Bioinformatics Lab Assignment 7: Sequence Variation and Clustering

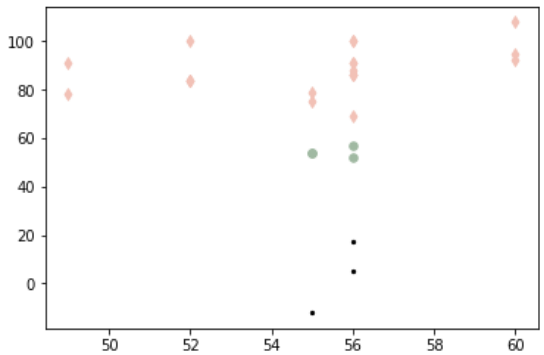
By Noah Segal-Gould

Background: For bioinformaticians seeking to compare mutated genomic data, it is useful to perform sequence alignment on two sequences and produce a similarity score between them. Utilizing the euclidean distance between these scores, clustering algorithms can be utilized to identify (predicted) distinct groups among multiple sequences. Using the Python programming language, these techniques are employed and tested against the known truth of which mutated sequences actually came from which specific original sequences.

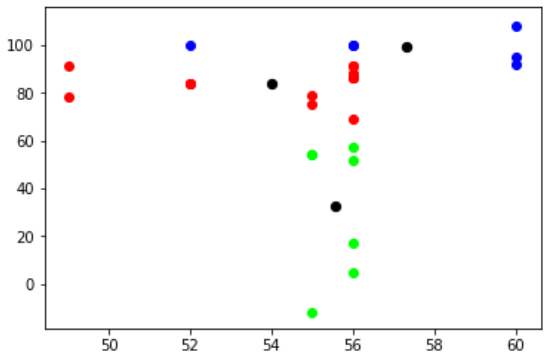
Methods: The most important functions in the program are `generate_additional_sequences` and `caluclate_similarity_matrices`. The former takes as its inputs the number of original sequences to generate and the desired length for each of those sequences (the default was 60 nucleotides). It outputs a dictionary of `"nucleotides"` and `"amino_acids"` each of which is itself a list of dictionaries of `"original_sequence"` and `"mutated_sequences"` which map respectively to a single string and a list of strings. The latter utilizes the former to calculate the similarity scores of each mutated sequence in comparison with its original. The data for the similarity scores of both nucleotide sequences and amino acid sequences are respectively placed into the first and second indices of tuples, thus creating a two dimensional array of 27 pairs of similarity scores which is then fed to both the `simpleCluster` algorithm with a threshold parameter of 10 and the `kMeans` clustering algorithm with a value of 3 for `k`. Both clustering algorithms were used to create scatter plots.

Results: When the program is run, it generates and mutates the sequences, aligns each mutation with its original for both nucleotides and amino acids, uses the produced similarity scores to determine clusters using both clustering algorithms with the euclidean distance between the scores, and plots the results of those clustering algorithms on the data.

Threshold Clustering:



K-Means Clustering:



To measure the accuracy of these methods, an appropriate measurement is the percentage of score pairs which were correctly clustered into their respective groups which should contain all sequences which were mutated from each original sequence. Thus, the accuracy of threshold clustering was 22% and the accuracy of k-means clustering was slightly better at 25%.

Conclusion: I completed tasks 1 through 4 in the lab assignment 7 description. I generated all the necessary sequences, calculated similarity scores, and performed clustering on them.

Acknowledgements: I worked alone on this assignment and used code from the our course's Moodle 2 page to complete it.