Abhi Devathi
MATH 444
Assignment 5

Assignment 5

Problem 1

Summary:

This assignment takes in a genetic sequence from *Caulobacter crescentus*; creates a term-document matrix from this sequence, and then analyzes the matrix using different data visualization techniques such as PCA and Clustering.

Part A

This part was simple, running the genomic sequence in a *.fa* file to convert to a character array. The character array, shown in the workspace in MATLAB, had size 1 by 305400, implying the gene sequence is **305,400 nucleotides** long.

Part B

Part B is a quick testing of *CalcFreq.m*, which will be tested more heavily in the later sections. But, running *CalcFreq.m* should get a term document matrix whose number of words is the number of permutations of k-letter sequences of 4 choices of each letter. Running it from k = 1 to k = 6, we see that each document has number of rows 4, 16, 64, etc., which is equivalent to 4^k, the number of permutations we are looking for.

Part C

This section was a testing of the term-document matrix constructor function, *CalcFreq.m*, which takes in a sequence, k – length of the word, and L, the number of words. The first part was to vary k. Varying k from 1 to 6, the centered principal components of the data are in **Figure 1**.

Abhi Devathi
MATH 444
Assignment 5

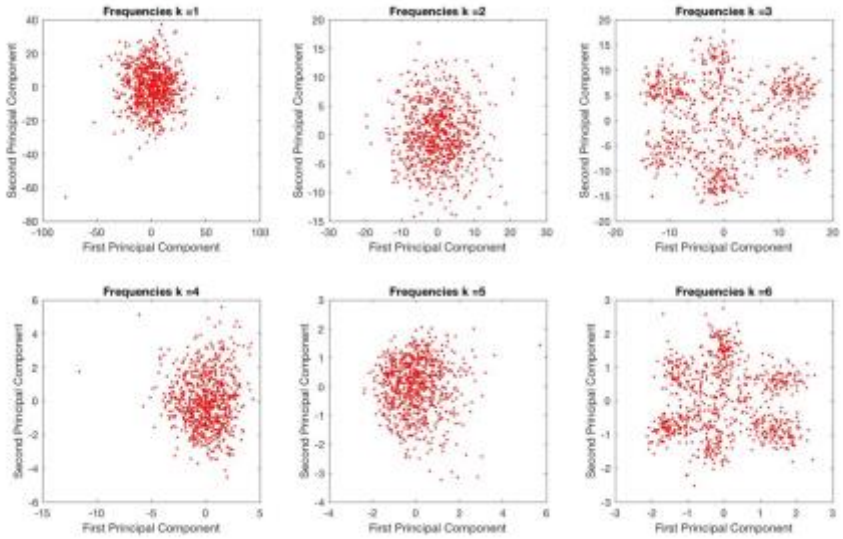

Figure 1: Each section shows the first principal components of the term document matrix, varying k (word length) from 1 to 6.

When k = 3 and k = 6, there is a clear sense of clustering of the data in the principal components. This is inherent in the data because the document is a gene sequence, and codons in these sequences are three characters long. Whenever the word length is a multiple of three, we can see some clear clustering of the data.

Varying L from L = 100 to L = 300, in multiples of 100 gave another interesting set of plots. As L grew, the data became sparser, leading to more strict clustering. The principal components are shown in **Figure 2**.

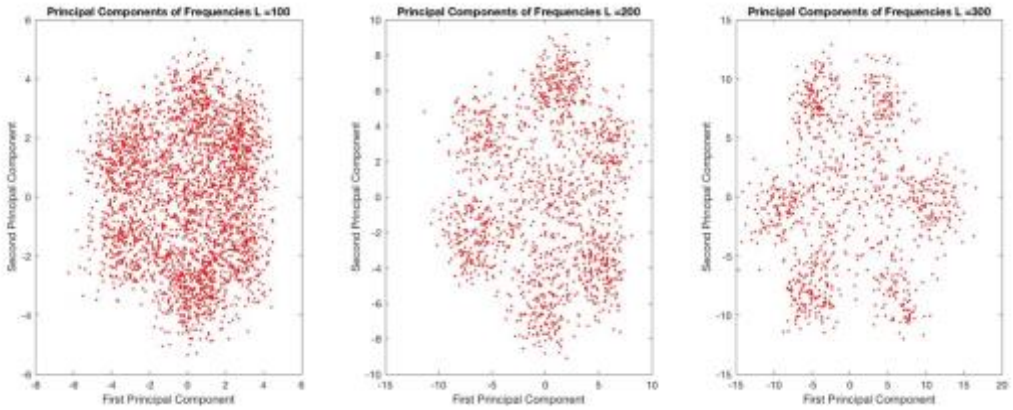

Figure 2: The figure shows the change in the principal components for a fixed k = 3 and varying L = 100,200,300 going from left to right.

Indicated by the axes, the data gets more and more sparse as L gets larger. This is also indicated by the number of columns decreasing as L grows larger. If we multiply the value of L by the number of columns, we get approximately the length of the gene sequence. For example, for L =

100, the term-document matrix has 3053 columns. Multiplying, we get 305,300 which is very close to the 305,400 length of the gene sequence. This works for the other L also.

