

Biostatistics & Medical Informatics / Computer Sciences 576
Introduction to Bioinformatics
Fall 2003 Midterm Exam

Name: _____

Write your answers on these pages and show your work. You may use the back sides of pages as necessary. Before starting, write your name on every page of the exam. Also, make sure your exam has every page (numbered 1 through 6).

Problem	Score	Max Score
1.	_____	16
2.	_____	16
3.	_____	20
4.	_____	8
5.	_____	20
6.	_____	20
Total	_____	100

1. Pairwise Global Alignment:

1a. (12 points) Show how the dynamic programming approach would be used to determine a global alignment for the two sequences below. Matching bases should be scored +1, mismatching bases should be scored -1, and each gap position should be penalized -2 (i.e. the gap penalty function is linear). Show the filled-in scoring matrix and the traceback pointers in the matrix.

x: GTCG

y: CGT

			C	G	T
	0	← -2	← -4	← -6	
G	↑ -2	↖ -1	↖ -1	← -3	
T	↑ -4	↖↑ -3	↖ -2	↖ 0	
C	↑ -6	↖ -3	↖↑ -4	↑ -2	
G	↑ -8	↑ -5	↖ -2	←↑ -4	

1b. (4 points) For these two sequences and the scoring scheme used above, show all of the optimal global alignments.

G T C G -
 - - C G T

- G T C G
 C G T - -

2. Pairwise Local Alignment:

2a. (12 points) Show the filled-in scoring matrix and traceback pointers for a local alignment of the same two sequences using the same scoring system: matching bases should be scored +1, mismatching bases should be scored -1, and each gap position should be penalized -2.

x: GTCG

y: CGT

		C	G	T
	0	0	0	0
G	0	0	1	0
T	0	0	0	↖ 2
C	0	1	0	0
G	0	0	↖ 2	0

2b. (4 points) For these two sequences and the scoring scheme used above, show all of the optimal local alignments.

C

G	T
G	T

 C G

G T

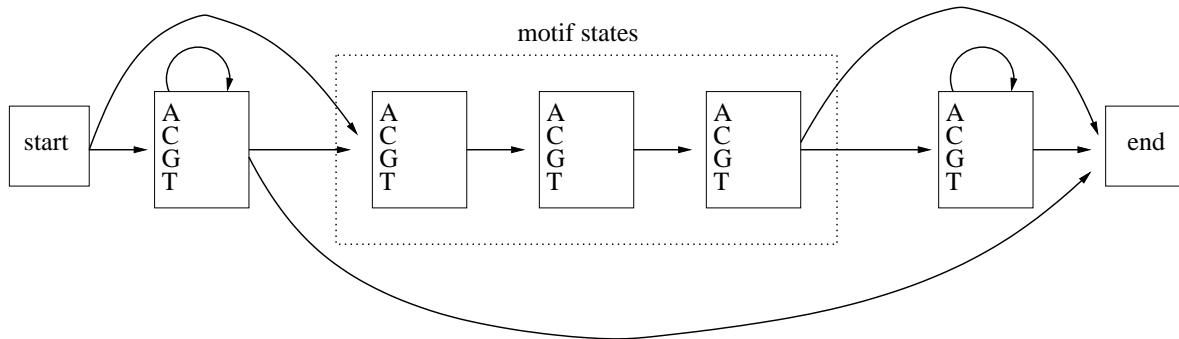
C	G
C	G

 T

3. Hidden Markov Models:

3a. (15 points) Suppose we want to design a hidden Markov model (HMM) to recognize occurrences of a particular type of sequence motif in DNA sequences. For example, we might want to recognize the sites in a genome at which a particular protein binds to the DNA. Suppose that we know that the binding sites are always three bases in length, and every sequence we search with this model will contain either one or zero occurrences of the binding site. Draw a picture showing the graphical structure of an HMM for this task. Your solution should use only one HMM.

- Show all of the states of the model, and for each, indicate what it is intended to represent.
- If your model will use emission probabilities, indicate which states will have them.
- Show all of the allowable transitions in the model.



There are various ways to do this. Here is one straightforward scheme. All states, except *begin* and *end* have emission probabilities. The three states in the middle represent the motif positions and the two bracketing states represent non-motif sequence.

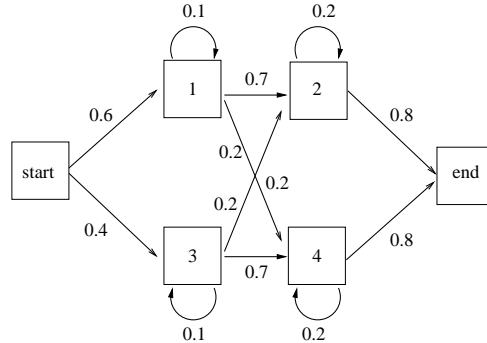
3b. (5 points) Assume that the model from **3a.** has been trained, and you are now given a set of sequences in which you want to predict occurrences of the binding site. Briefly describe how you would do this?

For each sequence you would run the Viterbi algorithm to find the most probable path for that sequence. Then you follow the Viterbi path to see if it passes through the *motif* states. If it does, then your motif prediction would correspond to the sequence positions that are aligned with these three states.

4. The BLAST Algorithm (8 points): Briefly describe one key feature of the BLAST algorithm that distinguishes it from dynamic-programming methods for pairwise alignment.

