氏　　名	Alexander BOWE
学位(専攻分野)	博士(情報学)
学 位 記 番 号	総研大甲第 2155 号
学位授与の日付	2020 年 3 月 24 日
学位授与の要件	複合科学研究科　情報学専攻 学位規則第6条第1項該当
学 位 論 文 題 目	Succinct de Bruijn Graphs
論 文 審 査 委 員	主　査　教授　宇野　毅明
	教授　河原林　健一
	助教　岩田　陽一
	教授　定兼　邦彦 東京大学大学院情報理工学系研究科
	准教授　渋谷　哲朗 東京大学　医科学研究所

(Form 3)

Summary of Doctoral Thesis

Name in full Alexander BOWE

Title Succinct de Bruijn Graphs

While consumer-grade genotyping – such as that used by 23andMe – has proven a popular and inexpensive method to determine Single Nucleotide Polymorphisms (SNPs) in individuals, such methods can only detect a set of reference genes, thus limiting their ability to detect all but the simplest variations.

Whole genome sequencing (without a reference) is a powerful alternative, albeit comparatively expensive. However, the price has been steadily declining: while the Human Genome Project cost $2.7 billion to complete in 2003, as of 2019 it is possible to have a genome sequenced for $299, and the price continues to drop.

This decline in price is in large part owed to the advent of Next Generation Sequencing (NGS) machines. The Sanger sequencing method used in the Human Genome project required a high degree of human interaction, which NGS machines have subsequently automated, greatly increasing the speed and decreasing the cost. And although NGS machines produce much shorter reads (200 bases versus 800 bases in Sanger sequencing – a human genome is 3.4 billion bases), this is overcome by re-sequencing the same DNA.

The process of combining short reads into longer sequences is called assembly, and while finding the best overlap is NP-hard, many practical approaches have been proposed.

Traditionally, assembly employed an overlap graph, where each read is a node, and an edge exists if two reads have sufficient overlap. Assembly then involves computing a Hamiltonian tour of all nodes. This was an acceptable drawback when dealing with Sanger reads, but is prohibitively expensive to deal with the abundant data that NGS machines produce.

Eulerian assembly replaces the overlap graph with a de Bruijn graph, where every k-length substring of the reads is a node, and the directed edges are defined by the $k+1$-length substrings that contain the k-length vertices, where k is a user-selected parameter. For example, for $k = 3$, the sequence 'TACGT' yields the edges 'TACG' and 'ACGT', and the edge 'TACG' connects the vertices 'TAC' and 'ACG' by dropping the initial 'T' and appending a 'G'. A complete example is given in Figure 1.

Contigs (contiguous sequences) are then found by following the edges between two branches (see Figure 1). Most modern assembler programs use this paradigm.

While the de Bruijn graph can be constructed more efficiently than the overlap

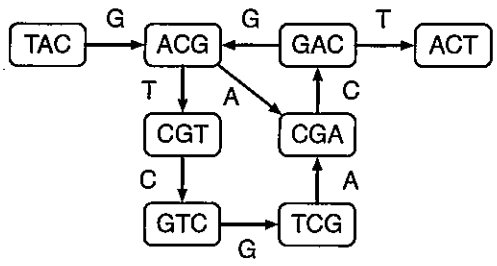

Figure 1: The $k = 3$ de Bruijn graph of reads 'TACGT', 'TACGA', 'ACGTC', 'GTCGA', 'CGACT', and 'CGACG'. The edges are given by the substrings of length $k + 1 = 4$ from all of the reads ('TACG', 'ACGA', 'ACGT', 'CGTC', and so on), and are represented by their right-most symbol connecting the two vertices given by their two substrings of length $k = 3$ (e.g. $TAC \xrightarrow{G} ACG$). The longest contig is found by starting at 'ACG', and following its branch labeled 'T', and all subsequent edges, until we reach another branch at vertex 'GAC' (which has two edges labeled 'G' and 'T'), giving us 'ACGTCGAC' (8 bases).

