1. 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.
2. 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, however, when the strings are very distant. Additionally, this constrains the centroid to be included in my set of sequences.