

GCB 535 / CIS 535
Fall 2004

Homework 4

Due in class, Wednesday, November 10, 2004

Comparative genomics

1. (6 pts) In Loots's paper (<http://www.seas.upenn.edu/~cis535/Lab/Sciences-Loots.pdf>), the authors used comparative genomic sequence analysis methods to identify conserved non-coding sequences (CNSs), which are potential candidates for distant regulatory sequences. This kind of study can be conducted using VISTA tools.

First, retrieve the nucleotide sequence between 53,240K to 53,277K on mouse Chr11 (contains IL-4 and IL-13 genes) from NCBI map viewer, save it as a FASTA format file.

Then, use the GenomeVISTA tool (<http://pipeline.lbl.gov/cgi-bin/GenomeVista>) to conduct an alignment of the saved nucleotide sequence with the human genome (i.e. use human July 2003 as base genome). Select "mouse" for the choice of organism.

- a) How many CNS regions are reported based on default parameters for the curve (i.e. Min. conservation length = 100, conservation threshold = 70%)? For this question, UTR regions are not considered as non-coding regions.
- b) Based on Figure 2 from the Loots's paper, CNS-1 locates between IL-4 and IL-13. Based on location and the size of the conserved regions, list the most likely CNS region(s) reported by GenomeVISTA that you think corresponds to CNS-1.
- c) In the text browser version output of the GenomeVISTA alignment, you can see a link to rVISTA. Use it to conduct a search for known vertebrate transcription factor binding sites that falls in this region. Select the cut-off to minimize false positives. Which of the predicted TFBs fall in the region corresponding to CNS-1?

Pseudogenes and gene finding

2. (9 pts) A pseudogene can be defined as a DNA sequence that has high sequence similarity to a functional gene, but is itself not functional -- that is, pseudogenes are not transcribed. One way in which pseudogenes are created is through duplication of a piece of the genome and insertion into some other location. The gene(s) that may be contained within that duplicated segment can lose functionality, and mutations (base changes) can accumulate over time. If we assume that the pseudogenes have no function, these base mutations will be randomly and uniformly distributed over the entire pseudogene.

As you might be able to guess, pseudogenes sometimes pose a problem for gene finding algorithms, depending on what sorts of features the algorithms use.

Let's say we have the following four DNA sequences:

- sequence A: contains a functional human gene
- sequence B: contains a paralog of the gene in A that is also functional, but has a different function from that gene
- sequence C: contains a mouse ortholog of the gene in A
- sequence D: contains a pseudogene derived from the gene in A

a) Compare the Ka/Ks ratios you would expect to get if you calculated them with respect to the mouse ortholog to the gene in A; the mouse ortholog to gene B; and the mouse ortholog to the pseudogene. Briefly explain why these ratios are similar or different.

b) Compare the lengths of the longest ORFs you would expect to get when analyzing sequence A versus sequence D, which contains the pseudogene. Explain why they are different.

c) Let's say you have several ESTs derived from the mRNA product of the gene in A. Compare the sequence similarity scores (using, for example, Needleman-Wunsch) you would expect to get when comparing the ESTs to sequence A versus the ESTs to sequence B versus the ESTs to sequence D, and explain the differences or similarities.

Linear regression

3. (15 pts) Linear regression can be used to predict the location of the Translation Initiation Site (i.e., the start codon) of a gene. Although all start codons are ATG, non-start ATG codons also appear in many other places along the gene, since ATG codes for an amino acid also, and there are plenty of non-coding regions along the gene in which an ATG triplet may be found. The TIS identification task is to choose the one ATG triplet that in fact is the start codon for the gene.

Several features can be used for predicting the TIS, and we will look at how to incorporate all of them into one predictor using linear regression.

