# CS1820 HW2

*Due: Thursday, February 13th 11:59pm*

Biologists should complete all parts of problems which do not require programming (that includes CS theory!). Final projects for Biology students, graduate credit, or anyone who feels up to the challenge will be discussed in a future class. Programming can be done in any language but we prefer Java, Mathematica, python, matlab, C/C++. Use the CS hand-in script cs182_handin to submit. Include a README that **fully** explains how to run any code.

## 0   Reading

The links to the reads are on the homework page.

- *Designing Life: Proteins 1, Computers 0*

- *NCBI's Molecular Biology Review*

- *Theoretical Biology in the Third Millenium*, Sydney Brenner.

## 1   Problem 1 (10)

- Answer the four following questions (which are derived from NCBI's Molecular Biology Review and linked to from our website under "Resources" or "Homework").

  1. Briefly explain the mechanism for transcription factor mediated gene regulation.
  2. What is the wobble hypothesis?
  3. Briefly describe the four levels of protein structure.
  4. What is the Central Dogma of Biology?

## 2   Problem 2 (10)

In making similarity matrices for specific purposes, one may use the three building blocks (match score, mismatch score, and gap score) in different ways to accomplish the same goal. Give two similarity matrices neither of which allow any gaps and both of which prioritize matches over mismatches (but which differ in more than the magnitude of the values). State the different side effects of each.

# 3   Problem 3 (15) CS Theory

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? Assume that all values of the array A are initialized to 0 before running the program.

# 4   Problem 4 Inexact Repeated Substrings (20)

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 the type of match as *inexact* when the substrings have at most one mismatch.

Complete all of the following:

- *Warmup:* Write a program that will find all exact (no mismatches) repeated substrings.

- Write a program that will find all inexact repeated 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. Optimize your algorithm if you wish, but make sure it is correct!

- List all of the inexact repeated substrings (of length at least 2) for the following string:

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

# 5   Problem 5 Megaman Revisited: Needleman vs Waterman (10)

Professor Farnsworth has sequenced two capsid proteins, "HRV1A" from human and "HRV16" from the chimpanzee. Suppose that he first runs the Needleman-Wunsch algorithm on the two sequences, then next the Smith-Waterman algorithm. What is the *biological* significance of the result Professor Farnsworth receives from Needleman-Wunsch? What is the *biological* significance of the result he receives from Smith-Waterman? Compare and contrast.

For those interested, we have posted the HRV sequences online. They are in FASTA format, in which the line starting with `>` is a descriptor, and all 50-column lines following that descriptor is the associated sequence.

# 6   Problem 6 Dot Plots (25)

*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. Test it on the "HRV1A" and "HRV16" sequences and attach a screenshot. (Your program should work with strings of DNA, RNA, and proteins!)

2. Sketch (on paper, or digitally) a dot plot which illustrates each of the following:

   - repeats
   - indels
   - exons/introns
   - SNPs

# 7   Problem 7 Edit Graphs (10)

Construct the edit graph for sequences X=AGGT and Y=AGT. Using the scoring matrix below and a gap penalty of 0, construct **all** optimal global alignments of the two sequences.

|   | A | C | G | T |
|---|---|---|---|---|
| A | 1 | 0 | 0 | 0 |
| C | 0 | 1 | 0 | 0 |
| G | 0 | 0 | 1 | 0 |
| T | 0 | 0 | 0 | 1 |

## 8   Extra Credit (20pts and pastiche pie?!?)

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 100,000bp, 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. Your algorithm should be lossy. Implementations in Mathematica and/or the implementation which compresses the saved dot plot to the *smallest serialized format* (we are talking bytes on disk) earn special consideration for the pastiche pie.

The sequences for the extra credit are two 1Mb segments of genome with high conservation between human and mouse.