CSC482B Project Proposal: RNA_LZW (title subject to change)

Quinn Gieseke, Claire Champernowne

February 20, 2021

Motivation

Detecting RNA secondary structures is a process which is integral to understanding the functions of non-coding RNA sequences. Though there are existing bioinformatics tools used for detection, these programs, thwy are not without their limitations. Specifically, RNAz 1.0 uses a sliding window approach in its analysis of a sequence [1], meaning there may be structures with loops existing outside the scope of the window.

The Lempel-Ziv-Welch(LZW) algorithm is a universal lossless compression algorithm used for a variety of applications such as Unix file compression. The algorithm compresses data by using a dictionary to assign common sequences of characters to a fixed-length code (typically 12-bit) [2].

Given the LZW algorithm's dictionary approach to pattern-matching, we hypothesize it could be used as a novel method of detecting new RNA secondary structures which are undetectable by the sliding window method used in RNAz 1.0.

Objectives

Given the time limitations on this project, we intend to focus solely on detection of possible structures without commenting on the structure's validity in nature. Our goal is to detect RNA secondary structures which may be impossible to detect using RNAz 1.0 using our novel method of detection, RNA_LZW.

In addition, we wish to achieve detection in a reasonable time frame, which will be assessed by comparing RNA_LZW's processing time to that of other RNA secondary structure detection tools.

Methods

The theoretical RNA_LZW algorithm will need to differ from normal LZW implementations in a few key ways. For one, it is beneficial for compression for one sequence to be matched to a large number of similar sequences, in this way maximally reducing the size of the file. However, for a RNA fold, each sequence can only be paired to at most 1 other sequence. Since RNA_LZW has this restriction, it will require additional overhead in the algorithm to maintain matches and ensure that only the longest sequence matches make it to the final output.

Our approach also will be easier in some ways than the compression task. Because we only have 4 base pairs to work with, instead of a full ASCII alphabet, our symbol table will have much more room for growth, allowing us to process large amounts of base pairs efficiently. Since LZW and similar algorithms like LZMA are capable of processing gigabytes of data efficiently, our new algorithm should be able to handle similarly large datasets, in contrast to algorithms like RNAfold, which do not scale efficiently past several thousand base pairs [3].

References

[1] https://psb.stanford.edu/psb-online/proceedings/psb10/gruber.pdf

[2] https://www2.cs.duke.edu/courses/spring03/cps296.5/papers/welch_1984_technique_for.pdf

[3] https://www.tbi.univie.ac.at/RNA/RNAfold.1.html