We have provided two files – a training set and a test set, in plain text format and in Microsoft Excel format, downloadable from <http://www.seas.upenn.edu/~cis535/HW/tis/>. The training set consists of one training instance per line. Each training instance contains one value for each feature that we're looking at (6 total) and a labeling in the final column saying whether that instance is or is not a TIS (1 means it is a TIS, 0 means it is not).

The features are: a neural network score for how likely the ATG is a TIS, based on particular surrounding nucleotide motifs; a value for how much more likely the sequence downstream of the ATG codes for a protein product (coding potential) versus the upstream sequence (also a neural net output); the distance in nucleotides from the ATG and the second splice donor found upstream of the ATG; the distance between the first splice acceptor and first splice donor found downstream of the ATG; the coding potential difference between upstream and downstream of the second splice donor found upstream of the ATG; and the coding potential difference between upstream and downstream the first splice acceptor found downstream of the ATG.

The test set contains a smaller number of instances labeled the same way as the training set. You will use these to evaluate the output of your linear regression. The Excel version of this file has some additional fields that will make it easier to do this evaluation; folks who want to use a different application are on their own.

a) Explain the second feature above – the coding potential downstream the ATG versus the coding potential upstream the ATG. Why is this a useful feature, and what would you expect to find if the ATG is a true start codon versus if it is not (HINT: there are two cases you might want to think about for when an ATG is not a start codon).

b) Do a linear regression on the training set. A useful reference if you haven't done this before in Excel is at <http://www.ksu.edu/stats/tch/malone/computers/excel/analyses/regression2.html>

Report the coefficients to three decimal places and the R square value for the regression. Do you think the regression produced a good fit to the data? Why? Which variables do you think are least predictive and why?

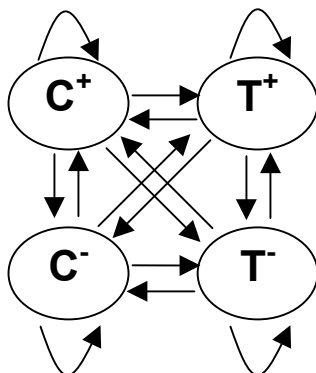
c) Now use the results of the regression to predict whether each of the test instances is a TIS. Excel users, simply put the values for each of the coefficients into the corresponding fields in the spreadsheet; the prediction will automatically be calculated for each instance. Non-Excel users, remember that you are summing the product of the coefficient and the value for each variable, and finally adding the intercept, to get your predicted output value.

Note that your predicted output will not be 0 or 1 but some number in between. You will then need to define a threshold, such that any output above the threshold is classified as a TIS, and below that threshold as a non-TIS.

First use a threshold of 0.6, and report the number of true positives, true negatives, sensitivity ($TP / TP + FN$) and specificity ($TN / TN + FP$). Then repeat for a threshold of 0.4.

Hidden Markov models

4. (9 pts) HMMs can be used to detect CpG islands in a way that does not depend on using any particular sliding window size. Let's consider the simpler problem of detecting "C islands" in a genome that contains only Cs and Ts. The following state transition diagram and Markov matrix define the HMM:



