Assignment 2
Due Monday, March 20

Current Topics in Computer Science:
Computational Genomics
CSCI 7000-005, Spring 2006

You are encouraged to work together or use any other resources, but you must tell me who you worked with and what resources you used (even if you didn’t work with anyone), and you must write up the solutions yourself, in your own words/style.

1. Read chapters 4.4-4.9, 5.5, 9, 11, 12 (material already covered), 8.10-8.15, 10.
2. Read supplementary readings about DNA microarrays on the course website.
3. page 211, Problem 6.2
4. The default for BLAST when finding similar DNA strings to a query string requires an exact seed of length 11. When searching for a query string of length 64 for strings that are 70% conserved (a standardly used comparison), the sensitivity of BLAST is 30% (i.e., it will only report 30% of these conserved sequences). In class (and in chapter 12) we discussed a randomized algorithm that considers projections of a number of $(k,l)$-templates. So, BLAST uses a $(11,11)$-template, or, more generally, for $l > 11$, the BLAST template is $(1 2 3 \ldots 11)$. In fact, even if this algorithm is not run many times, we can expect an improvement. Many other $(11,l)$-templates will do better than the BLAST default. For example, the best such template achieves a sensitivity of 45% on the same queries described above for which BLAST’s sensitivity is 30%. Explain why this is.

5. In DNA sequencing, the goal is to determine the complete nucleotide sequence of a long string of DNA. Current common laboratory methods can only accurately sequence a small number of nucleotides, ~300-1000, from one end of a longer string. It is possible to replicate substrings of a DNA string starting at almost any point as long as you know a small number of nucleotides (~9) to the left of that point (this replication is done using a technology called polymerase chain reaction (PCR), which has had a tremendous impact on experimental molecular biology). These 9 nucleotides define a primer that allows copying of the string starting from that point. We want to be as sure as possible that the copied substrings start at the desired point, so we don’t want this 9-mer to appear anywhere else in the previously sequenced portion of the DNA string, nor should it be part of any known frequently repeated substrings in the genome. In the following problems, you may assume you can build a suffix tree in linear time.
   a. Given a set $\mathcal{R}$ of DNA strings (known repeats), give an algorithm to find a shortest string $S$ that is a substring in none of the strings of $\mathcal{R}$. The algorithm should run in time proportional to the sum of the lengths of the strings in $\mathcal{R}$.
   b. Given a string $S$ of the known sequence so far and a set $\mathcal{R}$ of DNA strings, give an algorithm to find the furthest right substring of length 9 in $S$ that does not appear twice in $S$ and is not a substring of any string in $\mathcal{R}$. What is the run-time of your algorithm?
6. Table 1 (below) contains a set of 10 patterns representing a motif of length $l=8$ (motif is highlighted in each).
   a. Construct a profile.
   b. Construct a consensus sequence.
   c. Compute the consensus score of your profile.
   d. Compute the entropy score of your profile.
   e. Assuming a uniform nucleotide distribution in a genome of length $10^6$, how many times would you expect to find the consensus sequence with:
      i. no mismatches?
      ii. up to 1 mismatch?
      iii. up to $k$ mismatches?

   Table 1
<table>
<thead>
<tr>
<th></th>
<th>TCAAGCAAGGC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>AAAAGCAGGGTC</td>
</tr>
<tr>
<td>3</td>
<td>GAACTCAGGGTG</td>
</tr>
<tr>
<td>4</td>
<td>CTAAATCTAGGGC</td>
</tr>
<tr>
<td>5</td>
<td>AAGTATGGACTC</td>
</tr>
<tr>
<td>6</td>
<td>ACTAAGCAGGGT</td>
</tr>
<tr>
<td>7</td>
<td>TCTCAGGGCCCA</td>
</tr>
<tr>
<td>8</td>
<td>CCTGTGGTGAGG</td>
</tr>
<tr>
<td>9</td>
<td>TACCCGTATGGTT</td>
</tr>
<tr>
<td>10</td>
<td>ACCACTCGTCGA</td>
</tr>
</tbody>
</table>

7. page 224, Problem 6.57
8. page 407, Problem 11.3
9. Give the algorithm to traceback the Nussinov algorithm. Explain each step. It is OK to look up the algorithm, but explain it in your own words.
10. Get the file trna_phe_delabeled.txt from the course website. This file has an alignment of tRNA-Phe (for Phenylalanine) sequences.
    **Note:** in the following exercises, it is important to choose the sequences randomly rather than just taking consecutive lines from the alignment, because if your sequences are all identical or are biased towards a particular taxonomic group you may get the wrong structure.
    c. Choose 5 sequences at random from the alignment and run them through the mfold server: [http://www.bioinfo.rpi.edu/applications/mfold old/mta/](http://www.bioinfo.rpi.edu/applications/mfold old/mta/)
    d. Paste the same sequences individually into BayesFold: [http://bayes.colorado.edu/Bayes/](http://bayes.colorado.edu/Bayes/)
       The predictions will be essentially the same on single sequences.
    e. Choose random sets of 2, 3, 4, 5, and 10 aligned sequences. Paste each set into BayesFold. Notice how the predictions get better as you add sequences.