Homework 1

CS 1810, Fall 2024

Out: Sept. 14
Due: Sept 27th, 11:59 PM EST

Please upload your solutions on Gradescope. You can use LaTeX or a word document to write up your answers, but we prefer you use LaTeX. You may scan hand-written work or images for parts of solutions **only if** they are extremely clean and legible. Please ensure that your name does not appear anywhere in your handin.

Problem 1: Global alignment table

It is a warm summer night at Sorin's Farm, and the cows are hungry! The Brahman breed is especially hungry and is excited to munch on barley for dinner. Inspired by this tranquil scene, you (a computational biologist) decide to perform a global alignment on `BRAHMAN` and `BARLEY`!

Create and fill out the global alignment dynamic programming table to align `BRAHMAN` and `BARLEY`. For scoring, use the match bonus $+1$, mismatch penalty -1, and indel penalty -1. When there is a tie between a mismatch and an indel alignment, use a back pointer that indicates a mismatch. Make sure to include back pointers for all cells in the table. If there is a tie between indels, you can include both arrows. What is the score of the optimal alignment and to which alignment does this score correspond?

Problem 2: Multiple alignment

In class, you have learned about the global alignment of two strings. A natural question to ask is how we might go about aligning three strings. Your task is to design an efficient algorithm that solves the following problem.

> GLOBAL ALIGNMENT OF THREE STRINGS
>
> **Input:** Strings u, v, and w and a scoring function δ that takes in three arguments.
>
> **Output:** An alignment of u, v, and w for which the score is maximal (as defined by δ) among all alignments of u, v, and w.

Whereas we had to work fairly hard to build up the original binary global alignment algorithm from scratch, it turns out that we can obtain an algorithm for the global alignment of three strings by making some careful modifications to the algorithm you've seen in class.

In your solution, you *must*

- describe how the structure of the dynamic programming table changes in the extension to three strings,
- give a new recurrence relation for finding the score of an optimal alignment of a given combination of prefixes of u, v, and w,
- specify the initialization and order in which the new dynamic programming table should be filled out.

If you wish, you may also include diagrams, pseudocode, mathematical expressions, and plain English text in your answer. However, please do not submit a block of code with no accompanying explanation.

Once we know how to extend our global alignment algorithm to three strings, we might consider making the general extension to k strings:

- Show that if we continue to insist on constructing dynamic programming tables, the runtime of our algorithm will be $O((2n)^k)$ for k sequences of length n. Since this approach is exponential in k, we will quickly hit a computational wall in attempting to align sequences of even modest length.

  Hint: How many cells are in the new dynamic programming table and how many operations are in the new recurrence relation?

Bonus: Sketch in a few sentences a reasonable heuristic we might use to skirt this problem if we would still like to align multiple sequences.

Problem 3: Counting global alignments

When implementing algorithms, there are different approaches to solving a problem. Brute force programming is a direct approach to solving a task that depends more on computer processing power and memory, while dynamic programming optimizes by breaking a task into smaller problems and drawing upon solutions to the smaller problems in order to solve the overall task.

A brute force approach to global alignment would iterate through every possible alignment of the two input strings, computing the score and storing the max along the way. To understand the performance of such an approach, we need to know the number of possible alignments between two strings, one of length m and the other of length n. Finding an exact expression for the number of possible alignments is actually fairly difficult. In this problem, we simply ask you to determine whether the number of alignments is

1. polynomial in m and n
2. logarithmic in m and n
3. exponential in m and n

Indicate your answer choice by number and justify your response.

Problem 4: A Fowl Virus

One of the pride and joys of Sorin's Farm is his chickens! One day, while tending to the chicken coop, you have a disturbing realization: the chickens have started dying at alarming rates! Being the seasoned computational biologist you are, you realize that there is a viral epidemic ravaging the chicken coops. Luckily, you have a sequencing center on the farm property. You manage to isolate some of the virus and sequence it. Now, with the sequence in your flash drive, it is up to you to save the chickens!

Here, you'll find the sequence for the terrible virus plaguing the fowl. You know that the virus is a retrovirus and is inserting genes into the poor hens' DNA. It's up to you to find out what gene or genes are causing this farmhouse mayhem. Fortunately, you know about BLAST (Basic Local Alignment Search Tool). Navigate to the BLAST website. You're only interested in your chicken's genome so type "chickens" into the species search box and click "search." Now, to find out what gene the virus is infecting your chickens with, copy and paste the viral genome into the box asking for a FASTA sequence and click "BLAST." Once NCBI returns your query, investigate the alignments it returns and try to get to the bottom of this plague.

a. Figure out what gene your chickens are being infected with and what disease is killing them as a result. Specifically, make sure you check out the alignment for chromosome 20! Be sure to provide justification for your answers. The virus itself has an interesting history; if you're interested, try to figure out what virus it is exactly.

Hint: Pay attention to the **features** section and look up specific words to figure out this mysterious disease.

Bonus: The general structure of a retrovirus genome is composed of coding regions for the gag-pol-env polypeptides (NOT proteins) and other proteins that assemble the polypeptides into proteins. Using this information, find a closely related virus for the virus above. Give a short description of how you found the related protein.

Problem 5: DNA Sequencing

DNA sequencing has become indispensable to biological research, enabling us to inspect genomes like those you'll see in this class. With constant innovation, this technology has become more readily available, cheaper, and quicker. There are various companies that provide sequencing technology- arguably the most prevalent is Illumina:

> "Illumina is a leading developer, manufacturer, and marketer of life science tools and integrated systems for large-scale analysis of genetic variation and function. These systems are enabling studies that were not even imaginable just a few years ago, and moving us closer to the realization of personalized medicine."

Read Illumina's beginner's guide to next-generation sequencing.

a. In four or less sentences, explain the process of next-generation sequencing.

b. Describe an application of how DNA sequencing can be used in computational biology research. Why would sequencing be necessary? Please keep your answer to four or fewer sentences. Possible areas of focus include:

   - Tumor profiling
   - GENSCAN
   - De novo genome assembly

Problem 6: MSA Transformer

Earlier in this homework, you designed an algorithm that could globally align three strings. This type of alignment is referred to as multiple sequence alignment (MSA) and is still a widely research problem in computational biology. To learn more about this field, we will explore a type of MSA called MSA Transformer.

Read the Introduction and Related Work sections of this article. (Here is an optional resource if you would like more background information on machine learning and deep learning; the sections up until Half-way Bonus: Valuable Resources provides helpful context.)

a. In 2-3 sentences, why does this model use multiple reference sequences instead of a single reference sequence?

b. In 2-3 sentences, compare and contrast unsupervised and supervised protein language models.

c. MSA Transformer is only one application of multiple sequence alignment. What is another area in computational biology in which MSA may be preferred over standard global alignment and why? Please keep your answer to four or fewer sentences. Possible areas of focus include:

   - Remote homology detection
   - Protein design
   - Pathogenicity prediction