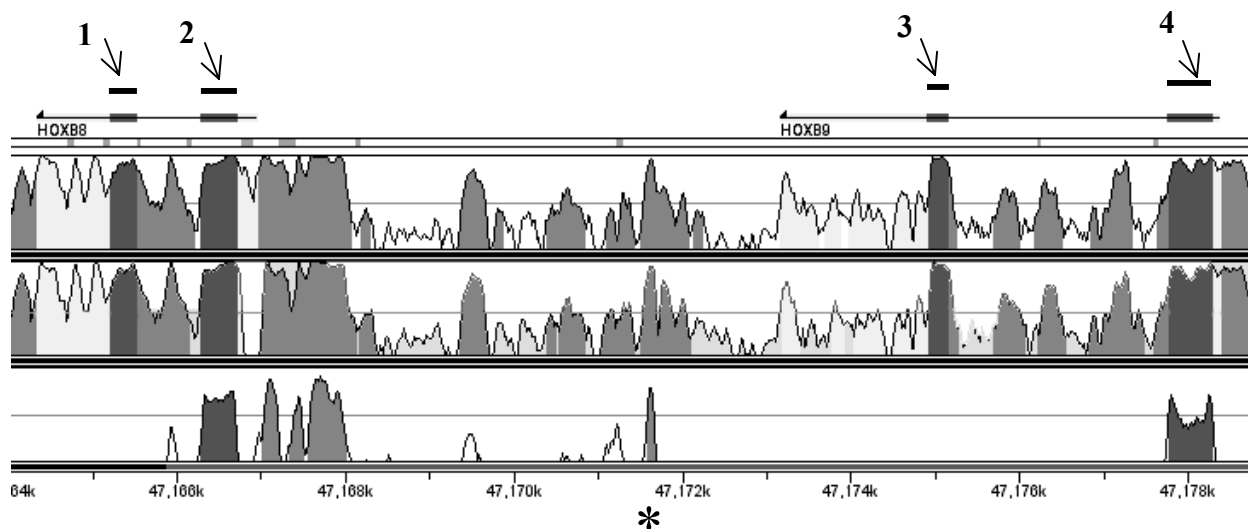


GCB 535 / CIS 535: Introduction to Bioinformatics
Midterm II Solutions
12/15/04

1a. A synonymous mutation is a mutation in a codon that does not affect the amino acid that is coded for. A nonsynonymous mutation changes the amino acid that is coded for.

1b. You can calculate Ka/Ks only in coding regions, since those are the only regions where it makes sense to refer to synonymous versus nonsynonymous mutations. These regions are marked below. You would expect, given the high degree of sequence conservation in regions 2 & 4, that the Ka/Ks between human and capuchin in those regions is low, $\ll 1$. In regions 1 & 3, there is very little sequence conservation, so the ratio would be closer to 1 if the capuchin portion is no longer functional, or greater than 1 if there is functional divergence (positive selection).



2a. You are making the Markov independence assumption, that any nucleotide's state annotation depends only on the state of the one previous nucleotide.

2b. You extract the first 200 nucleotides from all the capuchin 3' UTRs you have available and allow your HMM to "generate" that nucleotide sequence and obtain a probability that the sequence corresponds to the model defined by your HMM. Return as potential targets those UTRs that have probability above a certain threshold, which you can define based on your training set. (To be rigorous, you would want to have a separate test set that you know are targets to assess what this threshold should be)

