AARHUS UNIVERSITY

BIOINFORMATICS RESEARCH CENTER

Range Minimum Queries and Interval-Trees for Genome Scale String Processing

Pediotidis-Maniatis Dimitrios
202002459

Supervisor:
Thomas Mailund

Master's Thesis in Bioinformatics

June, 2022

Acknowledgements

I would like to thank my supervisor Thomas Mailund for his guidance, patience and support throughout this thesis.

Next, I would like to thank my colleagues for the amazing environment they provided. Sharing this experience with them everyday is something I will treasure.

And of course, I would like to thank my friends and family back home. Their unwavering support and encouragement under all circumstances are what eventually made this work a reality.

Abstract

Suffix Trees have been considered one of the most important data structure in string analysis, since their conception in 1973. Nevertheless, their large memory requirements have pushed for the conception of more succinct data structures. This thesis analyzes the concepts of LCP-Interval trees and Range Minimum Queries as an alternative to suffix trees. More specifically we are concerned with the simulation of a top-down traversal of the suffix tree to perform exact pattern matching. In our approach, we first create the necessary framework. We analyze the construction of the suffix and lcp arrays using the SA-IS and Kasai algorithms respectively. Then, we proceed to explore the LCP-interval trees and different implementations to the Range Minimum Query problem. We end the analysis of Range Minimum Queries with the Fischer-Heun structure, which achieves constant time queries with linear preprocessing of the lcp array. In our experiments we aimed to verify the relationship between the Fischer-Heun structure for Range Minimum Queries and two other implementations. Namely, the sparse table and a hybrid solution which allows queries in logarithmic time. Lastly, we compare two exact pattern matching implementations. The first makes use of the Range Minimum Queries and the LCP-interval tree. The second makes use of the suffix tree. These were implemented in Python programming language.

Contents