BIOINFORMATICS

Vol. 00 no. 00 2016
Pages 1–2

Succinct Colored de Bruijn Graphs

Martin D. Muggli [1,*], Alexander Bowe [2,*], Travis Gagie [3], Robert Raymond [1], Noelle R. Noyes [4], Paul Morley [4], Keith Belk [5], Simon J. Puglisi [3], and Christina Boucher [1]

*These authors contributed equally to this work.
[1]Department of Computer Science,Colorado State University, Fort Collins, CO
[2]National Institute of Informatics, Chiyoda-ku, Tokyo, Japan
[3]Department of Computer Science,University of Helsinki, Finland
[4]Department of Clinical Sciences, Colorado State University, Fort Collins, CO
[5]Department of Animal Sciences, Colorado State University, Fort Collins, CO

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: Iqbal et al. (Nature Genetics, 2012) introduced the *colored de Bruijn graph*, a variant of the classic de Bruijn graph, which is aimed at "detecting and genotyping simple and complex genetic variants in an individual or population". Because they are intended to be applied to massive population level data, it is essential that the graphs be represented efficiently. Unfortunately, current succinct de Bruijn graph representations are not directly applicable to the colored de Bruijn graph, which require additional information to be succinctly encoded as well as support for non-standard traversal operations.

Results: Our data structure dramatically reduces the amount of memory required to store and use the colored de Bruijn graph, with some penalty to runtime, allowing it to be applied in much larger and more ambitious sequence projects than was previously possible. In particular, we use our method along with a custom curated database of antimicrobial resistant genes to track changes in the resistome across food production facilities. A short video of our work is available at `http://cdbg.martindmuggli.com`.

1 INTRODUCTION

In the 20 years since it was introduced to bioinformatics by Idury and Waterman [11], the *de Bruijn graph* has become a mainstay of modern genomics, essential to genome assembly [6, 21, 17]. The near ubiquity of de Bruijn graphs has led to a number of succinct representations, which aim to implement the graph in small space, while still supporting fast navigation operations. Formally, a de Bruijn graph constructed for a set of strings (e.g., sequence reads) has a distinct vertex v for every unique $(k-1)$-mer (substring of length $k-1$) present in the strings, and a directed edge (u, v) for every observed k-mer in the strings with $(k-1)$-mer prefix u and $(k-1)$-mer suffix v. A contig corresponds to a non-branching path through this graph. See Compeau et al. [6] for a more thorough explanation of de Bruijn graphs and their use in assembly.

In 2012, Iqbal et al. [12] introduced the *colored de Bruijn graph*, a variant of the classical structure, which is aimed at "detecting and genotyping simple and complex genetic variants in an individual or population." The edge structure of the colored de Bruijn graph is the same as the classic structure, but now to each vertex ($(k-1)$-mer) and edge (k-mer) is associated a list of colors corresponding to the samples in which the vertex or edge label exists. More specifically, given a set of n samples, there exists a set $\mathcal{C}$ of n colors $c_1, c_2, .., c_n$ where c_i corresponds to sample i and all k-mers and $(k-1)$-mers that are contained in sample i are colored with c_i. A *bubble* in this graph corresponds to a directed cycle, and is shown to be indicative of biological variation by Iqbal et al. [12]. CORTEX, Iqbal et al.'s [12] implementation, uses the colored de Bruijn graph to develop a method of assembling multiple genomes simultaneously, without losing track of the individuals from which $(k-1)$-mers (and k-mers) originated as well as their coverage. This assembly is derived from either multiple reference genomes, multiple samples, or a combination of both.

Variant information of an individual or population can be deduced from structure present in the colored de Bruijn graph and the colors of each k-mer. As implied by Iqbal et al. [12], the ultimate intended use of colored de Bruijn graphs is to apply it to massive, population-level sequence data that is now abundant due to next generation sequencing technology (NGS) and multiplexing. These technologies have enabled production of sequence data for large populations, which has led to ambitious sequencing initiatives that aim to study genetic variation for agriculturally and bio-medically important species. These initiatives include the *Genome 10K* project that aims to sequence the genomes of 10,000 vertebrate species [9], the *iK5* project [8], the 150 Tomato Genome ReSequencing project [3, 13], and the 1001 Arabidopsis project, a worldwide initiative to sequence cultivars of *Arabidopsis* [23]. Given the large number of individuals and sequence data involved in these projects, it is imperative that the colored de Bruijn graph can be stored and traversed in a space- and time-efficient manner.

