7.36/7.91/20.390/20.490
Computational and Systems Biology

Homework 2

Due: Tuesday, February 27, 2007

General

Files to be downloaded for problem sets can be found on the course website at:
http://stellar.mit.edu/S/course/7/sp07/7.91J/homework/index.html

Python Scripts

All Python scripts must work on athena using /usr/athena/bin/python. Scripts using input files must be able to run if placed in a directory also containing these input files, and should not alter the contents of the host file system (i.e. should not add, delete, or modify any files) unless explicitly requested by the question in the problem set. Electronic submissions are subject to the same late homework policy as outlined in the syllabus and submission times are assessed according to the server clock.

All python programs must be submitted electronically on the course website. To submit a file electronically:

1. Go to https://stellar.mit.edu/S/course/7/sp07/7.91J/homework/index.html
2. Click on the corresponding problem set.
3. On the Assignment Details page, click the Add Submission link.
4. On the Add Submission page, select the appropriate file on your computer. Alternatively, you can paste your script/file into the window provided.
5. Click Submit when ready.

Written Portion

The written portion of the problem set must be turned in during the first 15 minutes of lecture on the due date. Late problem sets should be turned in to the box located in the Education Office on the first floor of building 68 (see the syllabus for late homework policy).

Problems

Question 1 - Dynamic Programming

For this problem, you may either copy the partially completed tables into your writeup or attach the completed problem printout to the end of your writeup.

You and your TA Robin have decided to align two protein sequences (FENDER and DEER) by hand for fun. Robin got started, filling in the global dynamic programming matrix using a linear gap penalty of -2 and the BLOSUM62 matrix available in the lecture notes. However, he left halfway through to eat a couple of delicious burritos.

(a) Complete the alignment by filling in the matrix with traceback arrows (for every square, one of ↓, →, ↘, or multiple arrows if there is a tie), circling the final alignment path. Next, write out the aligned sequences, one on top of the other, with any gaps represented by dashes.

	-	F	E	N	D	E	R
-	0	-2	-4	-6	-8	-10	-12
D	-2	-3	0	-2	0	-2	-4
E	-4	-5	2	0	0	5	3
E	-6	-7	0	2	2	5	5
R	-8	-9	-2	0	0	2	10

(b) You decide that you'd rather have a local alignment of these two proteins than a global one. Fill in the table below with the local dynamic programming alignment, again using BLOSUM62 and a -2 linear gap penalty. Again, draw the traceback path, circle it, and write the final alignment. If there is a tie, write out all optimal alignments. Hint: Most of the numbers in the local alignment matrix will be different than in the global one.

	-	F	E	N	D	E	R
-							
D							
E							
E							
R							

Question 2 - Nucleotide substitution models

In class, we saw various models for nucleotide substitution, like the Jukes-Cantor and the Kimura models. These models make different assumptions about the probability of a given nucleotide mutating into another nucleotide. To see what kind of impact these assumptions can have, examine the following transition probability matrices:

Model 1:

	A	G	C	T
A	0.88	0.04	0.04	0.04
G	0.04	0.88	0.04	0.04
C	0.04	0.04	0.88	0.04
T	0.04	0.04	0.04	0.88

Model 2:

	A	G	C	T
A	0.88	0.06	0.03	0.03
G	0.06	0.88	0.03	0.03
C	0.03	0.03	0.88	0.06
T	0.03	0.03	0.06	0.88

(a) What assumptions about nucleotide mutation rates are being made by these two models? What are the benefits of using each kind of model?

(b) Like PAM1, this matrix represents mutation probabilities over one unit of time. For a particular A nucleotide at time t=0, write out the possible paths by which it could become a G at time t=2. For example, one possibility is A→ G followed by G→ G. Which of these scenarios are more likely with model 2 than with model 1?

(c) Suppose that at time $t = 0$, a certain stretch of DNA econdes the amino acid methionine with the nucleotides ATG. For each of the models above, calculate the probability that, at time $t = 2$, the stretch of DNA has evolved to encode glycine, with GGC. Assume each position evolves independently. (Hint: this problem is easier if you use matrix multiplication)

(d) For each of the models above, what is the probability that a “C” nucleotide is a “C” after a very long time ($t \to \infty$)? What method could you use to determine this probability for any arbitrary nucleotide substitution matrix?

Question 3 - BLAST statistics

This problem will allow you to apply what you have learned about sequence alignments, how BLAST works, and the alignment statistics.

(a) You are reconstructing the neanderthal genome and have isolated a fragment of DNA that you would like to compare with modern-day genomes. The sequence is contained in the file **DNA-sequence.fa** on the course website. Use nucleotide-nucleotide BLAST (blastn) to search for this fragment with default options in the human genomic plus transcript database. What is the E-value of the top hit? Do you think this is significant? Why or why not?

(b) Now go back and BLAST with the same database, but with a word size of 7 instead of the default 11. What is the E-value of the top hit? Do you think this is significant? Looking at this top alignment, apply what you know about the BLAST algorithm to guess why the top hit is different from part **a)**.

