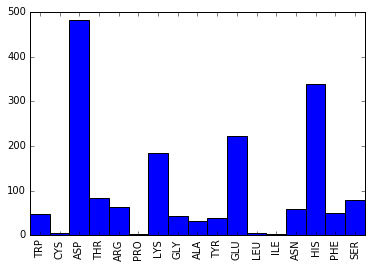
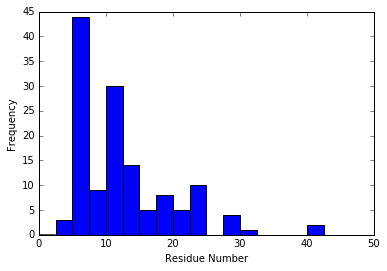
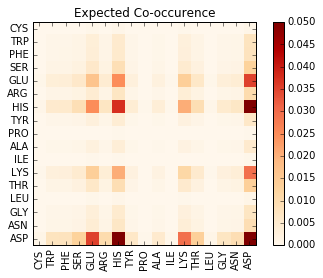
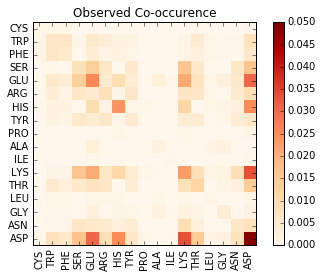
**Similarity Metric Description and Justification**

Before considering any similarity metrics, I summarized the data to look at global properties:



*Figure 1: Summary of active site dataset properties – the distribution of the total number of amino acid residues present in the active site (left) and the frequency of different residue types across the dataset (right).*

Based on observed frequencies of different residue types, it is possible to predict the probability of co-occurrence under the assumption of independence. If that prediction were similar to observed frequencies of co-occurrence, then amino acid composition would offer little information about the active site. However, there are dramatic differences between expected and observed co-occurrence of different amino acids.



*Figure 2: Observed and expected (under assumption of independence) co-occurrence of residue types in active sites in the dataset.*

To analyze the active sites, I chose to start with a simple similarity metric based on amino acid composition: the Jaccard similarity of the amino acid types present in the active site. Jaccard similarity is the ratio of residue types present in both active sites to the total number of residue types present in both active sites. For example, if active site A contains lysine, histidine, and aspartate and site B contains lysine, aspartate, and serine, their Jaccard similarity is 0.5.

**Partitioning Algorithm**

I implemented a k-medioids algorithm to cluster the active site data. It begins by randomly choosing k data points as centers, then assigns all other data points to the ‘nearest’ center (where nearest means the center with the largest similarity). Within each cluster, the data point that has the maximum average similarity to all other data points within the cluster becomes the new center. All other points are then assigned to their new nearest center. This process is repeated until the clusters converge to a single answer.

**Hierarchical Algorithm**

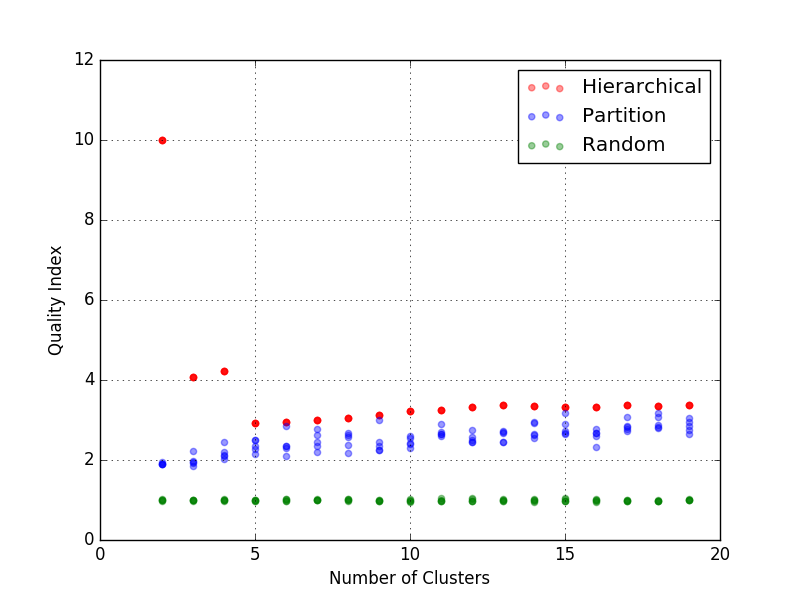
I implemented a hierarchical clustering algorithm that iteratively combines the two closest clusters into a single cluster (and maintains the structure of the relationships in a nested list). For clusters composed of multiple active sites, the distance is defined as the average distance of all objects in the cluster to each other point in the dataset.

**Clustering Quality Metric**

In order to quantify clustering quality, I calculated a weighted average of inter-cluster and extra-cluster distances for each node in the data set.

**Comparison of Clustering Algorithms**

In terms of runtime, the partitioning algorithm seems to run much faster than the hierarchical algorithm (at least as I’ve implemented it). However, at any given cluster size, hierarchical clustering appears to produce higher quality clustering as measured by the quality metric described above.



*Figure 3: Clustering quality as a function of the number of clusters in the partition clustering, the hierarchical clustering, and randomly generated clusters.*

**Biological Significance of Results**

Clustered active sites have similar amino acid compositions. To improve on these results, it might be interesting to incorporate a substitution matrix approach along with a method for finding optimal alignments. However, handling different sized active sites remains a challenge in this approach.