CSI 5180. Topics in Artificial Intelligence

Machine Learning for Bioinformatics Applications

Assignment 1

Submitted By:

Nikhil Oswal (300074118)
noswa023@uottawa.ca

PROBLEM STATEMENT

Prepare and encode biological sequence data as frequency vectors; apply an unsupervised learning algorithm (KMeans) and determine the optimal number of clusters.

SOLUTION

The report is primarily divided in two parts. First, we talk about the key steps for encoding the biological sequence data which is later followed by a brief discussion on the results obtained by application of KMeans algorithm in the second part.

Part 1: Encoding

- **Data Download and Preparation**

  We first read the "**human_skin_microbiome.csv**" file and download FASTA files for each organism. The python code download and stores all the files inside "GenomeFiles" directory. The code checks the existence of this files before downloading them; this way the files are downloaded only for the first run and reuse them for all the subsequent runs. We use urllib for downloading the files.

  Once downloaded, the files are opened using gzip. We remove all the sequence IDs i.e. the lines starting with ">" as we are only interested in the actual sequences and save the extracted files as .txt.

  By end of this step, we have a folder "GenomeFiles" which has all the original FASTA files along with their respective cleaned .txt files.

- **Data Preprocessing**

  Next, we iterate over all the cleaned files (.txt) and remove any unwanted characters (characters apart from A, C, G, T) if they exist. Once done we create pairs of A's, C's, G's, T's for each of the sequences based on the input tuple size. If a file has multiple sequences, we create pairs for each of the sequences and then append the generated pairs together. We add the generated pairs to a new dataframe. We repeat this process for all the organisms one by one.

  By end of this step, we have a dataframe (27 * 2) which comprise of all the generated pairs for each organism.

This is how the dataframe looks like –

Sentence	Pairs (*Total Number of Pairs*)
(aaaaaa) (aaccgg) (ggcctt) (aaccgg) ….	1322988
…	
…	
27 Rows; one for each organism!	

- **Generating Frequency Vectors**

  We use sklearn's Count Vectorizer to generate frequency vectors. We pass all the generated pairs as input to Count Vectorizer which then returns us the vectors with appropriate frequencies. We later divide all the values with their respective total number of pairs to normalize our frequency vectors.

  Sample frequency vector for tuple size as 9

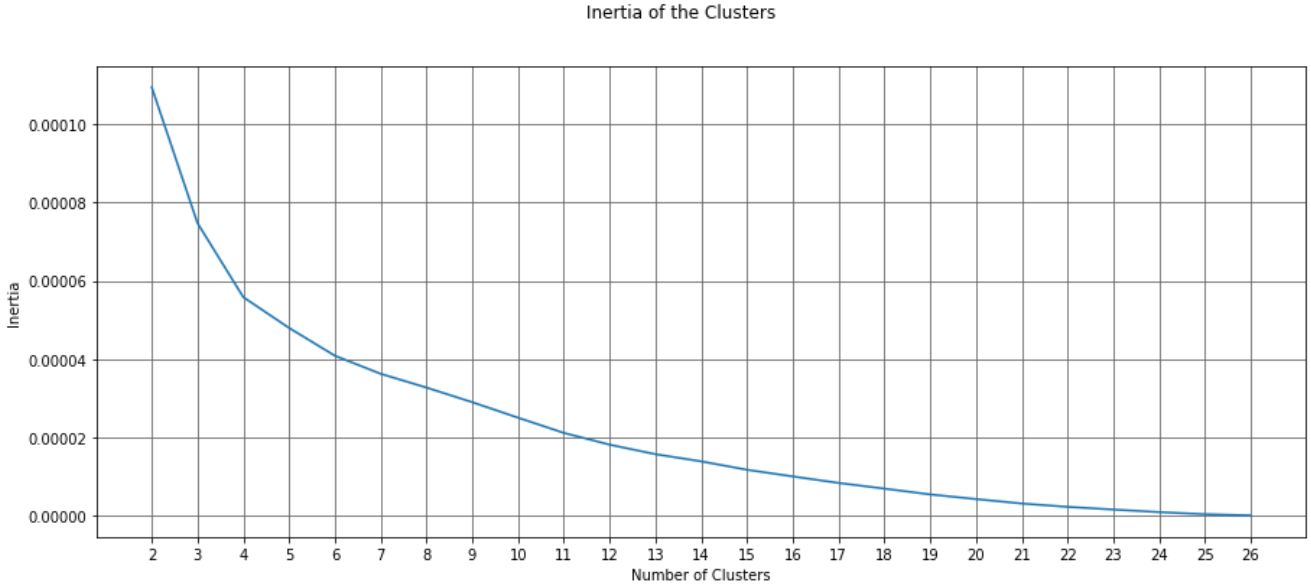
```
Generating frequency vectors
--------------------------------------------------------------------------------
Frequency vector : [[1.00389252e-05 1.95041976e-05 1.63491068e-05 ... 1.54886275e-05
  2.43802470e-05 7.74431374e-06]
 [1.87946823e-05 1.38196193e-05 1.52937121e-05 ... 1.45566657e-05
  1.56622352e-05 8.84455637e-06]
 [1.03597934e-03 1.15871641e-04 1.82704497e-04 ... 2.24597630e-04
  1.77100065e-04 1.01853554e-03]
 ...
 [4.00286805e-06 5.50394358e-06 7.50537760e-06 ... 5.00358507e-06
  9.00645312e-06 4.50322656e-06]
 [6.07655346e-04 2.09692162e-05 6.75172606e-05 ... 5.89307281e-05
  3.33970921e-05 6.61027424e-04]
 [6.67821961e-04 2.49727821e-05 8.34499730e-05 ... 6.58090575e-05
  3.78590932e-05 6.73731890e-04]]
Generating Frequency vectors completed, size: (27, 262144)
================================================================================
```

