Novel Grammar-Based Compression Algorithms for Pangenome Analysis

Jordan Dood* and Alan Cleary‡

* Montana State University, Bozeman, Montana; ‡ National Center for Genome Resources, Santa Fe, New Mexico

Introduction

Recent advancements in DNA sequencing and assembly have drastically lowered cost and improved quality. This has allowed for collections of genomes to be created that better reflect the variability within a single species. These *pangenomes* continue to grow in size and scope as new sequences are added, yet such collections have already proven to be challenging to handle without significant computational infrastructure, with the primary challenge being the large data size. Unfortunately, existing compression algorithms do not allow analysis to be performed directly on the compressed data. Furthermore, many common compression paradigms do not take advantage of the high similarity between genomes from the same species, resulting in compression that scales relative to data size rather than relative to information content.

In this work, we present and propose novel grammar-based compression algorithms designed specifically for pangenome analysis. By leveraging maximal repeats, these algorithms have the potential to enable pangenome analysis at unprecedented scales.

1. Pangenome Representation

When designing data-intensive applications it is important to represent the data using a model that is best suited for the intended usage of the data [1]. Much work has been done to represent pangenomes as graphs, or *pangenome graphs* (**Fig. 1**) [2]. While this is a fairly intuitive visual representation of sequence-level variation within a population, it has proven to be a challenging data model for computation for the following reasons:

- Pangenome graphs can be computationally expensive to construct
- The graph construction method can affect downstream analysis
- Pangenome graphs require secondary indexing for many analyses
- Adding new genomes to an existing pangenome graph invalidates secondary indexes and prior graph analyses
- Pangenome graphs and secondary indexes can be much larger on disk than the data they represent
- Pangenome graphs are a novel data model that is largely unrelated to existing string data structures

The problem of data model size is especially important given the current and projected growth of pangenomic data sets [2]. Not only should the data model accommodate pangenomic analyses, it should also represent the data in a manner that uses less space than the original data and allows computation to be performed directly on the compressed data.

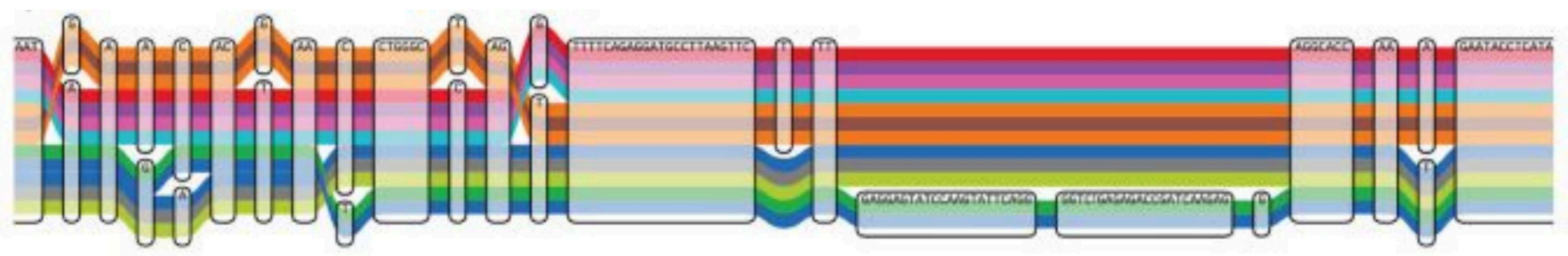

Fig 1. Example of a pangenome graph of GRCh38 and its alternate sequences in the gene HLA-DRB1 built with VG msga (Variation Graph multiple sequence/graph aligner) [2].

2. Compression Requirements

Generally the need for better compression of genomic data is widespread and immediate [3], yet the requirements of compression techniques for pangenome analysis have not been formalized. Here we briefly outline a set of formal requirements:

1. **Lossless** - the original data can be reconstructed from the compressed data with no loss of information
2. **Self-indexing** - the compressed data should serve as an index of the original data, e.g. can be efficiently searched without additional indexing
3. **Random access** - any subset of the original data should be efficiently accessible from the compressed data
4. **Updatable** - the compressed data can be updated (add/remove/edit) without requiring total recompression
5. **Scalable** - compression and computation should be parallelizable and be done using finite memory

Unfortunately no genomic or general purpose compression techniques satisfy all of these criteria. Additionally, compression of genomic data is a generally hard problem [3]. This necessitates the need for novel pangenomic compression algorithms.

