02510/02710 - Midterm 1 - Spring 2024

Name: _____ Andrew ID: _____

Problem 1 [30 pts]. Sequence Alignment

(a) During the lecture, the dynamic programming method for global and local sequence alignment was covered. Please complete the dynamic programming score table for the *global* alignment algorithm of the two sequences `ACCCGA` and `ACTA`. Then state the best alignment and its final score.

The algorithm parameters are:

$$\texttt{gap\_penalty} = -1$$
$$\texttt{mismatch\_penalty} = -2$$
$$\texttt{match\_score} = 2$$

	—	**A**	**C**	**C**	**C**	**G**	**A**
—							
A							
C							
T							
A							

		A	C	C	C	G	A
	0	-1	-2	-3	-4	-5	-6
A	-1	2	1	0	-1	-2	-3
C	-2	1	4	3	2	1	0
T	-3	0	3	2	1	0	-1
A	-4	-1	2	1	0	-1	2

(b) Imagine you have two genome sequences, each with $100M$ base pairs, that require alignment. Would you still employ the approach described in the previous question? If so, explain your reasoning. If not, suggest an adaptation to the algorithm to make it suitable for this larger problem when you are given the promise that the two sequences are similar.

Employing *global* alignment algorithms on extensive genomic sequences is impractical due to its $O(|S|^2)$ time and space complexity (assuming sequences are similar in size). Restricting the search to a diagonal strip (i.e., excluding "traces" beyond a fixed diagonal strip) reduces the algorithm's complexity to $O(w|S|)$, where w represents the strip's "width," and $w << |S|$.

(c) Suppose you are searching for a genomic substring that matches a specific reference sequence, with the substring being much shorter than the initial string. Considering algorithms for *local* and *global* alignment, suggest a dynamic programming algorithm to identify the best substring match using arbitrary match, mismatch, and gap scores. Write out the update rule and initialization for the DP matrix, in addition to any further steps needed to get the alignment from the matrix.

Solution. Let S be the longer sequence and s be the shorter sequence. Define T as the dynamic programming matrix with the following elements:

$$\begin{aligned} T[0,j] &= 0, \quad j \in [0, |S|] \\ T[i,0] &= -\texttt{gap\_score} \times i, \quad i \in [1, |s|] \\ T[i,j] &= \max(T[i,j] + \texttt{match/mismatch}, \\ &\qquad T[i,j-1] + \texttt{gap\_penalty}, \\ &\qquad T[i-1,j] + \texttt{gap\_penalty}) \end{aligned}$$

The backtracing algorithm is as follows: 1) Find the largest $T[|s|, j]$ for all j and backtrace it until reaching $T[0,k]$ for some k.

(d) Let's modify the setup of part (c). Suppose you are searching for a genomic substring of length 100 that matches a random reference sequence of length 100M, and you know for certain that the best alignment has at most 3 single-character edits. Suggest an algorithm to efficiently map that substring to the location of the best alignment in the reference; you may preprocess the reference.

Solution. Use an FM-index or hash table to store the location of all 32-mers in the reference. Since there are at most 3 single-character edits, you are guaranteed that there will be at least one matching 32-mer. Since the reference sequence is random, in expectation, there will only be one matching 32-mer. Thus, you can find the location of the matching 32-mer in constant time. After that, you only need to do a DP extension around that one location.

Problem 2 [20 pts]. Suffix tree vs Suffix trie.

(a) Draw a *suffix trie* and a *suffix tree* for the sequence `GATACA`. We will accept responses for both "GATACA" and "GATACA$" since we did not specify on the end-of-string delimiter.

(b) What is the primary limitation of employing a suffix trie for genomic data in contrast to a suffix tree?

Solution. The primary limitation of employing a suffix trie for genomic data, in contrast to a suffix tree, is its space complexity. Suffix tries can be very large, especially for long genomic sequences, as they store all suffixes explicitly. This can result in significant memory usage, which may become prohibitive for very large genomes. In contrast, suffix trees compress the trie structure by merging nodes with a single child, reducing the overall space required to store the suffixes. This makes suffix trees more efficient in terms of space utilization compared to suffix tries for genomic data.

(c) If you have access to a *suffix tree* data structure, how would you determine if a specific substring exists in the original sequence? Propose an algorithm to solve this problem using the suffix tree. What is the query complexity, and what is the overall space complexity, including the suffix tree data structure?

Solution. To determine if a specific substring exists in the original sequence using a suffix tree, you can perform a depth-first search (DFS) traversal of the suffix tree starting from the root. Start at the root of the suffix tree. For each character in the substring, check if there is an edge labeled with that character from the current node. 1) If there is, follow that edge to the next node. 2) If there isn't, the substring does not exist in the original sequence. 3) If you reach the end of the substring and are at a node in the suffix tree, the substring exists in the original sequence.
Since suffix tree has linear space complexity, the overall space and time complexity is linear.

