We Are What We Index; a Primer for the Wheeler Graph Era

Ben Langmead
Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA

Abstract

Since the arrival of second-generation sequencing, we have needed to build indexes over reference sequences – e.g. genomes and transcriptomes – in order to solve read alignment and classification problems efficiently [11, 12, 13]. The rule has been: what we can index determines what we can do. When indexing strings, we can use methods like suffix arrays [15], the Burrows-Wheeler Transform (BWT) [4] / FM Index [7], or k-mer indexes [16]. What if we want to index objects more complex than strings? A pangenome, for example, is a large collection of similar strings, e.g. the hundreds of assemblies that make up the Human Pangenome Reference [14] or all the bacteria in the Refseq database [10]. We may wish to combine these strings into a multiple sequence alignment (MSA) or a graph first. Can we index those efficiently? In many useful cases the answer is "yes," but in others the answer is "no." The story of how we learned exactly when the answer is "yes" versus "no" unfolded through a sequence of insights. Here we review this story, eventually arriving at the definition of Wheeler graphs as discovered and formalized by Gagie, Manzini and Sirén [8].

We will focus on indexes based on the BWT, since these (a) are lossless full-text indexes, (b) are widely used in practice [11, 12], and (c) form the theoretical throughline for all the indexing strategies on the path to Wheeler graphs. We will trace the BWT-based indexing story from the early days of the FM Index, though its step-by-step gobbling up of trees (XBW-transform [6]) and de Bruijn Graphs (BOSS representation [3]), and to the eventual formalization of Wheeler graphs [8]. Along the way, we will define and update our notions of what it means to track a consecutive range of elements in the structure, and what it means for an index to be efficient. We will also connect these notions to automata [17], noting how the indexability of Wheeler graphs (also called Wheeler automata) is connected to the mechanics of how to efficiently represent and simulate a finite automaton [1].

With this context, we can imagine improved indexes for the future of genomics and pangenomics. De Bruijn are extremely practical and are the most widely used among the non-string data structures that are also Wheeler graphs. But we might prefer other options. For example, de Bruijn graphs have the undesirable property that they usually encode not only the true longer-than-k substrings of the original text, but also "false" substrings that span repeats. Related to this, paths through the de Bruijn graph can "glue" substrings together that are horizontally distant in the MSA. Could other Wheeler graphs be practical alternatives to de Bruijn graphs? For instance, the original GCSA study by Sirén, Välimäki and Mäkinen proposed a way to convert a multiple alignment into an automaton that either is a Wheeler graph or can be made into one [18]. This warrants further exploration, possibly with the help of improved tools for solving the NP-complete problem of recognizing whether a graph is a Wheeler graph [5]. The notion of BWT tunnels [2] gives another route: we can begin with a concatenated pangenome strings and compress it by identifying and collapsing BWT tunnels. This yields a Wheeler graph that is compressed like the de Bruijn graph, but without departing from the exact contents or coordinate systems of the original genomes. The future might need us to explore all these Wheeler-graph indexes, along with the also highly practical and always-improving world of indexes buiover collections of strings [9].

2012 ACM Subject Classification Applied computing → Computational genomics; Applied computing → Computational genomics

Keywords and phrases Indexing, Burrows-Wheeler Transform

Digital Object Identifier 10.4230/LIPIcs.WABI.2025.2

Category Invited Talk

Funding *Ben Langmead*: NIH NIGMS grant R35GM139602 and NIH NHGRI grant R01HG011392

25th International Conference on Algorithms for Bioinformatics (WABI 2025).
Editors: Broňa Brejová and Rob Patro; Article No. 2; pp. 2:1–2:2
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

References

1 Jarno Alanko, Giovanna D'Agostino, Alberto Policriti, and Nicola Prezza. Wheeler languages. *Information and Computation*, 281:104820, 2021. doi:10.1016/J.IC.2021.104820.

2 Uwe Baier. On Undetected Redundancy in the Burrows-Wheeler Transform. In Gonzalo Navarro, David Sankoff, and Binhai Zhu, editors, *29th Annual Symposium on Combinatorial Pattern Matching (CPM 2018)*, volume 105 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CPM.2018.3.

3 Alexander Bowe, Taku Onodera, Kunihiko Sadakane, and Tetsuo Shibuya. Succinct de bruijn graphs. In Ben Raphael and Jijun Tang, editors, *Algorithms in Bioinformatics*, pages 225–235, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. doi:10.1007/978-3-642-33122-0_18.

4 Michael Burrows and David J Wheeler. A block-sorting lossless data compression algorithm. *Digital Equipment Corporation*, 1994.

5 Kuan-Hao Chao, Pei-Wei Chen, Sanjit A. Seshia, and Ben Langmead. WGT: Tools and algorithms for recognizing, visualizing, and generating Wheeler graphs. *iScience*, 26(8), August 2023. doi:10.1016/j.isci.2023.107402.

6 Paolo Ferragina, Fabrizio Luccio, Giovanni Manzini, and S. Muthukrishnan. Structuring labeled trees for optimal succinctness, and beyond. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 184–193, 2005. doi:10.1109/SFCS.2005.69.

7 Paolo Ferragina and Giovanni Manzini. Opportunistic data structures with applications. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 390–398, 2000. doi:10.1109/SFCS.2000.892127.

8 Travis Gagie, Giovanni Manzini, and Jouni Sirén. Wheeler graphs: A framework for bwt-based data structures. *Theoretical computer science*, 698:67–78, 2017. doi:10.1016/J.TCS.2017.06.016.

9 Travis Gagie, Gonzalo Navarro, and Nicola Prezza. Optimal-time text indexing in bwt-runs bounded space. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1459–1477. SIAM, 2018. doi:10.1137/1.9781611975031.96.

10 Tamara Goldfarb, Vamsi K Kodali, Shashikant Pujar, Vyacheslav Brover, Barbara Robbertse, Catherine M Farrell, Dong-Ha Oh, Alexander Astashyn, Olga Ermolaeva, Diana Haddad, et al. NCBI RefSeq: reference sequence standards through 25 years of curation and annotation. *Nucleic Acids Research*, 53(D1):D243–D257, 2025.

11 Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome biology*, 10:1–10, 2009.

12 Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009. doi:10.1093/BIOINFORMATICS/BTP324.

13 Ruiqiang Li, Chang Yu, Yingrui Li, Tak-Wah Lam, Siu-Ming Yiu, Karsten Kristiansen, and Jun Wang. SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*, 25(15):1966–1967, 2009. doi:10.1093/BIOINFORMATICS/BTP336.

14 Wen-Wei Liao, Mobin Asri, Jana Ebler, Daniel Doerr, Marina Haukness, Glenn Hickey, Shuangjia Lu, Julian K Lucas, Jean Monlong, Haley J Abel, et al. A draft human pangenome reference. *Nature*, 617(7960):312–324, 2023.

15 Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *siam Journal on Computing*, 22(5):935–948, 1993. doi:10.1137/0222058.

16 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):1–12, 2021.

17 Michael Sipser. *Introduction to the Theory of Computation.* International Thomson Publishing, 1st edition, 1996.

18 Jouni Sirén, Niko Välimäki, and Veli Mäkinen. Indexing graphs for path queries with applications in genome research. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(2):375–388, 2014. doi:10.1109/TCBB.2013.2297101.