Christa Caggiano
BMI 203 HW 2

GITHUB REPO: https://github.com/christacaggiano/hw2-skeleton

I chose to use a Damerau-Levenshtein edit distance algorithm to calculate the distance between two amino acid sequences. I decided that sequence similarity was the best indication of similarity between proteins as this naturally encompasses some distinguishing properties of proteins- such as charge or hydrophobicity. Sequence similarity also is useful for measuring how evolutionarily close two proteins are. Since we often think of proteins that are closer together in evolutionary time as more similar, this would be a useful feature of my metric. For example, two proteins may have similar levels of hydrophobicity in their active sites, but have some from extremely divergent evolutionary histories, which may not be useful in assessing the functional similarities between proteins. I specifically chose the Damerau-Levenshtein algorithm because it takes in account insertions and deletions, which are expected in a biological context.

For a partitioning algorithm, I chose a k-means clustering algorithm that calculated centroids by taking the "average" of all the sequences in a given cluster. This means that I chose the sequence one with the smallest distance to all other strings, in hopes that this would be the most representative sequence in its cluster. Since I was working with strings and not integers, I felt this was the fairest way to calculate an average. Averaging in this way, however, could be a drawback when the sequences in a cluster are very distant. Additionally, this constrains the centroid to be included in my set of sequences.

My hierarchical clustering was a simple agglomerative clustering with a single linkage. Single linkage was chosen because it seemed to make the most sense for the strings I was working with. Average or centroid linkage was difficult to compute for a set of strings, and complete linkage was very biased toward outlying strings. I found that this was a straightforward algorithm, without needing a specified number of clusters or being dependent on initial starting conditions, as with k-means clustering. However, I felt that this algorithm was less biologically relevant than k-means as at the end, I had a difficult time interpreting what the hierarchy of sequences actually means.

The most intuitive way I thought of to calculate a quality metric was by generating sequences that were extremely similar – off by one or two amino acids- and performing multiple simulations to observe how often they clustered together. I found that this metric was very sensitive in k-means (Image 1) where the initial starting conditions and number of clusters changed the number of successes. In hierarchical clustering, because it was not dependent on random initial conditions the results were more consistent. Thus, this metric was less useful for hierarchical clustering. This metric could have been improved by having a more robust test dataset, many test datasets, by using cross-validation, or a combination of all three.

I found kmeans clustering to be fairly biologically relevant. It tended to group active sites with similar motifs together and grouped the sequences of similar lengths. Both of these things may affect protein functionality. Hierarchical clustering was more consistent than k-means, as pairs of simulated sequences that were expected to be close were always adjacent in the hierarchy (Image

2). However, I found this difficult to interpret with biological meaning. What does it mean for molecule D and molecule A to be nearby in a hierarchy? In this case, I think exploring other similarity metrics that more easily lend to numerical ranking, such as hydrophobicity or charge, would be better suited for hierarchical clustering.

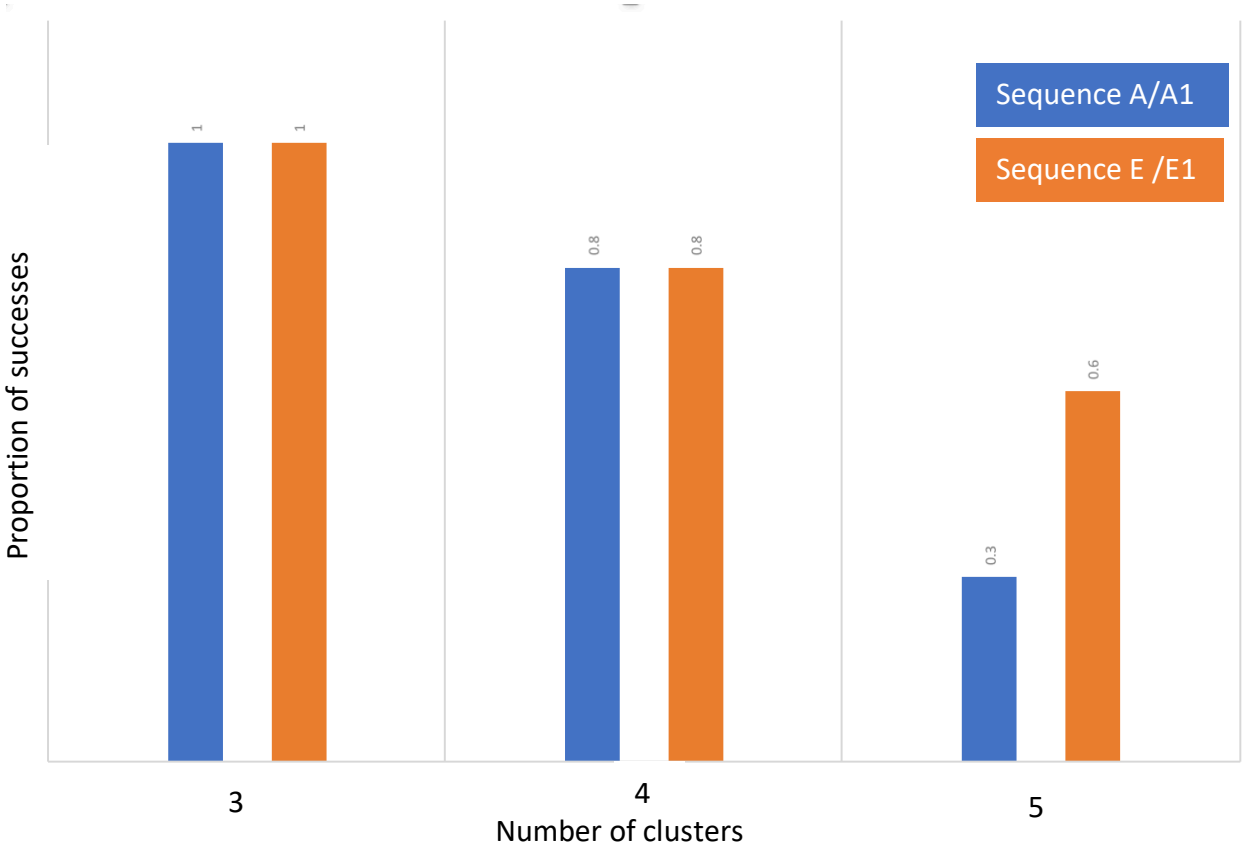

Image 1: Number of times (out of 10 simulations) that two sets of nearly identical sequences were clustered together in my portioning algorithm

Christa Caggiano
BMI 203 HW 2

Image 2: Hierarchy of simulated sequences using agglomerative clustering algorithm