TRELLIS: GENOME-SCALE DISK-BASED SUFFIX TREE INDEXING ALGORITHM

By

Benjarath Phoophakdee

An Abstract of a Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Computer Science

The original of the complete thesis is on file
in the Rensselaer Polytechnic Institute Library

Examining Committee:

Prof. Mohammed J. Zaki, Thesis Adviser
Prof. Christopher Bystroff, Member
Prof. Lee Newberg, Member
Prof. David Spooner, Member
Prof. Bülent Yener, Member

Rensselaer Polytechnic Institute
Troy, New York

July 2007
(For Graduation August 2007)

ABSTRACT

With the exponential growth of biological sequence databases, it has become critical to develop effective techniques for storing, querying, and analyzing these massive data. Suffix trees are widely used to solve many sequence-based problems, and they can be built in linear time and space, provided the resulting tree fits in main-memory. To index larger sequences, several external suffix tree algorithms have been proposed in recent years. However, they suffer from several problems such as susceptibility to data skew, non-scalability to genome-scale sequences, and non-existence of suffix links, which are crucial in various suffix tree based algorithms.

In this thesis, we propose a novel disk-based suffix tree algorithm for indexing DNA sequences called Trellis. Our algorithm does not suffer from any of the above drawbacks, effectively scales up to genome-scale sequences, and is also able to quickly rebuild suffix links. Specifically it can index, on a typical modern computer, the entire human genome using 2GB of memory in about 4 hours and can recover all the suffix links within an additional 2 hours. Despite the success of Trellis, the algorithms main limitation is that it requires the entire input sequence to be kept in memory. To handle larger DNA sequences, we introduce a novel string buffering strategy that allows our algorithm to assign a very small amount of memory for the input string. As a result, Trellis is able to index very large sequences in a reasonable amount of time and memory. The buffer strategy also speeds up Trellis when the input string barely fits in memory. Trellis was compared to various state-of-the-art persistent disk-based suffix tree construction algorithms, and was shown to outperform the best previous methods, both in terms of indexing time and querying time.