3. Grammar-Based Compression (GBC)

A *context-free grammar* (CFG) is a set of recursive rules that can be used to derive a string (**Fig. 2**). Computing a CFG for a single string is a form of lossless compression known as *grammar-based compression* (GBC). Grammar-based compressors are generally fast to compute and achieve good compression in practice as they tend to scale relative to information content rather than relative to data size. Additionally, <u>all grammar-based compressors satisfy the lossless, self-indexing, and random access compression requirements</u> [4,5], making GBC a good candidate data model for pangenome analysis.

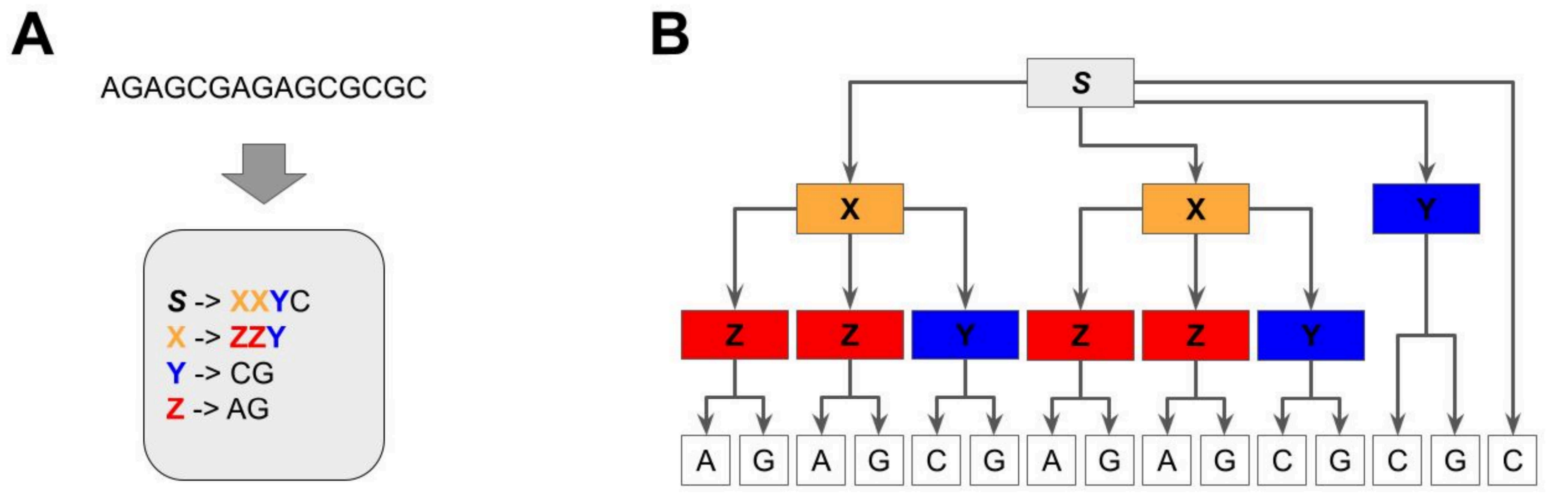

Fig 2. Example of a context-free grammar (CFG). **A)** A string and a CFG that encodes it. **B)** The parse tree that generates the original string from the CFG.

4. GBC for Pangenome Analysis

GBC is already competitive with existing compression algorithms on single genomes (**Fig. 3**). Additionally, the GBC property of scaling relative to information content rather than relative to data size is ideal for pangenomics. A population that has a finite number of genes has a *closed pangenome*, meaning after a sufficient number of genomes are added to the pangenome, adding additional genomes will introduce very little additional sequence content [2]. This means <u>the size of the full pangenome can be theoretically predicted and bounds the size of the pangenome's CFG</u> (**Fig. 4**), making GBC truly scalable in the number of genomes it can represent in a single pangenome.