This is the part 1 of the assignment where we encode our dataset as frequency vectors. These frequency vectors are of (27 * (4^l)) dimension.

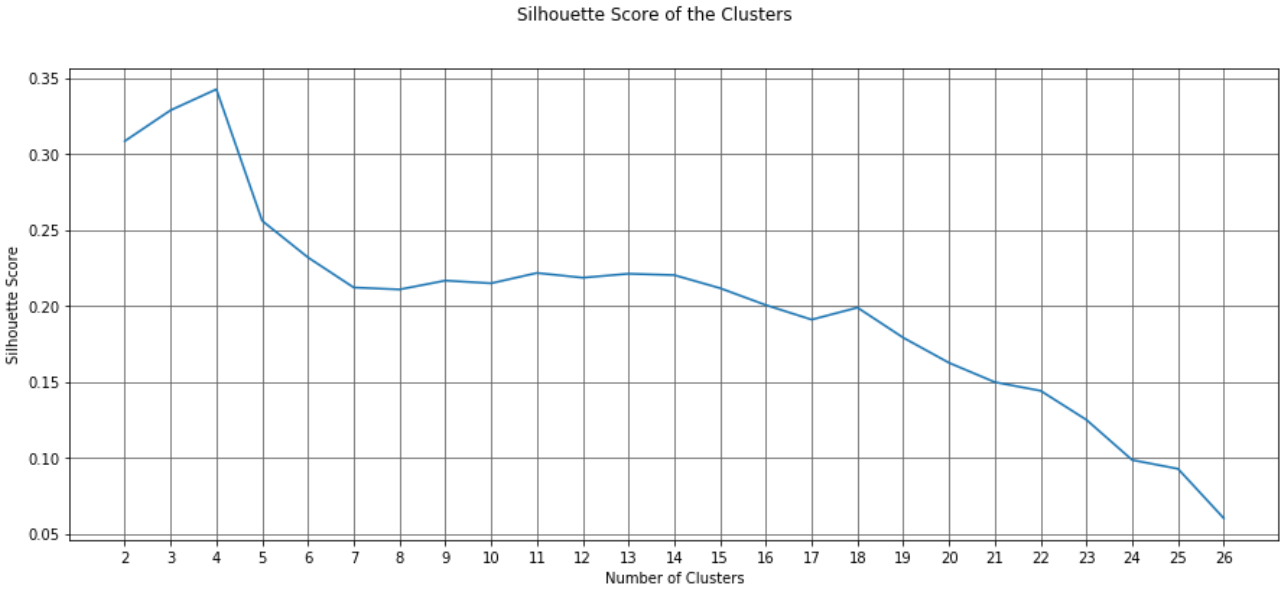
Overall, we have 27 organisms with a total of 1175 sequences.

