Biol 591 Introduction to Bioinformatics (Fall 2003)
Problem Set 3 – Blast

This problem set is somewhat smaller than usual (bearing in mind that it combines questions concerning bioinformatics, molecular biology, and programming) because of the heft of the study questions. Please understand that study questions are to be given the same consideration as questions on problem sets and are deemed to be part of this problem set.

P3.1. What can you use to determine whether a string of characters in a Perl program is:

a. a comment **c.** a subroutine **e.** a regular expression
b. a variable **d.** a call to a subroutine **f.** an array

P3.2. Discover! Consider the following below taken from the Main Program of *BlastN*:

```
while ($target =~ /$query_word_pattern/g) {
   $target_word_start = pos($target) - $word_length;
   Process_match($target_word_start, $query_word_start);
   pos($target) = $target_word_start+1;
}
```

Write a minimal program to help you answer the questions below:

2a. What does pos($target) mean in the second line?

2b. What does pos($target) mean in the fourth line?

P3.3. Modify BlastN so that it no longer prints out a complete match but prints out instead only each initial exact match of a word.

P3.4. Examine BlastN and determine the values used for the following quantities:

a. Match reward **c.** Gap open penalty **e.** Word size
b. Mismatch penalty **d.** Gap extension penalty

P3.5. Modify BlastN so that it prints out for each hit both the raw score and the score in bits. To do this you may need to find values for lambda and K. Do this by running ANY pairwise sequence comparison at the NCBI site, using the same parameters you use in local BlastN, and noting the values of lambda and K at the end of the output.

P3.6. What is a frequently sighted amino acid sequence that aligns with the amino acid sequence DIVIT to give a score of 13 using BLOSUM62 as the scoring table. (see notes as a source of BLOSUM62)

P3.7. Modify *BlastN* so that it will check for accuracy the scoring table you calculated in class. See Scenario 3 web page for copy of scoring table.

P3.8. Estimate how much more efficient BlastN is than a full Smith-Waterman algorithm. Proceed as follows.

A. Presume that the total time spent by each program is proportional to the number of cells in scoring tables each has to calculate (so your job is reduced to figuring out how many cells that is in each case).

B. Consider a specific case of a comparison of a 100-nucleotide query sequence with the E. coli genome. How big would the Smith-Waterman scoring matrix be?

Don't know how big the E. coli genome is? You have a program that can tell you! Recall that SequenceSearch *reads in the genome of Nostoc in order to search for putative NtcA binding sites. Well, perhaps you can change where it reads in the Nostoc sequence and have it read in the E. coli sequence. Once the sequence is in a variable you can add the line*

print length(variable);

and you have it! (you'll have to put in the right name in place of variable).

C. OK, you got half the job done. Now you need to find out how many cells Blast would need to calculate. First of all, how many word matches would you expect Blast to find? Consider two cases: a word-size of 11 and a word-size of 7.

How do you find how many exact word matches there will be? Again, you have a program for the job – in fact the same program! Modify SequenceSearch *to read in the E. coli sequence (you may have already done this in Step 2). Make up some 11-bp sequences and have the program search for them. Count how many it finds. Make up some 7-bp sequences and have the program search for them. Count how many it finds. If counting is too difficult, then notice that* SequenceSearch *puts all exact matches in an array. How can you find out how many elements are contained in that array?*

D. For each word match found by Blast, how many cells does it have to calculate while attempting to extend the match forwards and backwards? This is difficult to estimate, but take as typical the last figure in the notes for Monday, September 22.

P3.9. How do you explain the fact that *BlastN* cannot find the evident similarity between DG47 and the *lef* gene?