However, GBC is currently not scalable in its search algorithms, with the current state of the art being linear in the size of the CFG rather than the search string [6]. There is also no known algorithms for updating a CFG that has already been computed. <u>Addressing these algorithmic challenges will enable pangenome analysis at unprecedented scales</u>.

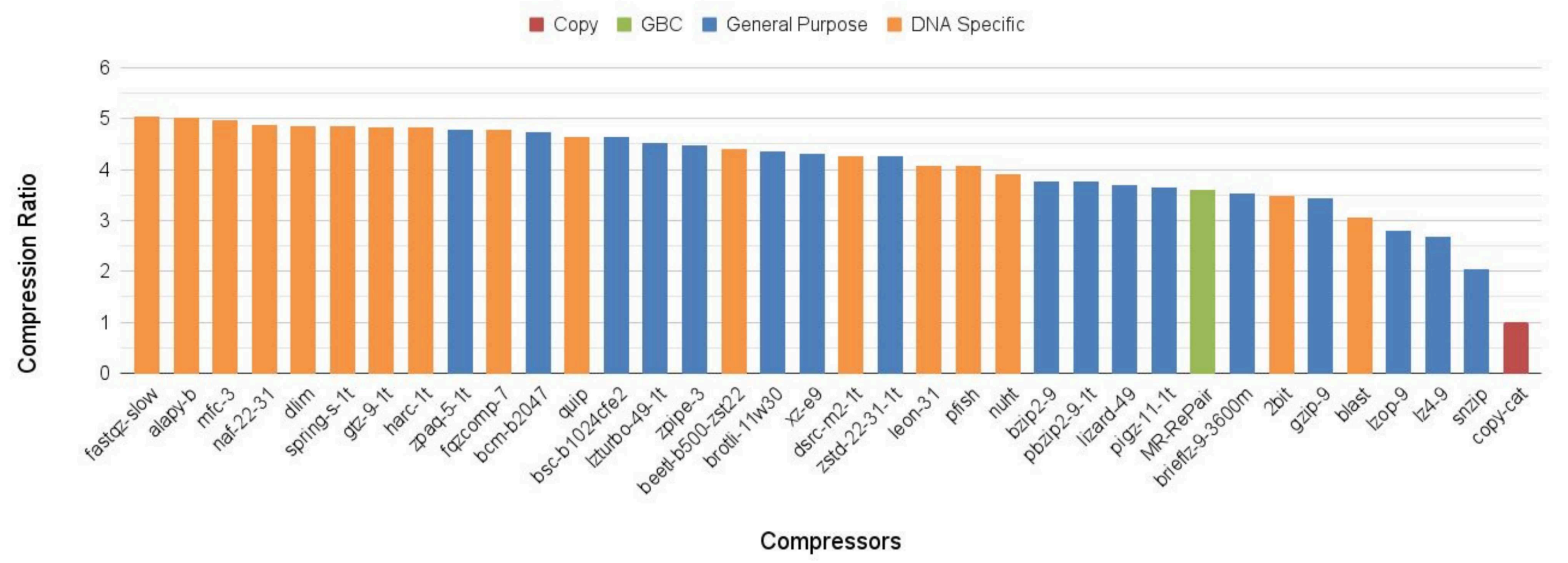

Fig 3. The compression ratio of the MR-RePair GBC algorithm (green) [4] versus DNA specific (orange) and general purpose (blue) compression algorithms on a *Homo sapien* genome [3]. Copy-cat (red) simply copies the sequence and is included as a control.

