Indexing Compressed Text: A Tale of Time and Space

Nicola Prezza
LUISS Guido Carli, Rome, Italy
https://nicolaprezza.github.io/
nprezza@luiss.it

Abstract

Text indexing is a classical algorithmic problem that has been studied for over four decades. The earliest optimal-time solution to the problem, the suffix tree [11], dates back to 1973 and requires up to two orders of magnitude more space than the text to be stored. In the year 2000, two breakthrough works [6, 3] showed that this space overhead is not necessary: both the index and the text can be stored in a space proportional to the text's entropy. These contributions had an enormous impact in bioinformatics: nowadays, the two most widely-used DNA aligners employ compressed indexes [9, 8]. In recent years, it became apparent that entropy had reached its limits: modern datasets (for example, collections of thousands of human genomes) are extremely large but very repetitive and, by its very definition, entropy cannot compress repetitive texts [7]. To overcome this problem, a new generation of indexes based on dictionary compressors (for example, LZ77 and run-length BWT) emerged [7, 5, 1], together with generalizations of the indexing problem to labeled graphs [2, 10, 4]. This talk is a short and friendly survey of the landmarks of this fascinating path that took us from suffix trees to the most modern compressed indexes on labeled graphs.

2012 ACM Subject Classification Theory of computation → Data compression; Theory of computation → Sorting and searching; Theory of computation → Pattern matching

Keywords and phrases Compressed Text Indexing

Digital Object Identifier 10.4230/LIPIcs.SEA.2020.3

Category Invited Talk

References

1. F. Claude and G. Navarro. Improved grammar-based compressed indexes. In *Proc. 19th International Symposium on String Processing and Information Retrieval (SPIRE)*, LNCS 7608, pages 180–192, 2012.
2. Paolo Ferragina, Fabrizio Luccio, Giovanni Manzini, and S. Muthukrishnan. Compressing and indexing labeled trees, with applications. *J. ACM*, 57(1), November 2009. doi:10.1145/1613676.1613680.
3. Paolo Ferragina and Giovanni Manzini. Opportunistic data structures with applications. In *41st Annual Symposium on Foundations of Computer Science, 2000.*, pages 390–398. IEEE, 2000.
4. Travis Gagie, Giovanni Manzini, and Jouni Sirén. Wheeler graphs: A framework for BWT-based data structures. *Theoretical Computer Science*, 698:67–78, 2017. Algorithms, Strings and Theoretical Approaches in the Big Data Era (In Honor of the 60th Birthday of Professor Raffaele Giancarlo). doi:10.1016/j.tcs.2017.06.016.
5. Travis Gagie, Gonzalo Navarro, and Nicola Prezza. Fully functional suffix trees and optimal text searching in bwt-runs bounded space. *J. ACM*, 67(1), January 2020. doi:10.1145/3375890.
6. Roberto Grossi and Jeffrey Scott Vitter. Compressed suffix arrays and suffix trees with applications to text indexing and string matching (extended abstract). In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, STOC '00, page 397–406, New York, NY, USA, 2000. Association for Computing Machinery. doi:10.1145/335305.335351.

7 S. Kreft and G. Navarro. On compressing and indexing repetitive sequences. *Theoretical Computer Science*, 483:115–133, 2013.

8 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(3):R25, 2009.

9 Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009.

10 Jouni Sirén, Niko Välimäki, and Veli Mäkinen. Indexing graphs for path queries with applications in genome research. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 11(2):375–388, March 2014. doi:10.1109/TCBB.2013.2297101.

11 Peter Weiner. Linear pattern matching algorithms. In *Switching and Automata Theory, 1973. SWAT'08. IEEE Conference Record of 14th Annual Symposium on*, pages 1–11. IEEE, 1973.