Part 2: Analysis

- **KMeans**
  Here we apply KMeans algorithm on the encoded dataset with number of clusters ranging from 2 to 26. We calculated inertia and silhouette score for all the possible values of k.
  Below is the plot of Inertia for different values of k with **tuple size as 9**.



Below is the plot of Silhouette for different values of k with **tuple size as 9**.

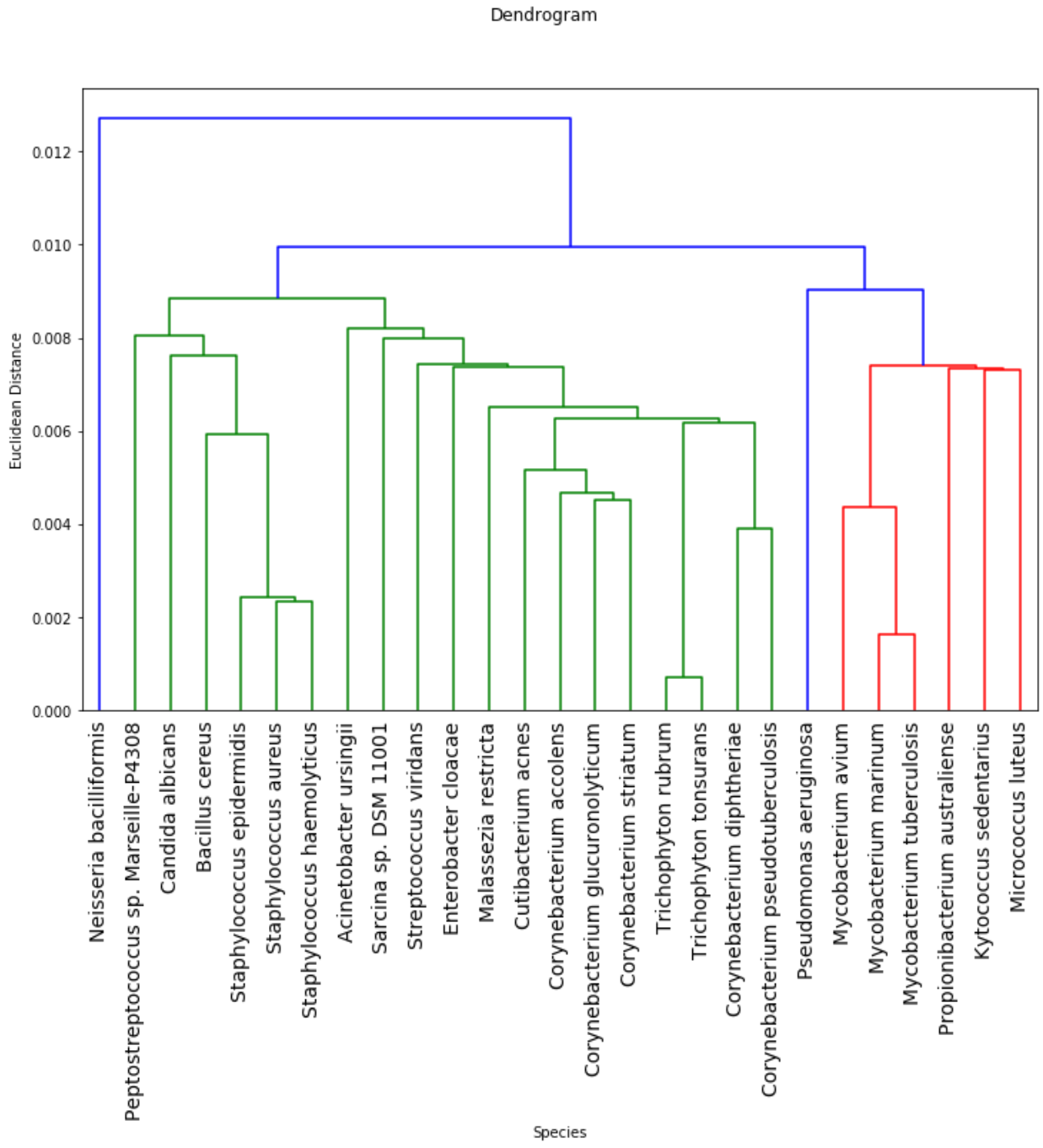


Based on the plot we can say that the **optimal number of clusters are 4**.