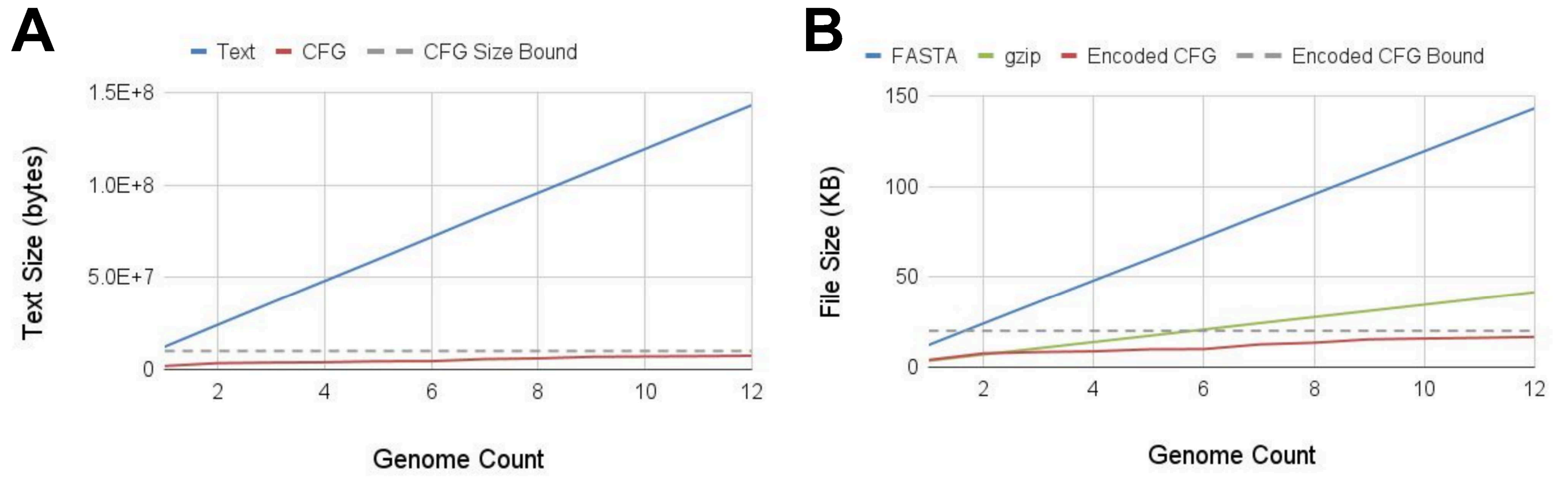

Fig 4. The size of MR-RePair CFGs [4] for a yeast pangenome of increasing size [7]. **A)** The number of characters in the uncompressed genomes (blue) and the number of characters in the CFGs (red). **B)** The file sizes of the uncompressed genomes (blue), the genomes compressed with gzip (green), and the encoded CFGs from **A** (red). **A & B)** <u>The dashed grey line depicts the bound on the CFG size determined by the theoretical size of the full pangenome</u>.

5. Novel Algorithms

A *maximal repeat* is a repeat in a string that can not be extended into a larger repeat unless the larger repeat occurs fewer times in the string. In [4] the authors present a GBC algorithm based on maximal repeats (used in **Fig. 3 & 4**). In [5] we present a similar algorithm that establishes a connection between maximal repeat-based GBC algorithms and other string data structures.

<u>The maximal repeats of a string are idempotent relative to other strings, meaning strings in a set may share maximal repeats but adding or removing a string in the set will not change the maximal repeats of the other strings</u>. This provides an opportunity to extend GBC algorithms like [4] and [5] to update CFGs over time and to construct CFGs in parallel, since a CFG can be computed independently for each string and then combined with others by identifying shared maximal repeats (**Fig. 5**). <u>The space required to combine CFGs is relative to their size, which is bounded by the theoretical size of the full pangenome</u> (**Fig. 4**), potentially making this a highly scalable approach to satisfying the updatable compression requirement.