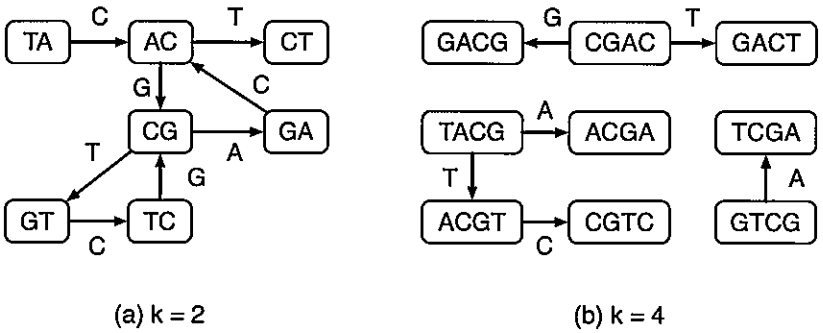

Figure 2: (a) The $k = 2$ de Bruijn graph of strings 'TACGT', 'TACGA', 'ACGTC', 'GTCGA', 'CGACT', and 'CGACG', and (b) the $k = 4$ de Bruijn graph of strings 'TACGT', 'TACGA', 'ACGTC', 'GTCGA', 'CGACT', and 'CGACG'. The longest contig for (a) is 'CGTCG' (5 bases), and the longest contig for (b) is 'TACGTC' (6 bases).

graph, it remains a bottleneck in assembly, both in terms of speed and size, with a de Bruijn graph of a human genome requiring 300 GB of RAM. Previous work has reduced this to 30 GB. This thesis reduces this to 2 GB, bringing it in line with commodity hardware – a student or field biologist could now perform this on their laptop. Around the same time as the work done in this thesis, an alternative approach with similar performance was published, but the Burrows-Wheeler based approach taken in this thesis offers more flexibility and faster edge traversal.

It is common for modern assemblers to build multiple de Bruijn graphs. This is because the k parameter significantly influences the topology – if k is too large there may be too few edges, causing gaps in the graph. But if k is too small, the vertices may have too many edges, increasing ambiguity. Both of these issues lead to shorter contigs, as is demonstrated in Figure 2. In fact, due to non-uniform coverage of NGS data, different areas of the same graph may benefit from differing

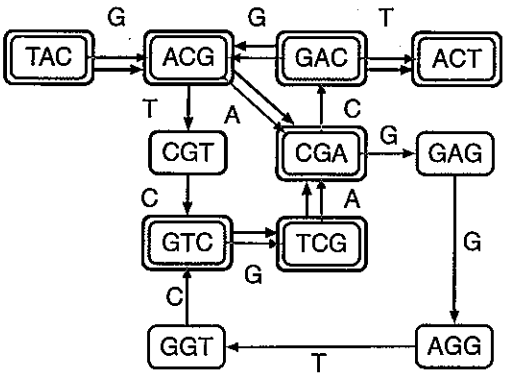

Figure 3: A $k = 3$ Colored de Bruijn Graph for two sets of reads. The black nodes and edges represent the reads 'TACGT', 'TACGA', 'ACGTC', 'GTCGA', 'CGACT', and 'CGACG' (the de Bruijn Graph from Figure 1). The gray nodes and edges represent the reads 'TACGA', 'GTCGACG', 'CGACT', 'CGAGGTC'.

k values. To overcome this, assemblers such as Spades and IDBA build de Bruijn graphs for increasing values of k, and use them in tandem. This yields better quality assemblies, but is slowed down proportionally to the number of k values used. This thesis introduces the first variation of the de Bruijn graph that can be built once, yet change k values on-the-fly, at only a modest increase in size over the base succinct de Bruijn graph, taking only 3.5 times the space, and only 30% longer to construct than a graph for a *single* value of k.

