# Partitioned Multi-MUM Finding for Scalable Pangenomics

**Vikram S. Shivakumar**[1]
Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA

**Ben Langmead**[2]
Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA

## Abstract

Pangenome collections continue to grow and proliferate to hundreds of high-quality genomes, for example, the expanded v2 version of the Human Pangenome Reference Consortium (HPRC) dataset spanning 474 human haplotypes [4]. As the size and complexity of these collections grow, it is increasingly important that our methods for studying and indexing pangenomes be scalable and updateable. Maximal Unique Matches (multi-MUMs), exact substring matches present exactly once in all sequences in a pangenome collection, represent conserved anchor sequences that can comprise a common coordinate system. We previously proposed a framework and tool called Mumemto for rapidly identifying multi-MUMs during construction of a compressed pangenome index [6]. Using prefix-free parsing (PFP) [1], a compressed-space method for computing full-text indexes, Mumemto outperforms existing methods for identifying multi-MUMs. However, one drawback remains updateability and scalability. Mumemto can become memory-intensive for large pangenomes (> 300 human genomes), and as newly assembled genomes are added to a pangenome collection, Mumemto requires re-running on the entire updated collection.

To address this, we developed a partition-merging approach to compute multi-MUMs with Mumemto. We introduce two strategies for merging of multi-MUMs computed across different collections (see Figure 1), enabling parallelization across partitions and simple computation of multi-MUMs for incrementally-updated collections. The first strategy requires a common sequence in each partition (which we call "anchor-based merging"), which serves as a coordinate system to identify multi-MUM overlaps between partitions. By tracking the next longest match for all multi-MUMs and unique matches (UMs) in an auxiliary data structure, intersections between matches can be filtered out if no longer unique in the union collection. The second strategy identifies overlaps directly from the multi-MUM substrings (called "string-based merging"). The overlaps are identified by running Mumemto over the extracted multi-MUM sequences and are similarly filtered out if they are too short to considered unique. Lastly, we propose an extension to anchor-based merging to enable the computation of partial multi-MUMs, present in only a subset of sequences in the union set.

The partition-merging framework introduces a tradeoff space in Mumemto between running time and memory, depending on partition size and the number of threads. Running parallel, per-partition Mumemto processes and merging the results reduces the running time but increases the peak memory footprint, while running a single Mumemto thread over each partition serially yields longer running time but a smaller memory footprint. To evaluate this tradeoff, we computed multi-MUMs across 474 haplotypes of chr19 from the HPRC v2 dataset [4] and 69 assemblies of *A. thaliana* [3] (Table 1).

The string-based method also enables merging multi-MUMs between disjoint collections, for example subclades in a phylogenetic tree. By merging multi-MUMs along the shape of the tree, we can compute matches at internal nodes of the tree along with the root, revealing clade-specific conservation and structural variation. Multi-MUM merging also enables interspecific match computation, which was previously infeasible with Mumemto due to high memory usage for highly-diverse input sequence collections. We use partition-merging to compute multi-MUMs across 29 primate assemblies, and found a correspondence to ultraconserved elements previously found across mammalian genomes [2].

[1] Corresponding Author
[2] Corresponding Author

25th International Conference on Algorithms for Bioinformatics (WABI 2025).
Editors: Broňa Brejová and Rob Patro; Article No. 23; pp. 23:1–23:4
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We show that a partitioned Mumemto enables scalability to growing pangenome collections and expands the applicability of Mumemto to larger, more diverse datasets. As a result, Mumemto is the only method capable of computing exact matches across the entire HPRC v2 dataset (474 haplotypes [4]), and can easily incorporate future releases of assemblies without recomputation. This increases the scope for exploration of genomic conservation and variation and highlights the potential for Mumemto as a core method for future pangenomics and comparative genomics research.

The partitioned Mumemto framework is implemented in v1.3.0 and is available open-source at `https://github.com/vikshiv/mumemto`.

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

**Keywords and phrases** Pangenomics, Comparative genomics, Compressed indexing

**Digital Object Identifier** 10.4230/LIPIcs.WABI.2025.23

**Category** Extended Abstract

**Related Version** *Full Version*: `https://www.biorxiv.org/content/10.1101/2025.05.20.654611`

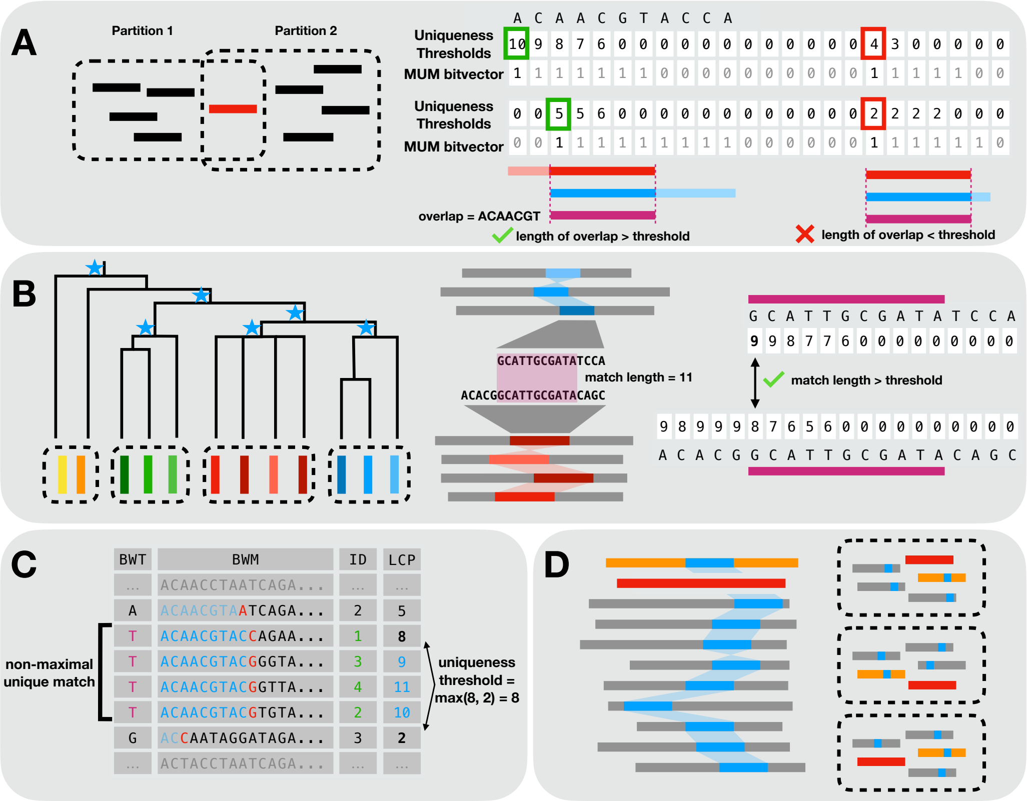
**Supplementary Material** *Software (Source Code)*: `https://github.com/vikshiv/mumemto` [5]

