CICS 436/636 Computational Biology & Bioinformatics
Fall 2016
Homework 1

Due date: October 4, 2016

1. Write a program that simulates the evolutionary processes, and use it to estimate PAM matrices. PAM1 is given at http://cis.udel.edu/~lliao/cis636f16/pam1.txt, and is used as an evolutionary model. That is, a) the rate is one mutation per one hundred amino acids; b) for amino acid *i*, the probability to mutate to amino acid *j* is given by the corresponding PAM1 element PAM1(*i*,*j*).

Step 1. Write a subroutine "rand_seq" to generate random protein sequences of a given length, say 500. You can assume that amino acids are distributed according to the diagonal elements given in PAM1. That is, the probability for amino acid *i* is given as $p(i) = PAM1(i,i)/[\Sigma_{j=1 \text{ to } 20} PAM1(j,j)]$, for i = 1 to 20.

Step 2. Write a subroutine "mutate_seq" that takes a protein sequence as argument, scans through the input sequence, performs a mutation at each position as specified by the evolutionary model PAM1, and outputs the mutated sequence.

Step 3. Call the above subroutine "mutate_seq" iteratively for 50 times. Use a sequence, let's call it "ancestor" which is generated by "rand_seq" from step 1, as the input to the first call of "mutate_seq". Afterwards, in each iteration, use the output of previous iteration as input. The output sequence of the 50th iteration is named as "mutant1". Repeat this whole process using the same "ancestor" as the initial input. The output sequence is named "mutant2".

From the alignment, generate a substitution scoring matrix as defined by

$$S(i,j) = \log [f(i,j) / (f(i) f(j))]$$

where f(*i*,*j*) is the probability that amino acids *i* and *j* are aligned in the alignment, and f(*i*) and f(*j*) are the probabilities that *i* and *j* appear in the sequences respectively.

Report your scoring matrix and compare with PAM50, which is given at http://cis.udel.edu/~lliao/cis636f16/pam50.txt

2. Consider two sequences x= TAGGACATG and y = CACGTACG and the following scoring scheme:
   Identity: +4
   Transition: -2
   Transversion: -3
   Gap: -8

a) Align the sequences x and y using the Needleman-Wunsch algorithm. Report the best score and all alignments achieving that score. You need to show the dynamic programming (DP) matrix.

b) [Section 636 only] Align the same two sequences given above with Smith-Waterman algorithm.

3. Implement the Needleman-Wunsch algorithm for global alignment of DNA sequences. You are strongly encouraged to use Perl. You can hard code the scoring scheme in Problem 2.

   Specifics:

   i. Get sequences from input file (in FASTA format), and write to the standard output.
   ii. Command line option **`-o 1`** to output the DP table. The default is to report only the best alignment and score.
   iii. Name your script as "xxxx_align", where xxxx is your last name and first name initial.

   Synopsis: **`xxxx_align [-o 1] <input_file>`**

   Test sequences can be found at http://cis.udel.edu/~lliao/cis636f16/test_seqs.txt

4. Gene finding. Download the DNA sequence from /~lliao/cis636f14/hw1_sequence.
   a. Scan the sequence and list the ***longest three open reading frames*** (ORFs); an ORF is a sequence segment delimited by a Start codon on 5' and a Stop codon on 3', with no Stop codon in between.
   b. Run BLAST search on these ORFs against the Genbank (http://blast.st-va.ncbi.nlm.nih.gov/Blast.cgi). For each ORF, extract and report the following information about the **top hit** from BLAST search: gene function (in the header line starting with >), e-value, bit score, percent identity, and the starting position of the match on the query sequence (a.k.a. the ORF), and report if the starting position corresponds to the first occurrence of AUG in the ORF.

Instruction for submission: For Problems 1 and 3, you should hand in hardcopy of final results and email your code (along with a readme file) to the TA.