Finally, in population genomics, biologists assemble multiple genomes in order to study the variations, among, for example, 10,000 vertebrate genomes. To avoid constructing multiple graphs, Iqbal et al. proposed the Colored de Bruijn Graph. This graph capitalizes on the fact that DNA is rarely unique to an individual. It does this by first constructing a de Bruijn Graph of the entire populations NGS reads, and assigning each individual a unique *color*, which annotates the vertices and edges (see Figure 3). In this thesis, we further augment our succinct de Bruijn Graph to efficiently store these colors. When tested with four plant genomes, Iqbals structure required 101 GB RAM, while ours only requires 4 GB of RAM. Furthermore, our structure was able to store all known E. Coli genomes in 42 GB, where Iqbals was not able to complete, but is estimated to require 3 TB of RAM. We also demonstrate the use of our structure in creating a database of all Antimicrobial Resistance Genes, requiring 245 GB of RAM (an estimated 18 TB with Iqbals structure), for rapidly locating resilient bacterial outbreaks in food supply chains.

Original Papers

This thesis is comprised of the following three published papers, as well as a forth paper which is included as an appendix due to its relevance to the third paper while

not being core to this thesis.

Paper I: Succinct de Bruijn Graphs

Alexander Bowe, Taku Onodera, Kunihiko Sadakane, and Tetsuo Shibuya.
In *Algorithms in Bioinformatics. Proceedings of WABI 2012* (B. Raphael and J. Tang, editors). Lecture Notes in Computer Science, vol. 7534, pages 225–235. Springer, Berlin, Heidelberg, 2012.

We propose a new succinct de Bruijn graph representation. If the de Bruijn graph of k-mers in a DNA sequence of length N has m edges, it can be represented in $4m + o(m)$ bits. This is much smaller than existing representations. The numbers of outgoing and incoming edges of a node are computed in constant time, and the outgoing and incoming edge with given label are traversed in constant time and $\mathcal{O}(k)$ time, respectively. The data structure is constructed in $\mathcal{O}(Nk \log m / \log \log m)$ time using no additional space.

Paper II: Variable-Order de Bruijn Graphs

Christina Boucher, Alex Bowe, Travis Gagie, Simon J. Puglisi, and Kunihiko Sadakane.
In *Proceedings of the 2015 Data Compression Conference*, Snowbird, Utah, pages 383–392. IEEE, 2015.

The de Bruijn graph G_K of a set of strings S is a key data structure in genome assembly that represents overlaps between all the K-length substrings of S. Construction and navigation of the graph is a space and time bottleneck in practice and the main hurdle for assembling large genomes. This problem is compounded because state-of-the-art assemblers do not build the de Bruijn graph for a single order (value of K) but for multiple values of K: they build d de Bruijn graphs, each with a specific order, i.e., $G_{K_1}, G_{K_2}, \ldots, G_{K_d}$. This paradigm increases the quality of the assembly produced, at the cost of greatly increases runtime, due to constructing d graphs instead of one. In this paper, we show how to augment a succinct de Bruijn graph representation by Bowe et al. (Proc. WABI, 2012) to support new operations that let us change order on the fly, effectively representing all de Bruijn graphs up to some maximum order K in a single data structure. Our experiments show our variable-order de Bruijn graph only modestly increases space usage, construction time, and navigation time compared to a single order graph.

Paper III: Succinct Colored de Bruijn graphs

Martin D. Muggli, Alexander Bowe, Noelle R. Noyes, Paul S. Morley, Keith E. Belk, Robert Raymond, Travis Gagie, Simon J. Puglisi, and Christina Boucher.
Bioinformatics, 33(20):3181–3187, 2017.

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 requires

additional information to be succinctly encoded as well as support for non-standard traversal operations. 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.