2c. TP = 10, FN = 30 - 10; Sensitivity = 1/3

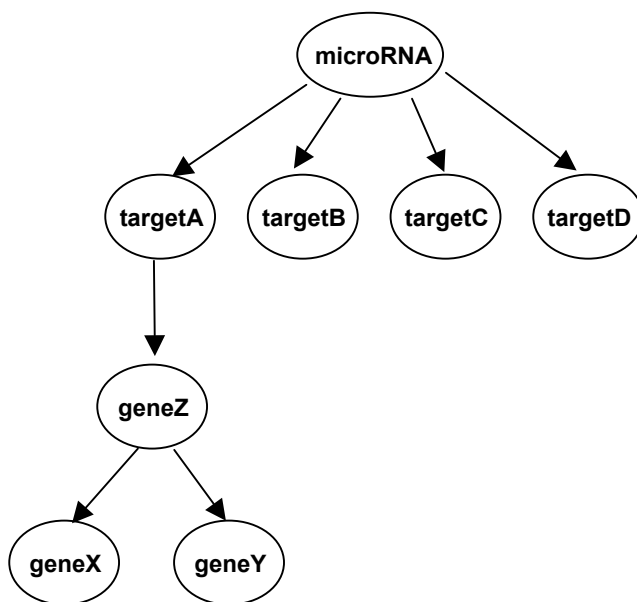
2d. This is essentially the same problem as the one we saw when trying to assess the significance of gene clusters we get from microarray experiments. Obtain Gene Ontology annotations for each of the 10 genes in your set and determine whether you have some overrepresentation of a particular class or classes of genes as compared to random, using the GO tools we covered in lab.

3a. This is a common reference experiment. Two experiments are shown; the first has Cy3 = experimental condition A, Cy5 = reference sample; the second has Cy3 = experimental condition B, Cy5 = reference sample.

3b. We believe that the microRNA affects protein translation, not transcription of mRNA, so the equal ratios between the two conditions tend to support that model that mRNA levels should be unaffected.

3c. GeneX is conditionally independent of the microRNA given targetA, targetB, targetC, AND targetD. You can see this by the fact that you can trace four different paths from the microRNA to geneX, each passing through one of the targets.

3d.



3e. Although we have changed our conditional independences (for example, now geneX is conditionally independent of the microRNA given only targetA), we have not created any absolute independences – every node in the network still has some common ancestor (or more simply in this case, the microRNA still has an effect on everything).

4a. No. Condition A shows high levels of the protein of interest, condition B shows lower levels. We expect expression of the microRNA to inhibit translation of the protein, so we would expect to see the reverse – condition A should show low levels of the protein, and B should show higher levels. There could be other mechanisms at work, such as a protein degradation process that is turned on in Condition B but not in A, or the spot in condition A might actually be a different protein altogether, that has the same mass and isoelectric point as targetA's protein.

4b. LC is a purification / separation step that makes sure you have the purest sample possible before you run the MS, since having multiple proteins in your mix would complicate analysis of the spectrum.

4c. The intensity of a particular peptide fragment – one of the products of trypsin digest from the original protein – indicating that peptide fragment's mass over charge ratio.

4d. Use some method of encoding the spectrum (e.g., fast fourier transform) and use it to compare against a database of spectrum encodings. These artificial mass spectra were created using all known proteins, which are digested with trypsin in silico. This will only be of limited accuracy, since (for example) mass/charge ratios don't necessarily uniquely determine a peptide sequence, and real peptides don't always have expected mass/charge ratios due to post-translational modifications.

4e. The intensity of a particular peptide ion (a, b, c, x, y, or z) fragment – one of the products of ionization from the peptide marked with an asterisk in the full scan – indicating that peptide ion's mass over charge ratio.

4f. The simplest explanation is that one of the tags was generated using y ions, and the other one by b ions – especially likely given the overlapping sequence (LA vs AL) and the overlapping masses. This is essentially generating the sequence from opposite ends.

5a. The most rigorous thing to do would be to assess informativeness based on the significance values assigned to each variable coefficient (feature) in the resulting regression. Coefficients with high p-values will tend to be uninformative, and removing them should not affect the regression much. Note that just looking at the absolute magnitude of the coefficients (i.e., looking for very small coefficients) is not necessarily indicative of informativeness because it depends on what your scale is – 0.1 might seem like a small coefficient when the feature values range from .000001 to .001, but actually has a noticeable effect when you've got very high values such as 1,000,000 to 10,000,000.

5b. Extract the features for your new sample, plug it into the regression equation, and get an output value from the equation. Classify the sample as having a targetable targetD if this output value is above some threshold that you've defined previously by looking at the output values of known targetable and known non-targetable cells.

5c. While you might be able to reproduce the regression using knn, one disadvantage is that knn requires you to store all of your "training" data points so you can actually compute the k nearest neighbors each time you have to classify a new point (expensive). The linear regression provides a much more compact model. However, the linear regression assumes a linear relationship between your features and the classification, and when that is not true, a linear regression can perform very badly. Knn doesn't have this particular assumption, so might be better in such a situation (e.g., when the data is clumped)