PROJECT PROPOSAL

TITLE: ST-PTD (Suffix Tree Partition and Top-Down) algorithm implementation on RNA sequences.

INTRODUCTION:

The proposed title above will experiment with RNA sequences using ST-PTD (Suffix Tree Partition and Top-Down) algorithm which previously studied DNA sequences in this[1] paper. The papers [1] and [2] discussed a common problem in suffix trees: memory greedy. ST-PTD is cache efficient, but it has an $O(n^2)$ worst-case complexity. Despite that, the results demonstrate that suggested technique improves performance in terms of running time.

What's the point of studying RNA?
Whereas DNA serves as the underlying blueprint for all biological functions, RNA is the molecule that is synthesised when those operations are required. The encoded functions are subsequently carried out by proteins translated from messenger RNA. As a result, RNA occupies a distinct space between DNA and protein.

PROBLEM STATEMENT:

Specific goal to study RNA sequences: To be precise , looking at DNA gives us a static image of what a cell or organism might do or become, whereas looking at RNA shows us what a cell or organism is doing right now. None of this implies that RNA sequencing is "better" or more important than DNA sequencing. The truth is that these two processes are interdependent and mutually beneficial [3].

Why implement this on RNA?
Based on this, an algorithm for building a suffix tree for DNA sequences is described, which uses partitioning strategies based on common prefixes to create separate subtrees. The proposed approach is memory-efficient and has a faster average running time, according to the trials. As a result, being able to quickly store and query these sequences can be extremely beneficial. The same problem exists for sequencing RNAs. Thus, expanding the experiment to RNA sequence will demonstrate how efficient algorithm is compared with Kurtz algorithm for RNA sequences.

OBJECTIVES:

The experiment compares running time for ST-PTD and Kurtz algorithm where for DNA sequences. There are same problems for suffix trees while sequencing RNA too. The ST-PTD algorithm has a running time of $O(n^2)$, while Kurtz's approach has a running time of $O(n)$ in the worst case. The partition phase reduces the number of problems it processes in order to deal with bigger amounts of data. After performing the same experiment for RNA , we can compare if the same fact holds true for RNA sequence.

PRELIMINARY LITERATURE REVIEW:

The paper [1] demonstrates with finding that on average, ST-PTD is a little faster than Kurtz's algorithm. This also demonstrates that memory reference locality has a significant impact on algorithm execution time. Algorithm performance is also influenced by partitioning schemes and sequence structure. When they compared the running times of the two algorithms in various setups, they observed that memory is still one of the bottlenecks limiting the algorithms' performance, as the suffix tree is extremely space demanding. Furthermore, as compared to Kurtz's approach, the ST-PTD technique is simpler to comprehend and implement. The ST-PTD algorithm is also easy to parallelize because each sub-creation tree's is independent.

RESEARCH METHODOLOGY:

In the experiments, two algorithms were used: Kurtz's approach and ST-PTD, which were implemented using C programming language and compiled using GCC. Programs were performed on two different platforms to demonstrate the impact of memory on algorithms. Intel Pentium 4.3GHz, 512MB RAM, Red Hat Linux 9 and Intel Pentium III 1.3GHz, 128MB RAM, Fedora 4 are the specific configurations for configl and config2, respectively.

This paper will use Python programming language and GCC compiler in MacBook with Apple M1 chip, 8-core GPU, 8 GB RAM, 256GB Memory and macOS Monterey v12.2.1(latest but might change).

REFERENCES:

[1] H. Huo and V. Stojkovic, "A Suffix Tree Construction Algorithm for DNA Sequences," 2007 IEEE 7th International Symposium on BioInformatics and BioEngineering, 2007, pp. 1178-1182, doi: 10.1109/BIBE.2007.4375711.

[2] Comin, Matteo, and Montse Farreras. “Parallel continuous flow: a parallel suffix tree construction tool for whole genomes.” *Journal of computational biology : a journal of computational molecular cell biology* vol. 21,4 (2014): 330-44. doi:10.1089/cmb.2012.0256

[3] "Why Look at RNA Instead of DNA? | Discovering the Genome", *Discoveringthegenome.org*, 2022. [Online]. Available: https://discoveringthegenome.org/discovering-genome/rna-sequencing/why-look-rna-instead-dna. [Accessed: 07- Mar- 2022]