Paper IV: Relative Select (Appendix)

Christina Boucher, Alexander Bowe, Travis Gagie, Giovanni Manzini, and Jouni Sirn
In *String Processing and Information Retrieval. Proceedings of SPIRE 2015* (C. Iliopoulos, S. Puglisi, and E. Yilmaz, editors). Lecture Notes in Computer Science, vol. 9309, pp. 149–155. Springer, Cham, 2015.

Motivated by the problem of storing coloured de Bruijn graphs, we show how, if we can already support fast select queries on one string, then we can store a little extra information and support fairly fast select queries on a similar string.

Form8・Separate Sheet
(様式8・別紙)

Results of the doctoral thesis screening

博士論文審査結果

Name in Full
氏名　Alexander BOWE

Title
論文題目　Succinct de Bruijn Graphs

出願者はバイオ情報学におけるアセンブリングの重要性と、アセンブリングに対する de Bruijn Graph 有効性に着目し、de Bruijn Graph とその拡張について、効率的な圧縮方法を提案することで実用での利便性を飛躍的に向上する方法を提案した。

第1章では、バイオ情報学におけるアセンブリング問題の紹介と、その重要性が説明され、アセンブリングを行う有効な手段として、de Bruijn Graph を使う方法が解説されている。また、de Bruijn Graph の定義とその性質、アセンブリングにおける利用法について説明している。

次に、第2章では、氏の開発した De Bruijn Graph の効率的な圧縮手法についての解説している。要素技術となる Suffix array や rank select アルゴリズムの利用法と、BWT アルゴリズムの利用による圧縮の効率向上の手法を紹介し、最後に計算実験の結果を示している。氏のアルゴリズムにより、既存手法による圧縮を 10 倍から 100 倍効率化することに成功し、計算コストに関しても大きな損失がないことを示している。

第3章では、de Bruijn Graph の長さパラメータ k を、追加的な計算用のメモリ空間の確保なしに変更し、新たな k に対する De Bruijn Graph を構築するアルゴリズムについての説明がある。de Bruijn graph の各ノードに suffix array 上で直上にあるノードとの LCS をデータとして付与することにより、効率良い更新が可能となることが説明されており、計算実験の結果によってもそれが支持されている。

第4章では、氏が開発した、色付き de Bruijn Graph に対する初めての Succinct データ構造について説明されている。既存の手法を用いた場合の計算コストを大幅に上回る効率化を実現しており、通常のコンピュータに搭載不可能であったデータをノート PC クラスのコンピュータに実装可能としていることが示されている。

なお、出願者の研究成果は、中心的な de Bruijn Graph に関する業績が、査読付きの英文ジャーナル Bioinformatics に掲載されている。出願者は第2著者であるが、バイオ情報学の分野では生物学の研究者と情報学の研究者が協力して研究を行うことが多く、両研究者ともに当該分野での第一著者に匹敵する研究活動が必要となるため、この分野では、第一著者と第2著者が生物学と情報学の研究者であり、かつ相当の貢献をしている場合、両者に博士取得要件としての論文の利用が認められている。このことはすでに情報学専攻専攻委員会で説明されて審議され、この業績を博士取得要件として認定することが可能であると許可されている。これ以外の業績に関しても、アルゴリズム分野、バイオ情報学分野でのトップレベル国際カンファレンスである SPIRE と DCC に、第2著者として採択されている。両者ともに英語で記述され、査読付きである。

発表、質疑応答は英語で行われた。質疑応答の時間には、計算実験結果についての質問、アルゴリズムの詳細に関する質問、発表者の貢献は　実装面なのか理論面なのか、といった質問が行われた。BOWE 氏は、論文に記述されていることに関しては、そのことに関して丁寧に回答を行い、貢献については実装と理論の両面であるなど納得のいく回答を行った。

審査委員会では、論文の書き方について、軽微な修正の要求があったが、そのことを除いては特に博士号の授与に異論を唱えるものはなく、技術的、学術的に高いレベルであるとの認識から、氏に博士号を授与することは至極妥当であるとの判断に至った。