluster analysis. Input in the linkage function is frequency vector and method single, input into dendrogram function is the linkage and labels (genome file names).

Example: for *l*=6 - Fig 4

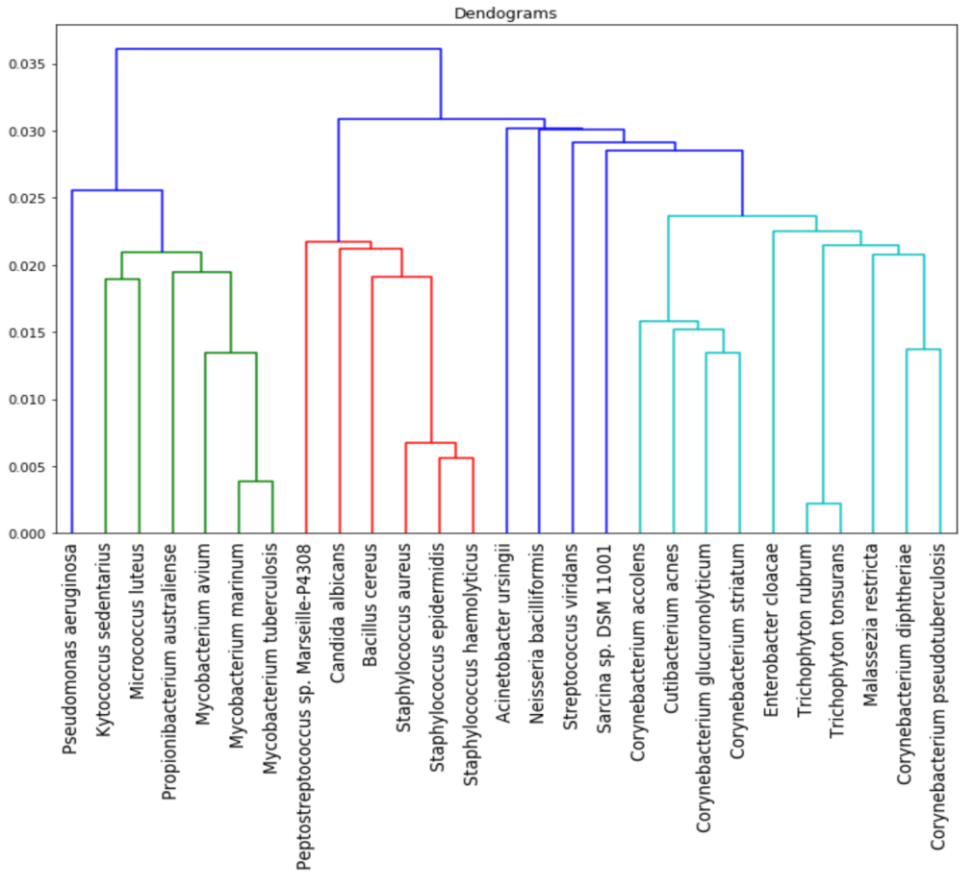

Fig 4: dendrogram for l = 6