(c) You have translated a different portion of the gene into a protein sequence, found in **protein-sequence.fa**. Use protein-protein BLAST (blastp) to search for the fragment with default parameters in the *nr* database. What is the bit score and E-value of the top hit? Scroll down to the bottom and report the value of λ and κ (gapped).

(d) Now go back and BLAST the fragment against the *refseq* database instead. What is the bit score and E-value of the top hit now? Why is the bit score the same and the E-value different?

(e) Suppose you only care about hits with E-values less than or equal to 10^{-8}. Using the values of λ and κ from part **c)**, what is the lowest bit score for an alignment that would satisfy this condition if you searched against the *nr* database? What if you searched against *refseq*?

Question 4 - Constructing substitution matrices

In this problem you will write a python script for deriving a substitution matrix from a multiple sequence alignment (MSA) of evolutionarily related proteins. You will follow the approach taken by Margaret Dayhoff for deriving PAM matrices. An MSA you can use for testing your program is posted on the course website in the file **evolution-msa.txt**. You can assume that the alignment given to you contains no gaps and all sequences are of the same length (one sequence per line, no spaces, single-letter amino acid code). Also assume that your program will be called with exactly one parameter, the name of the file containing the MSA to derive substitutions from (see sample run below). The goal of your program will be to print to the screen values that are derived from this MSA. Your program should produce output formatted exactly as shown in the sample run.

Before you panic, note that you have been given a starting point with the file **pam.py** on the course website. You have been given a skeleton file containing some code, some functions, and also pseudocode (in comments prefaced by ##)for the sections you must complete yourself. Pseudocode is a description of what your code should do, more or less in english. All you will need to do is translate the pseudocode into python syntax and then correctly format the output.

Importantly, the formulation and notation used here is slightly different from that shown in lecture. For our purposes, it is equivalent. Please try to match the formulation below.

Hints:

- Carefully read through and understand the functions provided for you; these will help give you correct syntax for using dictionaries, lists, and the print command.
- Run your program on the test data after each step to ensure that that part is working before moving on.
- Proficient programmers do not need to use the code or pseudocode provided, but still must title their script **pam.py** and match the output exactly.

(a) First, you will need to count the number of occurrences of each amino acid in your alignment and to convert occurrences into frequencies. Run through each amino acid alphabetically and print its fraction of the total number of amino acids as in the sample run. That is, if $occ[a_i]$ = the number of occurrences of amino acid a_i, then its frequency $f_i = \dfrac{occ[a_i]}{\displaystyle\sum_j occ[a_j]}$

(b) You will also need to count the number of substitutions (or changes) occurring for all pairs of amino acids, i.e. all those in the same column.

Since we do not know which direction a change happened (i.e. whether it was $a_i \rightarrow a_j$ or $a_j \rightarrow a_i$), we assume symmetry and count the substitution observed as both $a_i \rightarrow a_j$ and $a_j \rightarrow a_i$. Here, the number of times a_j could have changed into a_i will be denoted by $A[a_i a_j]$. Print the contents of this matrix to the screen as in the sample run.

(c) Next, you have to calculate the mutability of each amino acid. The mutability of amino acid i,

$$m[a_i] = \frac{\#\text{times } a_i \text{ changes}}{\#\text{occurrences of } a_i} = \frac{\sum\limits_{j \neq i} A[a_j a_i]}{occ[a_i]}$$

Print the amino acid mutabilities to the screen as in the sample run.

(d) In order to build a PAM1 matrix (i.e. a substitution matrix corresponding to an evolutionary period of 1 PAM), you have to normalize your frequencies such that there is on average 1 mutation expected per 100 amino acids. For this, you have to find the appropriate value of the evolutionary scale factor λ. This value must satisfy the equation:

$$\lambda \sum_j f[a_i] m[a_i] = 0.01, \text{ or } \lambda = 0.01 / \sum_j f[a_i] m[a_i]$$

Find the appropriate value for λ and print it to the screen (exactly as shown).

(e) You now have all the necessary components to calculate the PAM1 matrix.

The probability of substitution from amino acid a_j to a_i, called $M[a_i a_j]$, will be defined by:

$$M[a_i a_j] = \begin{cases} 1 - \lambda \cdot m\lfloor a_j \rfloor & \text{if } i = j \\ \\ \dfrac{\lambda \cdot m\lfloor a_j \rfloor \cdot A\lfloor a_i a_j \rfloor}{\sum\limits_{k \neq j} A\lfloor a_k a_j \rfloor} & \text{if } i \neq j \end{cases}$$

Print the matrix of M (as percentages) as shown in the sample run.

(f) You now have a PAM1 matrix. Finally, you will calculate the PAM5 matrix that corresponds to 5 times the evolutionary distance. To do this, simply raise your M matrix to the fifth power using standard matrix math. Print the final PAM5 matrix (as percentages) to the screen as in the sample run.