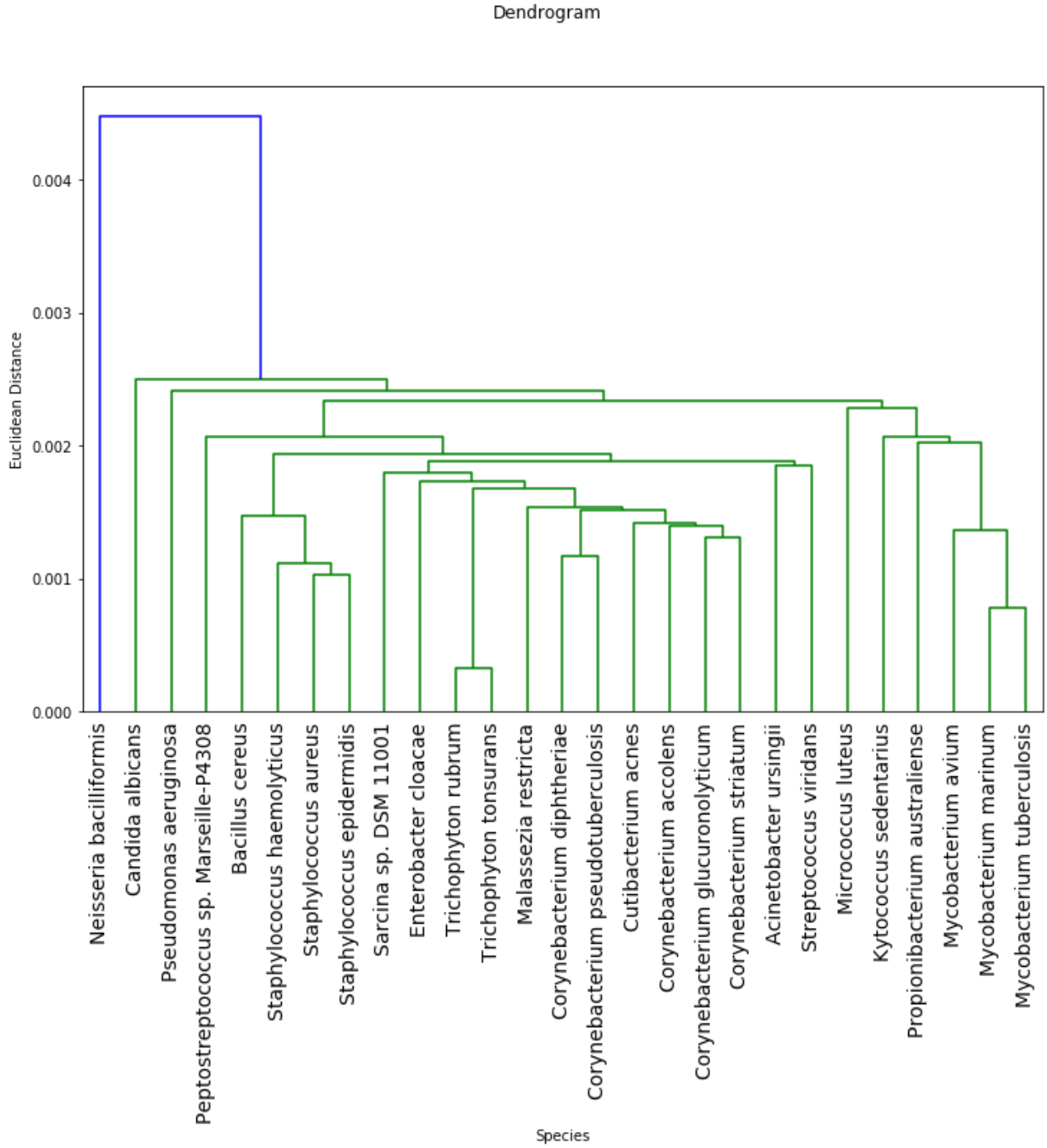
- **Dendrogram**

  Below is the Dendrogram plot for hierarchical cluster analysis based on single linkage –