**Funding** *Vikram S. Shivakumar*: R35GM139602 and R56HG013865.
*Ben Langmead*: R35GM139602 and R56HG013865.

**Acknowledgements** The authors would like to thank Christina Boucher and Travis Gagie for algorithmic discussions, and Kuan-Hao Chao for valuable discussions on human genome annotations.

**Table 1** Runtime and memory usage comparison between serial and parallel multi-MUM computation across different partition schemes. We report execution time (in hours) and peak memory usage (in GB) for two datasets, chr19 haplotypes from the Human Pangenome Reference Consortium (HPRC) [4] and the *A. thaliana* pangenome [3]. The lowest peak memory footprint and the fastest running times are bolded.

| Dataset | | | Serial | | Parallel | |
|---|---|---|---|---|---|---|
| | Partitions | Seqs per | Time (hrs) | Memory (GB) | Time (hrs) | Memory (GB) |
| chr19 HPRC ($N = 474$) | 1 | 474 | 4.77 | 44.05 | – | – |
| | 5 | 96 | 5.00 | 13.96 | 1.11 | 66.78 |
| | 10 | 48 | 5.91 | 9.56 | 0.63 | 88.63 |
| | 20 | 24 | 7.21 | 7.16 | 0.39 | 125.29 |
| | 40 | 12 | 8.32 | **5.35** | **0.26** | 190.39 |
| *A. thaliana* ($N = 69$) | 1 | 69 | 2.58 | 70.34 | – | – |
| | 5 | 15 | 3.75 | 27.14 | 0.80 | 128.08 |
| | 10 | 7 | 4.92 | 18.79 | 0.57 | 172.00 |
| | 20 | 4 | 6.48 | **13.58** | **0.40** | 242.36 |



| BWT | BWM | ID | LCP |
|---|---|---|---|
| ... | ACAACCTAATCAGA... | ... | ... |
| A | ACAACGTAATCAGA... | 2 | 5 |
| T | ACAACGTACCAGAA... | 1 | 8 |
| T | ACAACGTACGGGTA... | 3 | 9 |
| T | ACAACGTACGGTTA... | 4 | 11 |
| T | ACAACGTACGTGTA... | 2 | 10 |
| G | ACCAATAGGATAGA... | 3 | 2 |
| ... | ACTACCTAATCAGA... | ... | ... |

**Figure 1** (**A**) Anchor-based merging requires a common sequence (red) present in each partition. Multi-MUMs are merged by identifying overlaps between partition-specific matches in the anchor coordinate space, and a uniqueness threshold determines if a MUM is still unique in each partition after truncation. (**B**) String-based merging enables computation of multi-MUMs between partitions without a common sequence. An example tree (left) is shown, highlighting the use case where partial multi-MUMs specific to internal nodes (starred) can be computed by merging subclade-based partitions up a tree. (right) MUM overlaps are computed by running Mumemto on the MUM sequences, and the uniqueness threshold array ensures overlaps remain unique across the merged dataset. (**C**) An example Burrows-Wheeler Transform (BWT), matrix (BWM), and Longest Common Prefix (LCP) array, with sequence IDs for each suffix shown (ID). A non-maximal unique match (UM) is shown, and the uniqueness threshold for this match is found using the flanking LCP values. (**D**) A partial multi-MUM (in blue) is found in all-but-one sequence (excluded in red). Using two anchor sequences (red and orange), all-but-one partial MUMs can be computed using an augmented anchor-based merging method.

## References

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

2 Mitchell Cummins, Cadel Watson, Richard J Edwards, and John S Mattick. The evolution of ultraconserved elements in vertebrates. *Molecular biology and evolution*, 41(7):msae146, 2024.

3 Qichao Lian, Bruno Huettel, Birgit Walkemeier, Baptiste Mayjonade, Céline Lopez-Roques, Lisa Gil, Fabrice Roux, Korbinian Schneeberger, and Raphael Mercier. A pan-genome of 69 Arabidopsis thaliana accessions reveals a conserved genome structure throughout the global species range. *Nature Genetics*, pages 1–10, 2024.

4 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.

5 Vikram Shivakumar. vikshiv/mumemto. Software, version v1.3.0. (visited on 2025-06-27). URL: `https://github.com/vikshiv/mumemto`.

6 Vikram S Shivakumar and Ben Langmead. Mumemto: efficient maximal matching across pangenomes. *bioRxiv*, pages 2025–01, 2025.