Efficient Solutions to Biological Problems Using de Bruijn Graphs

Leena Salmela
University of Helsinki, Finland

Abstract

The de Bruijn graph has become a standard method in the analysis of sequencing reads in computational biology due to its ability to represent the information contained in large read sets in small space. A de Bruijn graph represents a set of sequencing reads by its k-mers, i.e. the set of substrings of length k that occur in the reads. In the classical definition, the k-mers are the edges of the graph and the nodes are the $k-1$ bases long prefixes and suffixes of the k-mers. Usually only k-mers occurring several times in the read set are kept to filter out noise in the data. De Bruijn graphs have been used to solve many problems in computational biology including genome assembly [4, 9, 1, 8], sequencing error correction [10, 7, 11, 5], reference free variant calling [13], indexing read sets [6], and so on. Next I will discuss two of these problems in more depth.

The de Bruijn graph first emerged in computation biology in the context of genome assembly [4, 9] where the task is to reconstruct a genome based on sequencing reads. As the de Bruijn graph can represent large read sets compactly, it became the standard approach to assemble short reads [1, 8]. In the theoretical framework of de Bruijn graph based genome assembly, a genome is thought to be the Eulerian path in the de Bruijn graph built on the sequencing reads. In practise, the Eulerian path is not unique and thus not useful in the biological context. Therefore, practical implementations report subpaths that are guaranteed to be part of any Eulerian path and thus part of the actual genome. Such models include unitigs, which are nonbranching paths of the de Bruijn graph, and more involved definitions such as omnitigs [12].

In genome assembly the choice of k is a crucial matter. A small k can result in a tangled graph, whereas a too large k will fragment the graph. Furthermore, a different value of k may be optimal for different parts of the genome. Variable order de Bruijn graphs [3, 2], which represent de Bruijn graphs of all orders k in a single data structure, have been proposed as a solution but no rigorous definition corresponding to unitigs has been presented. We give the first definition of assembled sequences, i.e. contigs, on such graphs and an algorithm for enumerating them.

Another problem that can be solved with de Bruijn graphs is the correction of sequencing errors [10, 7, 11, 5]. Because each position of a genome is sequenced several times, it is possible to correct sequencing errors in reads if we can identify data originating from the same genomic region. A de Bruijn graph can be used to represent compactly the reliable information and the individual reads can be corrected by aligning them to the graph.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Applied computing → Sequencing and genotyping technologies

Keywords and phrases de Bruijn graph, variable order de Bruijn graph, genome assembly, sequencing error correction, k-mers

Digital Object Identifier 10.4230/LIPIcs.WABI.2022.1

Category Invited Talk

Funding Supported by Academy of Finland [grants 308030 and 323233].

References

1 Anton Bankevich, Sergey Nurk, Dmitry Antipov, Alexey A. Gurevich, Mikhail Dvorkin, Alexander S. Kulikov, Valery M. Lesin, Sergey I. Nikolenko, Son Pham, Andrey D. Prjibelski, Alexey V. Pyshkin, Alexander V. Sirotkin, Nikolay Vyahhi, Glenn Tesler, Max A. Alekseyev, and Pavel A. Pevzner. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*, 19(5):455–477, 2012.

2 Djamal Belazzougui, Travis Gagie, Veli Mäkinen, Marco Previtali, and Simon J. Puglisi. Bidirectional variable-order de Bruijn graphs. In *Proceedings of LATIN 2016*, pages 164–178. Springer, 2016.

3 Christina Boucher, Alex Bowe, Travis Gagie, Simon J. Puglisi, and Kunihiko Sadakane. Variable-order de Bruijn graphs. In *Proceedgins of Data Compression Conference 2015*, pages 383–392, 2015.

4 Ramana M. Idury and Michael S. Waterman. A new algorithm for DNA sequence assembly. *Journal of Computational Biology*, 2(2):291–306, 1995.

5 Antoine Limasset, Jean-François Flot, and Pierre Peterlongo. Toward perfect reads: self-correction of short reads via mapping on de Bruijn graphs. *Bioinformatics*, 36(5):1374–1381, 2019.

6 Camille Marchet, Christina Boucher, Simon J. Puglisi, Paul Medvedev, Mikaël Salson, and Rayan Chikhi. Data structures based on k-mers for querying large collections of sequencing data sets. *Genome Research*, 31:1–12, 2021.

7 Giles Miclotte, Mahdi Heydari, Piet Demeester, Stephane Rombauts, Yves Van de Peer, Pieter Audenaert, and Jan Fostier. Jabba: hybrid error correction for long sequencing reads. *Algorithms for Molecular Biology*, 11(10), 2016.

8 Yu Peng, Henry C. M. Leung, S. M. Yiu, and Francis Y. L. Chin. IDBA – A practical iterative de Bruijn graph de novo assembler. In *Proceedings of RECOMB 2010*, volume 6044 of *LNBI*, pages 426–440. Springer, 2010.

9 Pavel A. Pevzner, Haixu Tang, and Michael S. Waterman. An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Acadademy of Sciences*, 98(17):9748–9753, 2001.

10 Leena Salmela and Eric Rivals. LoRDEC: accurate and efficient long read error correction. *Bioinformatics*, 30(24):3506–3514, 2014.

11 Leena Salmela, Riku Walve, Eric Rivals, and Esko Ukkonen. Accurate selfcorrection of errors in long reads using de Bruijn graphs. *Bioinformatics*, 33(6):799–806, 2017. (Also in RECOMB-seq 2016).

12 Alexandru I. Tomescu and Paul Medvedev. Safe and complete contig assembly through omnitigs. *Journal of Computational Biology*, 24(6):590–602, 2017.

13 Raluca Uricaru, Guillaume Rizk, Vincent Lacroix, Elsa Quillery, Olivier Plantard, Rayan Chikhi, Claire Lemaitre, and Pierre Peterlongo. Reference-free detection of isolated SNPs. *Nucleic Acids Research*, 43(2):e11, 2015.