	C+	T+	C-	T-
C+	0.7	0.8	0.1	0.1
T+	0.2	0.1	0.1	0.1
C-	0.05	0.05	0.4	0.4
T-	0.05	0.05	0.4	0.4

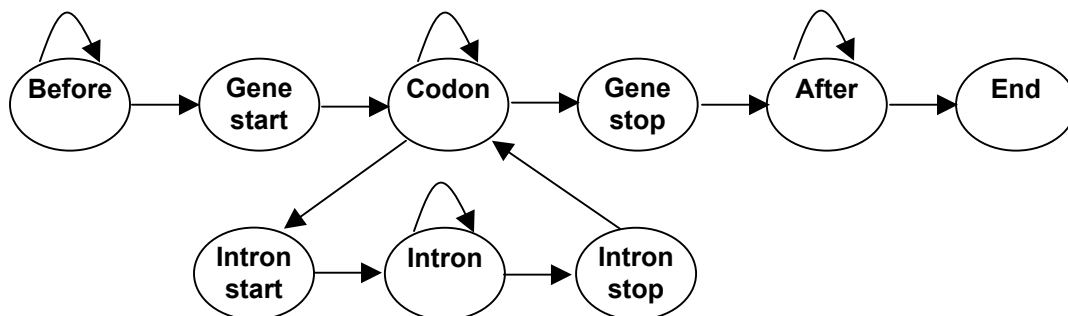
There are four states: C+ is a C nucleotide in a “C island”; T+ is a T nucleotide in a “C island”; C- is a C nucleotide in a non-“C island”; and T- is a T nucleotide in a non-“C island.” The C+ and C- states can only emit a “C” while the T+ and T- states can only emit a “T,” so the probability of any legal emission from a state is 1. To read the transition probabilities from the table, the probability of transitioning to state T+ given that you were previously in state C+ is 0.2. (In this case, “to” states are the rows in the table, “from” states are the columns). Thus, the probability of any sequence according to some order of state transitions is simply the product of the appropriate transition probabilities used to generate that state order (for those of you who are purists, we are indeed ignoring the prior probability term for the first nucleotide in the sequence).

Suppose you had the nucleotide sequence TTTCTCCCC. Calculate the probability of that sequence using the HMM according to each of the following cases.

- The entire sequence is predicted to be in a C-island
- The entire sequence is predicted not to be in a C-island
- The first five nucleotides are not in a C-island, but the last four nucleotides are.

The highest probability from those three cases will indicate the most likely state transition model to describe your sequence of the ones we’ve seen, though we haven’t tried the other $2^9 - 3$ other possible state transition models. There is a dynamic programming algorithm called Viterbi that allows efficient identification and calculation of the best order of state transitions for the nucleotide sequence in question, without having to enumerate all of the possible cases.

5. (12 pts) Recall back to the (imperfect) architecture of an HMM used to detect genes in genomic sequence:



Remember that an HMM is a “generative” model – we make decisions (e.g., is this a gene with high probability) based on what sort of path we can trace through the various states. One path might be

Before → Gene start → Codon → Codon ... → Codon → Gene Stop → After → After ... → After → End,

which would correspond to a gene with no introns. Other paths may include one or several introns, which would correspond to looping around the Intron start → Intron → Intron ... → Intron → Intron stop path several times. Which path is the most highly probably depends on the transition probabilities between states and the probabilities that the observed sequence is “emitted” at each state, as you saw in the previous question, but ultimately this depends on you HMM architecture – what states are connected to each other and in what direction. For example, according to the model above, it is impossible to transition from the Intron state to the Gene Stop, since all genes must end with an exon.

a) To make sure you remember what genes look like, what is biologically wrong with making the only non-intronic state in the above model a “Codon” state?

b) We can use an HMM to predict protein secondary structure (one level of their 3-dimensional shape). Let’s take an overly simplistic (and slightly wrong) model of protein secondary structure and attempt to create an HMM architecture that can be used for the prediction problem. Assume the following:

- Secondary structure is determined entirely by the amino acid sequence of the protein
- There are two structural motifs: helix and strand. All secondary structures can be decomposed into a series of helices and strands. No two helices may occur next to each other without being separated by at least one strand, but there can be any number of helices and strands (including none) in the protein as long as there is at least one motif (helix or strand). For example, helix-strand-helix-strand-strand-strand-helix is a legal protein structure, as is strand-helix-strand. Helix-helix is NOT.
- Proteins may optionally have localization “tags” at the beginning of the sequence that do not contribute to the secondary structure (these tags are signals to transport the protein to one place or another inside the cell). A protein may have zero, one, or two such tags.

Based on these assumptions, draw an HMM architecture that could be used to assign a secondary structure to a protein sequence. You will probably need to define a “start state” and an “end state” to mark the start and end of your protein.

c) What are the possible emissions? Are these the same at each state?