The third part of this section is to test the code on a changed sequence, by deleting the first few characters of the gene sequence. In this example, I cut the first 100 nucleotides of the sequence, and then ran the CalcFreq algorithm on the changed sequence for k = 3 and L = 300, to see if there are any differences.

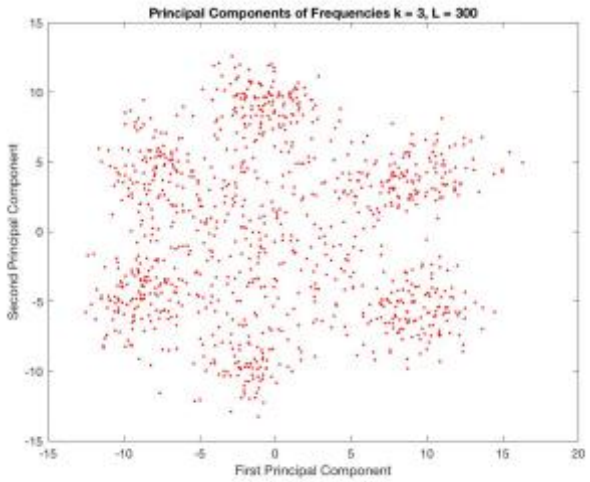

Figure 3: The principal components of a changed gene sequence, stripping the first 100 nucleotides for k = 3 and L = 300.

This figure can be compared to the top right subplot in **Figure 1**, because it used k = 3 and L = 300, only difference being the sequence itself. The data is not all that sensitive to the sequence itself, the only thing that changes is the number of data points, and number of words in the term-document matrix. Otherwise, the clustering is a little bit less but does not change the overall data too much.

Part D

A clustering algorithm was used to cluster the data for k = 3 and L = 300. In this case, the k-means algorithm was used, and the best clustering was found for 6 clusters, which was apparent in the principal component analysis done in part C. The results of the clustering are plotted in **Figure 4**.

Abhi Devathi
MATH 444
Assignment 5

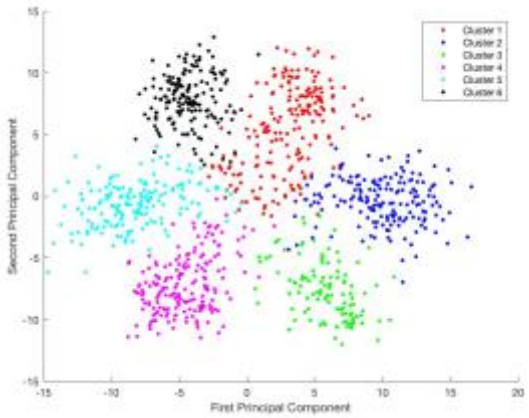

Figure 4: A k-means algorithm for 6 clusters was run on the term-document matrix for k = 3 and L = 300. The 6 clusters are shown with different colors. The figure shows the first principal components of the centered data.

Part E

This part asked to calculate the non-negative matrix factorization of the non-centered term document matrix, and pick the rank of the matrix as the number of clusters, 6.

This analysis of the data does not provide any valuable information to the project, and therefore was not included in this section. To be more thorough, the plotting of the feature vectors did not align with the clusters at all.

Part F

The cosines of the medoids and each document were calculated. The formula used was the usual dot product notation, normalizing the dot product by the multiple of the magnitudes of the vectors. This restricts the cosine value between -1 and 1. Then, the cosine of each data point was plotted for each medoid in **Figure 5**.

Abhi Devathi
MATH 444
Assignment 5

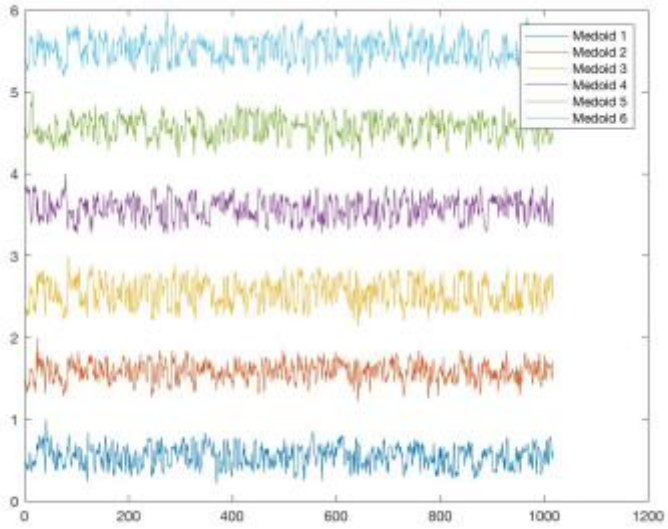

Figure 5: Each point on the x-axis is the cosine of the j-th data point and each medoid. Each different medoid was separated by a value to not overlap all the information, otherwise each graph is plotted between 0 and 1, because all the data is positive.

To identify which medoid represents the one for each cluster, then you would have to look at the medoid which has the highest cosine value. A higher cosine value corresponds to a higher correlation, or alignment, of the medoid and the data point.