ECE 598OM: Homework 1 - Due end of February 2023

Issued by: Olgica Milenkovic

Problem 1 (Computational): Familiarize yourself with two out of three software platforms for RNA folding that uses dynamic programming and deep neural networks to solve the problem.

(a) NN and thermodynamics: https://www.nature.com/articles/s41467-021-21194-4

(b) Thermodynamics: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-129

(c) Thermodynamics: http://rna.tbi.univie.ac.at/cgi-bin/RNAWebSuite/RNAfold.cgi

Problem 2 (Computational): Implement Nussinov's algorithm – you may use the pseudocodes from the lecture notes. Now, the goal is to see how closely the prediction by Nussinov matches the more accurate methods based on thermodynamics (Problem 1). You may change the award for pairs A, T and G, C - you do not need to use 1 for both.

(a) Analysis 1: Create at least a 100 RNA (or DNA) sequences of length 20 and 100 each (for a total of 200 sequences) and record the maximal number of paired bases produced by Nussinov. Then, compare the folds predicted by the algorithms in Problem 1. Count the number of pairings obtained by the thermodynamics algorithm and try to fit a "transfer" function for the two counts. For example, you can use an estimator for the thermodynamics count based on the Nussinov count or vice versa.

(b) Analysis 2: Repeat the same procedure as above but only for strings predicted by Nussinov's method to have less than 20% of paired bases. You may need to increase the number of examples from 100 to something higher in order to get a better estimate.

(c) Analysis 3 Repeat the same procedure as in 1) and 2) upon introducing errors in the generated strings. You want to determine how big of a change in the fold you get when you "mutate" a randomly selected number of positions. For example, if you have a string of length 20, you can change up to ℓ symbols, and you can try out values for ℓ up to say 5% of the length.

Problem 3 (Mathematical): This is a collection of combinatorial problems.

(a) Catalan numbers: Prove the "convolution" recursion for the Catalan numbers given in class and use the recursion to find the generating function. You are allowed to look up the proof from a math text but I strongly suggest you try this on your own first.

(b) Dyck paths: Write a report on the paper addressing the topic of restricted Dyck paths with limited runs of up and down symbols:

http://math.colgate.edu/~integers/v69/v69.pdf

(b) Motzkin lattice paths: Find a formula for Motzkin paths of length n that contain at least $\frac{n}{2} \leq k \leq n$ horizontal steps. For exactly $k = \frac{n}{2}$ horizontal steps, find the number of valid DNA sequences with this fold pattern.

Problem 4 (Mathematical): This is a collection of open-ended problems.

(a) Avoiding long stems: Propose an analytical method or devise an algorithm that can generate large collections (we do not know the capacity of the constraint, so the word "large" is relative) of strings over $\{A, T, G, C\}$ and

length that avoid stems of length $3 \leq k \leq n$. You can try a similar idea as the one used in repeat-removal:

https://ieeexplore.ieee.org/abstract/document/8805120

(b) Motzkin strings at a minimal Hamming distance: Propose a construction for Motzkin string that can correct any substitution error-pattern of ≤ 2 symbols. If you succeed, this will be the first instance of an error-correcting Motzkin code. Another problem we will discuss later in the semester is how to construct deletion-correcting Motzkin paths.