There are various aspects of BLAST that you could discuss here. The most significant difference between BLAST and DP is that BLAST first identifies short, ungapped “seeds” for potential alignments and then expands arounds these seeds to find more extensive local alignments. BLAST finds these seeds by searching for exact matches and near matches to short words that occur in the query sequence. This procedure may result in decreased sensitivity (BLAST may miss some significant alignments), but BLAST is usually much faster than dynamic programming.

5. The Backward Algorithm (20 points): Consider the hidden Markov Model shown below. The transition probabilities are shown in the figure, and the emission probabilities are shown in the table. The *start* and *end* states are silent. Show the values calculated by the **Backward** algorithm for the sequence **AGT**.



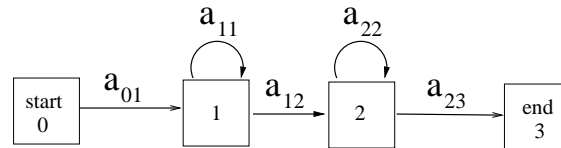
state 1	state 2	state 3	state 4
$e_1(A) = 0.5$	$e_2(A) = 0.2$	$e_3(A) = 0.3$	$e_4(A) = 0.1$
$e_1(C) = 0.2$	$e_2(C) = 0.2$	$e_3(C) = 0.3$	$e_4(C) = 0.1$
$e_1(G) = 0.1$	$e_2(G) = 0.4$	$e_3(G) = 0.2$	$e_4(G) = 0.1$
$e_1(T) = 0.2$	$e_2(T) = 0.2$	$e_3(T) = 0.2$	$e_4(T) = 0.7$

$$\begin{aligned}
 b_2(3) &= a_{2,end} &&= 0.8 \\
 b_4(3) &= a_{4,end} &&= 0.8 \\
 b_2(2) &= a_{22}e_2(T)b_2(3) &&= 0.2 \times 0.2 \times 0.8 = 0.032 \\
 b_4(2) &= a_{44}e_4(T)b_4(3) &&= 0.2 \times 0.7 \times 0.8 = 0.112 \\
 b_1(2) &= a_{12}e_2(T)b_2(3) + a_{14}e_4(T)b_4(3) &&= (0.7 \times 0.2 \times 0.8) + (0.2 \times 0.7 \times 0.8) = 0.224 \\
 b_3(2) &= a_{32}e_2(T)b_2(3) + a_{34}e_4(T)b_4(3) &&= (0.2 \times 0.2 \times 0.8) + (0.7 \times 0.7 \times 0.8) = 0.424 \\
 b_1(1) &= a_{11}e_1(G)b_1(2) + a_{12}e_2(G)b_2(2) + a_{14}e_4(G)b_4(2) &&= (0.1 \times 0.1 \times 0.224) + (0.7 \times 0.4 \times 0.032) \\
 &&&+ (0.2 \times 0.1 \times 0.112) = 0.01344 \\
 b_3(1) &= a_{33}e_3(G)b_3(2) + a_{32}e_2(G)b_2(2) + a_{34}e_4(G)b_4(2) &&= (0.1 \times 0.2 \times 0.424) + (0.2 \times 0.2 \times 0.032) \\
 &&&+ (0.7 \times 0.1 \times 0.112) = 0.01888 \\
 b_{start}(0) &= a_{start,1}e_1(A) * b_1(1) + a_{start,3}e_3(A)b_3(1) &&= (0.6 \times 0.5 \times 0.01344) \\
 &&&+ (0.4 \times 0.3 \times 0.01888) = .0062976
 \end{aligned}$$

6. The Forward-Backward Algorithm (20 points): Consider the hidden Markov Model shown below. The *start* and *end* states are silent, but both *state 1* and *state 2* have emission parameters for all four DNA bases. Suppose that we are learning the emission and transition parameters using the **Forward-Backward** (Baum-Welch) algorithm. We are training using two sequences, **ACG** and **TC**, which we denote x and y respectively. Assume that we have run the Forward and Backward algorithms on both sequences and calculated all of the relevant f and b values for sequence x and sequence y (here the superscript on the f and b values indicates which sequence the calculation is for):

$$\begin{aligned}
 & f_0^x(0), f_1^x(1), f_1^x(2), \dots, f_3^x(3) \\
 & b_0^x(0), b_1^x(1), b_1^x(2), \dots, b_3^x(3) \\
 & f_0^y(0), f_1^y(1), f_2^y(2), \dots, f_3^y(2) \\
 & b_0^y(0), b_1^y(1), b_2^y(2), \dots, b_3^y(2)
 \end{aligned}$$

Given these values, show how the Forward-Backward algorithm updates the transition probabilities for the transitions emanating from *state 1*.



	ACG	+	TC
$n_{1 \rightarrow 1}$	$\frac{f_1^x(1)a_{11}e_1(C)b_1^x(2)}{f_3^x(3)}$	+	0
$n_{1 \rightarrow 2}$	$\frac{f_1^x(1)a_{12}e_2(C)b_2^x(2)+f_1^x(2)a_{12}e_2(G)b_2^x(3)}{f_3^x(3)}$	+	$\frac{f_1^y(1)a_{12}e_2(C)b_2^y(2)}{f_3^y(2)}$

$$a_{11} = \frac{n_{1 \rightarrow 1}}{n_{1 \rightarrow 1} + n_{1 \rightarrow 2}}$$

$$a_{12} = \frac{n_{1 \rightarrow 2}}{n_{1 \rightarrow 1} + n_{1 \rightarrow 2}}$$