Student ID: _____ Student Name: _____

CompSci 369: Computational Biology
Midterm Test 2007

1 Short Answers

1. Homology and similarity

   (a) Two genes from the same genome are found to be homologous. Is the homology orthologous or paralogous? Explain.

   (***3 marks***)

   (b) Explain how sequence alignment can be used to infer homology.

   (***5 marks***)

2. Modeling genetic change

   (a) Explain the Jukes-Cantor genetic distance:

   (***3 marks***)

   (b) What is the p-distance?

   (***3 marks***)

(c) Can the Jukes-Cantor distance be calculated from the p-distance?

(***1 marks***)

3. Dot plots and sequence alignment

(a) What does a dot in a dot plot signify? If the window size is w and the two sequences are length $s > w$ and $t > w$, how many substring comparisons does the dot plot represent?

(***5 marks***)

(b) Explain the difference between local and global pairwise alignment and give an example of when you would use each:

(***5 marks***)

(c) What are the main differences between the progressive alignment methods of PILEUP and CLUSTALW?

(***5 marks***)

2 Algorithms

1. We want to design a dynamic programming algorithm that, given a sequence $a_1, a_2, \ldots, a_n$ of real numbers, finds a *clean subsequence* $a_{i_1}, a_{i_2}, \ldots, a_{i_t}$ of elements so that the sum $\Sigma_{j=1}^{t} a_{i_j}$ is a maximum. The subsequence is *clean* if $i_j < i_{j+1} \leq i_j + 2$ for all indices $1 \leq j < t$. In other words, the sequence is almost consecutive—allows gaps of size at most 1. If a_i is negative for all i, we define the maximum sum to be zero (which is obtained by selecting the empty subsequence).

   (a) For the sequence $(3, -4, \pi, -3.5, 8, -3, -2.5, 2\sqrt{2}, -3.2, -10, 5.5)$ find a clean subsequence with the largest sum.

   (***3 marks***)

   (b) For a given sequence of length n, consider the number of differenent clean subsequences that it may contain. Is this count a *polynomial* or *exponential* function of n? Explain why. [Hint: observe that there are $\binom{n}{2} + n + 1 = O(n^2)$ possible different consecutive subsequences.]

   (***3 marks***)

   (c) Explain precisely what are the parameters and subproblems with respect to your dynamic program design.

   (***3 marks***)

   (d) Give a high level description and informal justification of correctness of your algorithm, but not the implementation details.

   (***4 marks***)

(e) Assuming each of the real numbers can be encoded with a constant number of bits, what is the running time of your algorithm? Express your answer in terms of $\Theta(f(n))$ notation, for some simple polynomial function f.

(***2 marks***)

2. Recall the n-Queens problem, where we want to place n non-attacking queens on a $n \times n$ chess board. A queen may attack horizontally, vertically and diagonally any number of squares.

   (a) For the $n = 5$ case illustrate (by placing "Q" in the cells) the next several moves of a backtracking algorithm starting at the following board configuration.

2 queens:

	Q			
				Q

3 queens:

	Q			
				Q
Q				

	Q			
				Q
	-	**Q**		

4 queens:

	Q			
				Q
Q				
-	-	-	**Q**	
-	-	-	-	

	Q			
				Q
		Q		
Q				

5 queens:

	Q			
				Q
		Q		
Q				
-	-	-	**Q**	

(***9 marks***)

   (b) Recall that we can use a permuation on $\{1, 2, \ldots, n\}$ to represent an embedding of n queens on a $n \times n$ chess board. That is, the i-th element corresponds to which column to put a queen on the i-th row. For $4 \leq n \leq 6$ list the first solution of the n queens problem by filling in the least lexicographic permutation in the following table:

n	**permutation on** $\{1, 2, \ldots, n\}$
4	2 4 1 3
5	1 3 5 2 4
6	2 4 6 1 3 5

(***6 marks***)