HAL
archives-ouvertes.fr

Building the Assembly De Bruijn Graph from an Implicit Suffix Tree

Eric Rivals

▶ To cite this version:

Eric Rivals. Building the Assembly De Bruijn Graph from an Implicit Suffix Tree. Jens Stoye and Roland Wittler. Mini-Workshop on the Storage, Search and Annotation of Multiple Similar Genomes, Dec 2013, Bielefeld, Germany. 2013, <http://wiki.techfak.uni-bielefeld.de/didy/workshopPangenomeStorageSearch>. <lirmm-00938959>

HAL Id: lirmm-00938959
https://hal-lirmm.ccsd.cnrs.fr/lirmm-00938959

Submitted on 29 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Building the Assembly De Bruijn Graph from an Implicit Suffix Tree

Eric Rivals
LIRMM, CNRS - Univ. Montpellier 2, France
Institut de Biologie Computationnelle, Montpellier, France

4th November 2013

Work in collaboration with Bastien Cazaux and Thierry Lecroq.

Suffix trees belong to the most studied indexing data structures for strings. Generalised suffix trees can index the substrings of a set of input words. In its construction algorithm Ukkonen used implicit suffix trees, which relax the assumption that a suffix is not a prefix of another suffix, and thus represent some suffixes by internal nodes rather than by leaves. In computational biology, for the sake of genome assembly, one wishes to assemble a target sequence from a multitude of input strings, usually called the reads. Numerous algorithms build assembly related graphs that represent the overlaps of the input reads: for instance, the overlap graph, or a special version of the De Bruijn graph, in which only k-mers occurring within the reads are represented by nodes. We address the question of algorithms for building the overlap graph and the De Bruijn graph directly from indexing data structures of the read set, and investigate it when the index in an implicit generalised suffix tree. We will present the algorithms for the De Bruijn graph and for its contracted version, and show they run in a time that is linear in the size of the index. Generally, this work establishes algorithmic paths to convert classical indexing data structures in graph representations useful for assembly.

Acknowledgments: This work is supported by ANR blanc Projet ANR-12-BS02-0008 Colib'read, Defi MASTODONS SePhHaDe from CNRS.