  With tuple size as 6

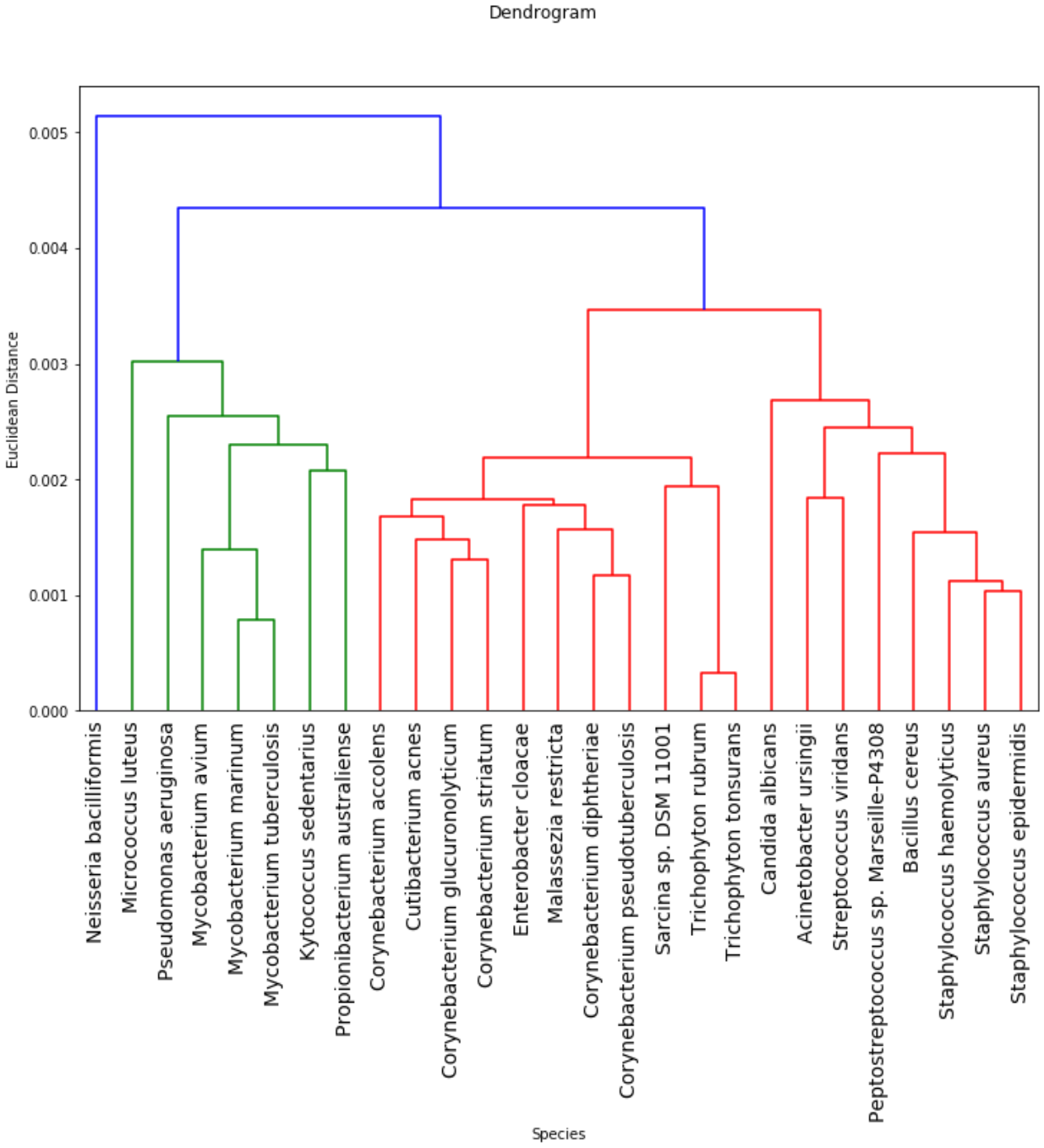


With tuple size as 9

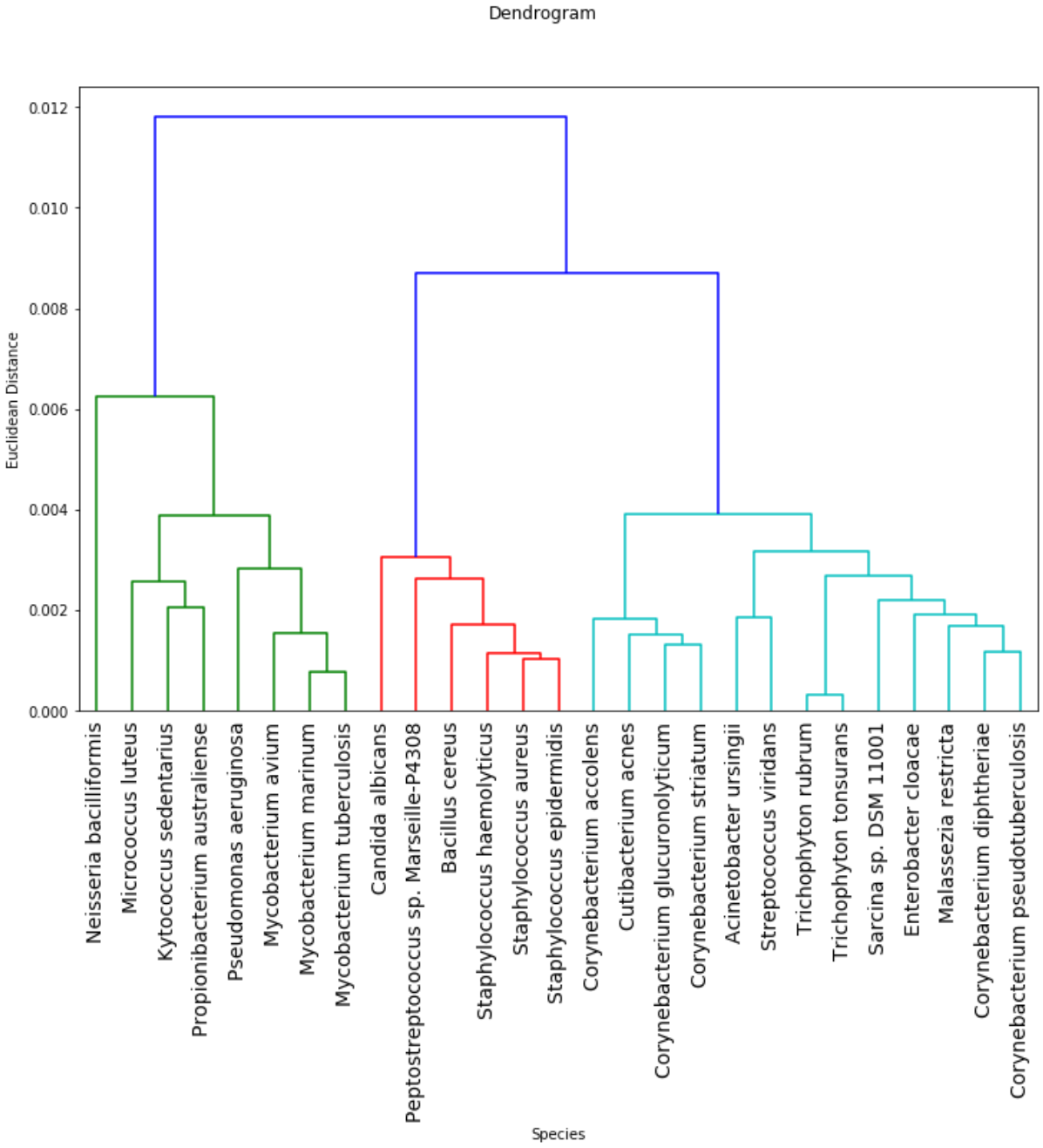


Dendrogram Plot with tuple size as 9 with ward and average linkage method:

Average Linkage Method



Ward Linkage Method



Tools and Technologies

Below are the tools and technologies used to complete this assignment –

Python (version 3.7.1), Google Colab's Jupyter Notebook, pandas, gzip, urllib, sklearn's CountVectorizer, Matplotlib, Sklearn's KMeans and scipy.cluster.hierarchy.

Link to my Jupyter Notebook:
https://colab.research.google.com/drive/1EIi0bdh3C3XhiHEc1YpsdkEZdpm9XjKk