We develop an efficient data structure for storage and use of the colored de Bruijn graph. Compared to CORTEX, Iqbal et al.'s [12] implementation, our new data structure dramatically reduces the amount of memory required to store and use the colored de Bruijn graph, with some penalty to runtime. In addition to demonstrating the memory and runtime of VARI, we validate its output using the *E.coli* reference genomes and AMR dataset. In particular, our experiment on the AMR dataset validates VARI's ability to correctly identify AMR genes from a metagenomics sample, which is of paramount importance since—when expressed in bacteria—AMR genes render the bacteria resistant to antibiotics and pose serious

risk to public health. Our experiments and results focus on beta-lactamases, which are genes that confer resistance to a class of antibiotics that are considered to be the last resort for infections from multi-drug-resistant bacteria [16, 20]. Our experiments demonstrate that all beta-lactamases were correctly identified and only two of the remaining 47 genes were identified to be in the sample, which had 97% and 95% sequence similarity to one of the beta-lactamases in the sample.

2 RELATED WORK

As noted above, maintenance and navigation of the de Bruijn graph is a space and time bottleneck in genome assembly. Space-efficient representations of de Bruijn graphs have thus been heavily researched in recent years. One of the first approaches was introduced by Simpson et al. [22] as part of the development of the ABySS assembler. Their method stores the graph as a distributed hash table and thus requires 336 GB to store the graph corresponding to a set of reads from a human genome (HapMap: NA18507).

In 2011, Conway and Bromage [7] reduced space requirements by using a sparse succinct bitvector (by Okanohara and Sadakane [18]) to represent the k-mers (the edges), and used the characteristic *rank()* and *select()* operations to traverse it. As a result, their representation took 32 GB for the same data set. Minia, by Chikhi and Rizk [5], uses a Bloom filter to store edges. They traverse the graph by generating all possible outgoing edges at each node and testing their membership in the Bloom filter. Using this approach, the graph was reduced to 5.7 GB on the same dataset. Contemporaneously, Bowe, Onodera, Sadakane and Shibuya [1] developed a different succinct data structure based on the Burrows-Wheeler transform [2] that requires 2.5 GB. The data structure of Bowe et al. [1] is combined with ideas from IDBA-UD [19] in a metagenomics assembler called MEGAHIT [14]. In practice MEGAHIT requires more memory than competing methods but produces significantly better assemblies. Chikhi et al. [4] implemented the de Bruijn graph using an FM-index and *minimizers*. Their method uses 1.5 GB on the same NA18507 data. In 2015, Holley et. al. [10] released the Bloom Filter Trie, which is another succinct data structure for the colored de Bruiin graph; however, we were unable to compare our method against it since it only supports the building and loading of a colored de Bruijn graph and does not contain operations to support our experiments. Lastly, SplitMEM [15] is a related algorithm to create a colored de Bruijn graph from a set of suffix trees representing the other genomes.

3 RESULTS

We demonstrate this reduction in memory through a comprehensive set of bubble calling experiments across the following three datasets: (1) 3,765 *Escherichia coli (E. coli)* genome assemblies downloaded from NCBI, (2) a set of 54 antimicrobial resistance (AMR) genes and a simulated metagenomics sample containing seven of these 54 AMR genes, and four AMR genes not contained in this set, and, (3) four plant genomes. We show our method, which we refer to as VARI (Finnish for color), has better peak memory usage during graph traversal on all these datasets. This observation is highlighted on two datasets: the plant reference genomes, where CORTEX required 101 GB and VARI required 19 GB, and the set of *E. coli* assemblies, which we could not successfully run CORTEX on (est. 3 TB for vertex storage) while VARI completed traversal in 11 hours using only 26 GB. VARI is a novel generalization of the succinct data structure for classical de Bruijn graphs due to Bowe et al. [1], which is based on the Burrows-Wheeler transform of the sequence reads, and thus, has independent theoretical importance.

