Suffix Arrays for DNA Pattern Matching

Progress Report

https://github.com/didjeridou/sequences

Guillaume Laliberté

CSCI 51

Prof. Greg Morrisett

Advisor TF Thomas Jiang

April 2015

Progress Report

This I was able make considerable progress both in terms of understanding the critical algorithms for my data structure and in terms of code.

With some research, I was able to find optimal or nearly-optimal algorithms for the important functions implemented by a suffix array. For the performance, one of the most critical function is the one that orders the suffixes. I found a few algorithm that I would like to implement in place of List.sort as they provide a lower complexity.

The first one is the radix sort. Radix sort runs in O(kn) where k is a constant for sufficiently large n and n the number of keys. Apart from the quicksort, another interesting algorithm is the Manber-Meyers repeated doubling algorithm, which was developed to run in O(N log N). I am interested in implementing the latter as an extra feature.

For the search in the suffix array, I decided to move forward with a binary search. It provides a very decent runtime of O(logN). I am also looking into the papers like *Suffix Arrays: A New Method for On-Line String Searches* from Manber and Myers to learn more about how to optimize the suffix array implementation.

On the coding side I was able to successfully implement a first version of suffix arrays. This week I will finish the bridge to DNA sequence, clean up the code and work on extensions.

So far the biggest challenge was to find consistent information about good implementations of suffix arrays. They are not an obscure structure, but there is not as much information out there as suffix trees or other more common structures.

As far as teamwork goes, this is not an issue because this is a solo project. I use my project schedule to make some progress every day.

Here are the next big steps for the project:

- DNASequence using SuffixArray
- Polish the code and finish writing unit tests
- Implement the command line utility
- Test the program with the command line arguments
- Implement interesting extensions or more performant algorithms