Recursive Parsing and Grammar Compression in the Era of Pangenomics

Christina Boucher
Department of Computer and Information Science and Engineering,
Herbert Wertheim College of Engineering, University of Florida, Gainesville, FL, USA

Abstract

Prefix-Free Parsing (PFP) and its recursive variant (RPFP) provide a scalable framework for compressing and indexing large genomic datasets. By enabling efficient construction of succinct data structures, these methods support fast and memory-efficient read alignment across thousands of genomes. Their deterministic and modular design makes them especially well-suited for pangenomics and large-scale sequence analysis.

2012 ACM Subject Classification Applied computing → Bioinformatics

Keywords and phrases Prefix-Free Parsing, Recursive Prefix-Free Parsing, Grammar-Based Compression, Succinct Data Structures, RePair Compression

Digital Object Identifier 10.4230/LIPIcs.WABI.2025.1

Category Invited Talk

1 Introduction

The exponential growth of genomic data, driven by advances in sequencing technologies, has led to a pressing need for algorithms and data structures that can efficiently index, store, and search massive and highly repetitive datasets. Traditional approaches do not scale well for this modern regime of data. Prefix-Free Parsing (PFP) has emerged as a powerful technique that addresses these scalability challenges by enabling the construction of compressed text indexes through deterministic and memory-efficient parsing [1, 2]. It has become central to grammar-based compression and succinct data structure design, particularly in pangenomics, where large-scale indexing of population-level genome variation is essential [3, 4, 7]. Prefix-Free Parsing segments an input string into non-overlapping phrases based on a rolling hash function and a set of trigger strings. Phrase boundaries are inserted at positions where a trigger occurs, ensuring that each resulting phrase is context-independent and that the parsing is deterministic. This parsing produces two outputs: a dictionary containing the unique phrases and a parse representing the input as a sequence of dictionary phrase identifiers. This method allows for significant compression of repetitive sequences while maintaining the ability to reconstruct the original text. It also serves as a natural foundation for downstream indexing data structures, such as the run-length FM-index (r-index), as the parse retains the necessary ordering and alignment properties of the original sequence [2, 6].

2 Recursive Prefix-Free Parsing

Recursive Prefix-Free Parsing (RPFP) generalizes this approach by applying PFP to the parse itself, yielding a second-level dictionary and parse. This recursive decomposition further compresses the representation and enables deeper redundancy to be exploited. RPFP results in a multi-layered hierarchy of phrase structures, which is particularly effective for large, highly repetitive inputs such as multiple genomes from the same species. The recursive application of PFP enables construction of the r-index and other compressed data structures directly from the recursive parse and dictionary, avoiding the need to reconstruct intermediate levels such as the suffix array or full BWT. This facilitates the construction of indexes at scales previously considered infeasible and allows for efficient querying, such as finding maximal exact match, even over extremely large genomic collections [3].

3 Applications to Prefix-Free Parsing

PFP is naturally compatible with grammar-based compression methods. Each phrase identified during parsing can serve as a non-terminal in a context-free grammar. This property allows for the construction of compressed grammars that support fast and space-efficient decompression. In particular, grammars built from PFP and RPFP parses can be used to create structures similar to RePair, with the added advantage of being computable using only the dictionary and parse [5]. The resulting grammars are compact and well suited for compressed indexing and searching. Because these grammars preserve phrase boundaries, they can be used for local decompression and read alignment, which is especially useful for high-throughput applications such as read mapping, pangenome alignment, and variant detection.

In pangenomics, the task of indexing large and variation-rich datasets demands data structures that scale with the amount of redundancy rather than the total sequence length. PFP and RPFP enable such scaling by capitalizing on the shared structure of homologous sequences. This allows the construction of succinct indexes over collections of hundreds or thousands of genomes using limited memory and computation. These indexes support efficient read alignment by identifying maximal exact matches between sequencing reads and indexed genomes, even in the presence of high variation. Most recently, we developed Moni-Align [8], which enables accurate and memory-efficient alignment of short reads to large pangenomic collections by leveraging the run-length compressed r-index.

References

1 Christina Boucher, Travis Gagie, Alan Kuhnle, Ben Langmead, Giovanni Manzini, and Taher Mun. Prefix-free parsing for building big BWTs. *Algorithms Molecular Biology*, 14(1):13:1–13:15, 2019. doi:10.1186/S13015-019-0148-5.

2 Christina Boucher, Travis Gagie, Alan Kuhnle, and Giovanni Manzini. Prefix-free parsing for building big BWTs. In *Proceedings of the Workshop of Algorithms in Biology (WABI)*, pages 2:1–2:16, 2018. doi:10.4230/LIPICS.WABI.2018.2.

3 Eddie Ferro, Marco Oliva, Travis Gagie, and Christina Boucher. Building a pangenome alignment index via recursive prefix-free parsing. *iScience*, 27(10):110933, 2024.

4 Aaron Hong, Massimiliano Rossi, and Christina Boucher. LZ77 via Prefix-Free Parsing. In *Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX 2023)*, pages 123–134. SIAM, 2023. doi:10.1137/1.9781611977561.CH11.

5 Justin Kim, Rahul Varki, Marco Oliva, and Christina Boucher. Re^{2}Pair: Increasing the Scalability of RePair by Decreasing Memory Usage. In *Proceedings of the 32nd Annual European Symposium on Algorithms (ESA 2024)*, pages 78:1–78:15, 2024. doi:10.4230/LIPICS.ESA.2024.78.

6 Taher Mun, Alan Kuhnle, Christina Boucher, Travis Gagie, Ben Langmead, and Giovanni Manzini. Matching Reads to Many Genomes with the r-Index. *Journal of Computational Biology*, 27(4):514–518, 2020. doi:10.1089/cmb.2019.0316.

7 Marco Oliva, Davide Cenzato, Massimiliano Rossi, Zsuzsanna Lipták, Travis Gagie, and Christina Boucher. CSTs for Terabyte-Sized Data. In *Proceedings of the 2022 Data Compression Conference (DCC 2022)*, pages 93–102. IEEE, 2022. doi:10.1109/DCC52660.2022.00017.

8 Ravin Varki, Matteo Rossi, Eddie Ferro, Marco Oliva, Erik Garrison, Ben Langmead, and Christina Boucher. Accurate short-read alignment through r-index-based pangenome indexing. *Genome Research*, June 2025. doi:10.1101/gr.279858.124.