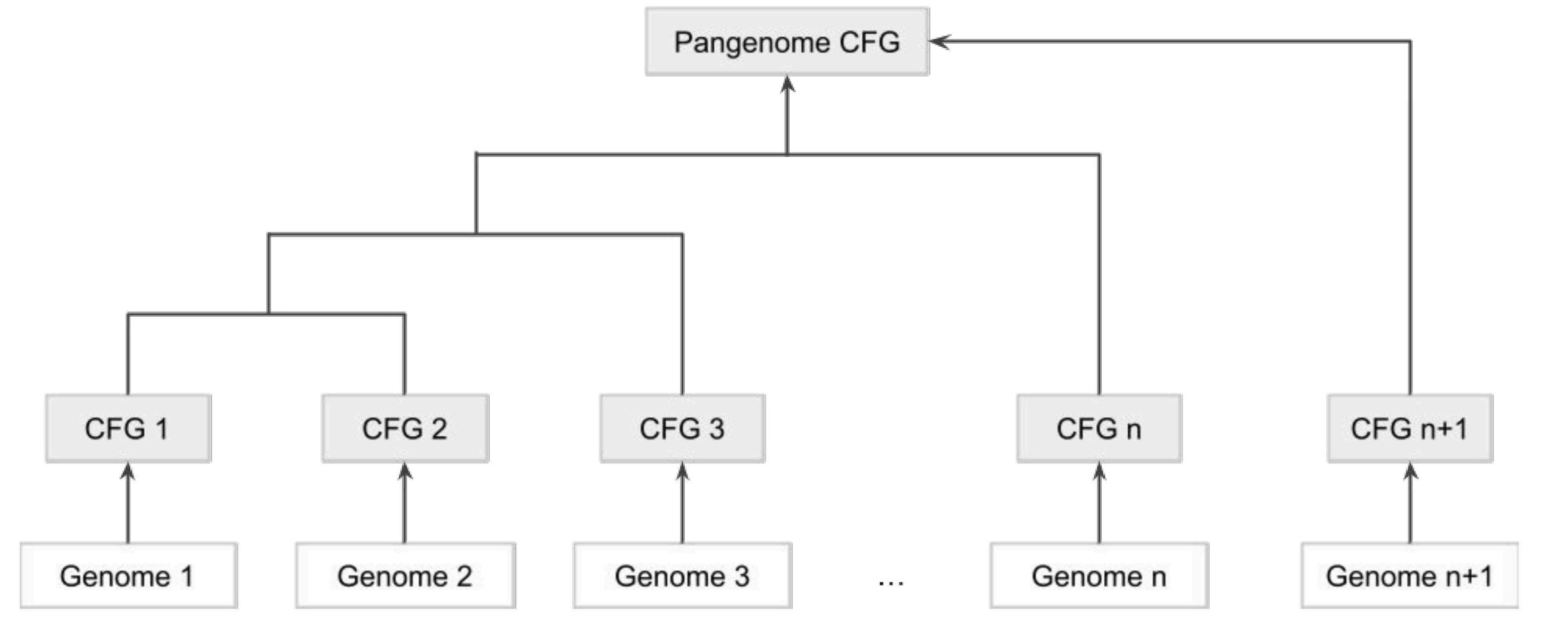

Fig 5. Constructing a pangenome CFG in parallel by pairwise combing the CFGs of *n* genomes. An *n+1* genome is later added using the same technique.

<u>Perhaps the most challenging algorithmic innovation required to use GBC for pangenome analysis at scale is search that scales independently of the CFG size</u>. Although the CFG size is bounded by the theoretical size of the full pangenome, search that scales relative to CFG size is not sufficient for search intensive analyses, such as read mapping. By establishing a connection between maximal repeat-based GBC algorithms and other string data structures, our work in [5] may allow such search algorithms that scale independently of the data size to be adapted from other data structures, such as the enhanced suffix array and FM-index.

References

1. Kleppmann, Martin. *Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems*. " O'Reilly Media, Inc.", 2017.
2. Eizenga, Jordan M., et al. "Pangenome graphs." *Annual review of genomics and human genetics* 21 (2020): 139-162.
3. Kryukov, Kirill, et al. "Sequence Compression Benchmark (SCB) database—A comprehensive evaluation of reference-free compressors for FASTA-formatted sequences." *GigaScience* 9.7 (2020): giaa072.
4. Furuya, Isamu, et al. "MR-RePair: Grammar compression based on maximal repeats." *2019 Data Compression Conference (DCC)*. IEEE, 2019.
5. Cleary, Alan, and Jordan Dood. "Constructing the CDAWG CFG using LCP-Intervals." *2023 Data Compression Conference (DCC)*. IEEE, 2023.
6. Ganardi, Moses, and Paweł Gawrychowski. "Pattern Matching on Grammar-Compressed Strings in Linear Time." *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. Society for Industrial and Applied Mathematics, 2022.
7. Yue, Jia-Xing, et al. "Contrasting evolutionary genome dynamics between domesticated and wild yeasts." *Nature genetics* 49.6 (2017): 913-924.

Funding

Award #2105391

We are looking for collaborators and funding. Contact Dr. Cleary if interested: acleary@ncgr.org