REFERENCES

[1]A. Bowe, T. Onodera, K. Sadakane, and T. Shibuya. Succinct de Bruijn graphs. In *Proc. WABI*, pages 225–235, 2012.

[2]M. Burrows and D.J. Wheeler. A block sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, 1994.

[3]M. Causse et al. Whole genome resequencing in tomato reveals variation associated with introgression and breeding events. *BMC Genomics*, 14:791, 2013.

[4]R. Chikhi, A. Limasset, S. Jackman, J.T. Simpson, and P. Medvedev. On the representation of de Bruijn graphs. In *Proc. RECOMB*, pages 35–55, 2014.

[5]R. Chikhi and G. Rizk. Space-efficient and exact de Bruijn graph representation based on a Bloom filter. *Algorithms for Molecular Biology*, 8(22), 2012.

[6]P.E. Compeau, P.A. Pevzner, and G. Tesler. How to apply de bruijn graphs to genome assembly. *Nature Biotechnology*, 29:987–991, 2011.

[7]T. C. Conway and A. J. Bromage. Succinct data structures for assembling large genomes. *Bioinformatics*, 27(4):479–486, 2011.

[8]K.J. G.E. Robinson et al. Creating a buzz about insect genomes. *Science*, 331(6023):1386, 2011.

[9]Genome 10K Community of Scientists. Genome 10K: A proposal to obtain whole-genome sequence for 10,000 vertebrate species. *Journal of Heredity*, 100(6):659–674, 2009.

[10]Guillaume Holley, Roland Wittler, and Jens Stoye. Bloom filter trie–a data structure for pan-genome storage. *Algorithms in Bioinformatics*, pages 217–230, 2015.

[11]R.M. Idury and M.S. Waterman. A new algorithm for DNA sequence assembly. *Journal of Computational Biology*, 2:291–306, 1995.

[12]Z. Iqbal, M. Caccamo, I. Turner, P. Flicek, and G. McVean. De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nature Genetics*, 44:226–232, 2012.

[13]M. Kobayashi et al. Genome-wide analysis of intraspecific DNA polymorphism in "micro-tom", a model cultivar of tomato (solanum lycopersicum). *Plant Cell Physiology*, 55(2):445–454, 2014.

[14]D. Li, C.-M. Liu, R. Luo, K. Sadakane, and T.-W. Lam. MEGAHIT: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics*, 31(10):1674–1676, 2015.

[15]Shoshana Marcus, Hayan Lee, and Michael C Schatz. Splitmem: A graphical algorithm for pan-genome analysis with suffix skips. *Bioinformatics*, 30(24):3476–3483, 2014.

[16]Maryn McKenna. Antibiotic resistance: The last resort. *Nature*, 499:394–396, 2013.

[17]M.D. Muggli, S.J. Puglisi, R. Ronen, and C. Boucher. Misassembly detection using paired-end sequence reads and optical mapping data. *Bioinformatics (special issue of ISMB 2015)*, 31(12):i80–i88, 2015.

[18]D. Okanohara and K. Sadakane. Practical entropy-compressed rank/select dictionary. In *Proc. ALENEX*, pages 60–70. SIAM, 2007.

[19]Y. Peng, H. C. Leung, S. M. Yiu, and F. Y. Chin. IDBA-UD: a *de novo* assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*, 28(11):1420–1428, 2012.

[20]Anne Marie Queenan and Karen Bush. Carbapenemases: the versatile beta-lactamases. *Clinical Microbiology Reviews*, 7(3):440–458, 2007.

[21]R. Ronen, C. Boucher, H. Chitsaz, and P. Pevzner. SEQuel: Improving the accuracy of genome assemblies. *Bioinformatics (special issue of ISMB 2012)*, 28(12):i188–i196, 2012.

[22]J.T. Simpson, K. Wong, S.D. Jackman, J.E. Schein, S.J. Jones, and I. Birol. ABySS: a parallel assembler for short read sequence data. *Genome Research*, 19(6):1117–1123, 2009.

[23]D. Weigel and R. Mott. The 1001 genomes project for Arabidopsis thaliana. *Genome Biology*, 10(5):107, 2009.