Problem Set #3 BCH364C/394P Systems Biology/Bioinformatics Marcotte Spring 2017
Due Monday, April 10, 2017

1. Download the yeast protein sequences from the course web site. These represent the complete sets of proteins (the "proteome") from *Saccharomyces cerevisiae*. The protein sequences were translated from predicted genes found when the genome was sequenced, and many were later verified by other means. Many of these genes were known prior to the genome sequence, but about ~1/3 of the genes were new. Each entry begins with a protein name (a common name or a unique id code from the Entrez genome database: `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Genome`). The name is followed immediately by the known location that the protein occupies inside a yeast cell. (The name and the location are fused together by an underscore character so that you can keep track of both of them together.)

2. Calculate the frequencies of each amino acid for each of the proteins. The file contains lines starting with ">" that separate the protein sequences—be sure to skip these lines for your calculation, but keep track of the protein name/location so that you can generate a feature vector of amino acid frequencies for each protein. Also, some entries contain non-amino acid characters (e.g., when protein sequences are ambiguous), so skip these characters as well, just keeping track of the 20 amino acids. Write the amino acid frequencies of each protein as a vector, separated by tabs. (In Python, you can print a tab character using "`\t`", just as you might print a newline character with "`\n`".)
So, you should have a vector for each protein sequence encoded by the genome, in the form:
ProteinName_location 0.069 0.011 0.048 0.069 0.054 0.058 0.021 0.072 0.088 0.112 0.023 0.059 0.033 0.037 0.035 0.068 0.044 0.057 0.007 0.037
where the first word is the name_location of the protein, followed by a tab, the fraction of Alanine (A) in the proteome, a tab, the fraction of Cysteine (C), etc. Be sure to write the amino acid frequencies in the same order for each protein!

3. We're going to explore these data using clustering. There are many tools for clustering, many of which you can download and install locally on your computer, but we'll take advantage of a nice web-based tool called Morpheus, available from the Broad at https://software.broadinstitute.org/morpheus/

4. The program should be fairly self-explanatory. Modify your amino acid frequency vectors from question 2 above to work with the Morpheus program by adding a line to the beginning of your data file that consists of the word PROTEIN, followed by a tab, then each amino acid in order separated by tabs, e.g.:
PROTEIN A C D E F G H I K L M N P Q R S T V W Y
Open your amino acid frequency vectors into the Morpheus program. Perform hierarchical clustering on your data by clicking the wrench button and selecting "hierarchical clustering", selecting "rows" (or "rows and columns", if you prefer), a similarity measure of your choice, and "average linkage clustering". After hierarchical clustering (which takes a little time), you should see a clustered set of feature vectors. You can interactively navigate the proteins to see how they cluster together

5. Do the clusters group proteins in a fashion consistent with their sub-cellular locations? Are there some locations clearly clustered better than others? Support your answer with 2 examples captured as screen shots of the clusters.

6. Download the file of yeast protein phylogenetic profiles from the course web site. Each entry in this file is the phylogenetic profile of a yeast protein, which captures the phylogenetic distribution of that gene family across species. Following the `name_location` are 149 numbers, each indicating the similarity of the protein to the best matching protein in one of 149 genomes. The numbers correspond to the genome names listed in the first line of the file.

7. In Morpheus, load in the phylogenetic profile data you downloaded in step 6. Cluster all of the genes with hierarchical clustering. Do genes with similar subcellular locations cluster? Several subcellular compartments show better clustering than others---print out 2 clusters where many of the proteins come from the same location.

8. Likewise, try clustering the proteins based on their chromatographic co-purification patterns (also downloadable from the course web site). For these data, yeast cells were lysed and native protein complexes were separated chromatographically, and the proteins in each chromatographic fraction identified by mass spectrometry. Print out 2 clusters where many of the proteins come from the same location.

9. Which types of features, co-purification, amino acid frequencies, or phylogenetic profiles, seem to be working the best to organize the proteins in a manner consistent with their subcellular locations? You may have noticed that many of these features have only limited power (at least for the relatively small datasets we're considering) to group co-localized proteins together. Why should you expect these features to inform us about subcellular locations of proteins?

10. Even though not all of the genes from the same location cluster together, the data can still be used to predict the subcellular location of proteins. For example, one such strategy for predicting the subcellular locations of yeast genes exploits the fact that very small clusters of co-inherited (or co-expressed or similar composition) proteins have similar subcellular locations. You can explore this a bit in Morpheus by selecting a set of genes using the search bar function on the top left of the screen. (If you press the small vertical arrow to the right of the search bar, you can redraw the clustergram to move your "matches to top" so you can see them. It is then possible to search for the "Nearest neighbors" (in the pulldown tools menu) of genes you select. For a dataset of your choice, select a set of proteins of your choice (specified e.g by name or location; examples include "actin", "golgi", etc.) & search for their nearest neighbors using Morpheus. You might test a few examples to find one that seems to have a bit of predictive power. Turn in a screen shot of your results.