

CS196-1 HW2

Due: Thursday, February 21th 2:30pm

Biologists should complete all parts of problems which do not require programming and should see Sorin for instructions on a final project for the course. A list of grad level final projects will be given in class. Programming can be done in any language but we prefer matlab, C, Java, mathematica, python, scheme, or Haskell. Email electronic handins to cs196-1tas@cs.brown.edu. Include a README that **fully** explains how to run any code.

0 Reading

- *Cartoon Guide to Statistics* (handed out in class)
- *Implication of the Human Genome...* (handed out in class)
- *Designing Life: Proteins 1, Computers 0* (handed out in class)
- *How to Lie with Statistics* Chapter 2
- *An Introduction to Bioinformatics Algorithms* Chapter 1

1 Repeated Substrings (30)

Local alignment between two different strings finds pairs of substrings from the two strings that have high similarity (for a given similarity matrix). It is also important to find substrings of a single string X that have high similarity with X . We will define this type of match as *inexact*, meaning that the strings have at most one mismatch.

Complete all of the following:

- Write a program that will find all inexact substrings (of length at least 3) for a given string. Your program should accept the query string as a command line parameter, and print the result to standard out. You're free to accept other parameters from the commandline, but please document all additional features.
- List all of the inexact repeated substrings (of length at least 2) for the following string:

$$X = A A T T C A A T$$

2 Problem 2 (15)

- Answer the five following questions (which are derived from the NCBI tutorial and linked to from our website under “Resources”).
 1. Briefly explain the mechanism for transcription factor mediated gene regulation.
 2. What is the wobble hypothesis? Why is it important in the context of translation?
 3. Briefly describe the four levels of protein structure.
 4. What is the Central Dogma of Biology?

3 Problem 3 (15)

1. Determine the big-O complexity of the following algorithm in terms of N and M .

```
for i := 0 to N do
  A[i] := 1

for j := 0 to M do
  i := N
  while i > 0 do
    A[j] := A[j] + i
    i := floor(i / 2)
  done
done
```

2. What is the value of $A[M]$ in the above code after it is run with N as 5 and M as 6?

4 Global versus Local (10)

Provide one example of a *biological* problem for which you should use global alignment, and one example of a *biological* problem for which you should use local alignment and explain why.

5 Dot Plots (30)

Dot plots provide a simple but useful means of visualizing similarities between two sequences. To draw a dot plot between two strings, $a = (a_1a_2 \cdots a_n)$ and $b = (b_1b_2 \cdots b_m)$, one uses a matrix with n rows and m columns. The a_i character indexes the i -th row and the b_j character indexes the j -th column. In its simplest form, the entry (i, j) in this Dot Plot matrix is 1 if $a_i = b_j$, and blank otherwise. By displaying the matrix one obtains the Dot Plot of the two strings.

Complete all of the following:

1. Write a program that will take two input strings and generate their Dot Plot.
2. Sketch a dot plot which illustrates each of the following:
 - repeats
 - indels
 - exons/introns
 - SNPs

6 Extra Credit (10)

Genomic sequences could be extremely long. For example, we may want to see the Dot Plot of two different assembly sequences of an entire chromosome. Even with two DNA sequences of sizes around 100,000, a full Dot Plot for them will require an enormous amount of memory. Summarizing Dot Plots by compressing their information reveals the biologically meaningful “big picture,” which is very important and a difficult computational task. Describe and implement an algorithm for compressing the Dot Plot of two DNA sequences which retains the information about where the sequences match and do not match. Assume that matches of less than 5 on either diagonal are not significant.