Problem 3 [30 pts]. Clustering/Classification

Inspired by the paper "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," you aim to develop a model for cancer classification based on gene expression data. You have obtained gene expression profiles from 38 bone marrow patient samples, including 27 samples of Acute Lymphoblastic Leukemia (ALL) and 11 samples of Acute Myeloid Leukemia (AML).

(a) The referenced paper used accuracy as the evaluation metric for the classifier. However, you have recently been aware of an alternative metric for binary classification: the Matthews Correlation Coefficient (MCC). The formula of MCC is:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Consider which metric would be more appropriate for your problem (MCC vs. accuracy) and explain why.

MCC would be more appropriate. Accuracy tends to overoptimistic estimate the classifier's ability on the majority class. In contrast, MCC offers a more balanced assessment by considering all aspects of the confusion matrix. More specifically, MCC produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (TP, FP, TN, FN), proportionally both to the size of positive elements and the size of negative elements in the dataset. Given the imbalance between ALL (27 samples) and AML (11 samples) in the dataset, MCC is more appropriate here.

(b) You have successfully developed a machine-learning model with high accuracy for classifying samples into two known cancer types, ALL and AML. Now you collect some new samples and deploy your model on these samples for prediction. However, after collecting some new samples and deploying your model for prediction, you discover that some of these samples do not fit well into the ALL/AML categories. How would you deal with this scenario, where your model originally trained on ALL and AML data, now encounters samples that likely represent a novel cancer subtype? Example answer: Cluster the training data from existing data types (AML and ALL), then measure the distance between the new sample and the existing clusters. If the distance is far, it's most likely from its new cluster/subtype. A semi-supervised method could be used to retrain the model using a combination of the labeled data (AML and ALL) and the newly identified samples with tentative labels (as the potential new subtype).

(c) Beyond class prediction, you are also interested in class discovery. For example, could the distinction between AML and ALL have been identified solely based on gene expression, even if it were not previously known? To explore this, you plan to use the K-means clustering algorithm to cluster tumors based on gene expression. Given that improper initialization might impede convergence, how would you address this issue?

Accepted answers:
1. multiple random start
2. K-means++ initialization method
3. Using domain knowledge
Partial credit will be given if the answer uses the Elbow Method for optimal K selection

Problem 4 [20 pts]. Multiple Hypothesis Testing

(a) Suppose we have 1000 genes with their differential p-values, using Bonferroni Correction, we found 100 genes with a corrected p-value ≤ 0.05. Calculate the FDR (in %).
The actual p-value would be $0.05/1000 = 0.00005$, and we expect to find $0.00005 * 1000 = 0.05$ genes by chance. Actually we find 100 genes, so the FDR is $0.05/100 \times 100\% = 0.05\%$

(b) Suppose we have N genes with their differential p-values, and a subset of significant M genes with an uncorrected differential p-value < 0.0005. The Bonferroni corrected p-values in the M genes is at most 0.05, and the FDR for the M genes is 0.05%. Compute the value for N and M.
Given that the corrected p-value is 0.05 and the uncorrected p-value is 0.0005, then $N = \frac{0.0005}{0.05} = 100$.
$M = 0.0005 * N/FDR = 100$

Problem 5 [15pts]. Bonus

Soil microbiomes are filled with a huge variety of bacteria, many species of which are unculturable in the lab, but which may have an impact on how well crops grow. We are still able to do shotgun (meta)genomic sequencing, which generates random reads (substrings) from the bacterial genomes present, but assembling or mapping them into genomes is very difficult.

(a) The similarity between soil samples can be measured directly from the mixed reads, even without assembly or mapping to individual genomes. Design an efficient streaming algorithm for doing so that only needs to look at each sample's reads once.

They can use a strategy similar to Mash. Use MinHash to keep a small sample of all k-mers in each soil sample. Then, the Jaccard index is a similarity metric between the soil samples.

(b) The local farmers tell you that based on their experience in the field, there are ten different types of soil. You therefore hypothesize that your soil microbiome samples can be categorized into ten classes based on bacterial composition (i.e. which bacteria are present). Using your similarity from part (a), propose a method for dividing the samples into ten classes.

After having a similarity metric, they can then use either hierarchical clustering or graph based clustering, stopping at exactly 10 clusters.

(c) After you've divided the soil samples into ten classes, the farmers come back to you and say that your classes aren't right. They helpfully provide you with their labelling of the soil samples into their ten classes. How should you measure how accurate your class assignments were compared to their ground truth labels?

Accuracy here is a bit more complicated because there are 10 classes in the confusion matrix instead of only 2, so you must generalize it to multi-class accuracy. The formula is simply the sum of the diagonal entries of the confusion matrix divided by the total number of samples.