ChromMiniGraph: Space-Efficient Minimizer-based Pangenome Reference Graph and Haplotype Mapping Tool

Jim Shaw[*1], **K. Jun Gao**[*2,5], Jared T. Simpson[4,5], Yun William Yu[1,3]

Corresponding author: kgao@cs.toronto.edu / jgao@oicr.on.ca * These authors contributed equally to this work

[1] Department of Mathematics, University of Toronto | [2] Department of Computer Science, University of Toronto | [3] Department of Computer and Mathematical Sciences, University of Toronto at Scarborough | [4] Department of Molecular Genetics, University of Toronto | [5] Ontario Institute for Cancer Research

Introduction

ChromMiniGraph (CMG) creates a chromatic minimizer graph: a directed acyclic graph where nodes are minimizers with colors representing haplotypes.

CMG does

- create a pangenome reference
- map reads to the pangenome reference and assign haplotype(s) to reads
- perform base-level alignment for the best haplotype and output BAM files

CMG does *not*

- map read to a reference graph generated using other tools (e.g. minigraph, vg)
- compress the reference sequences or the graph

CMG is

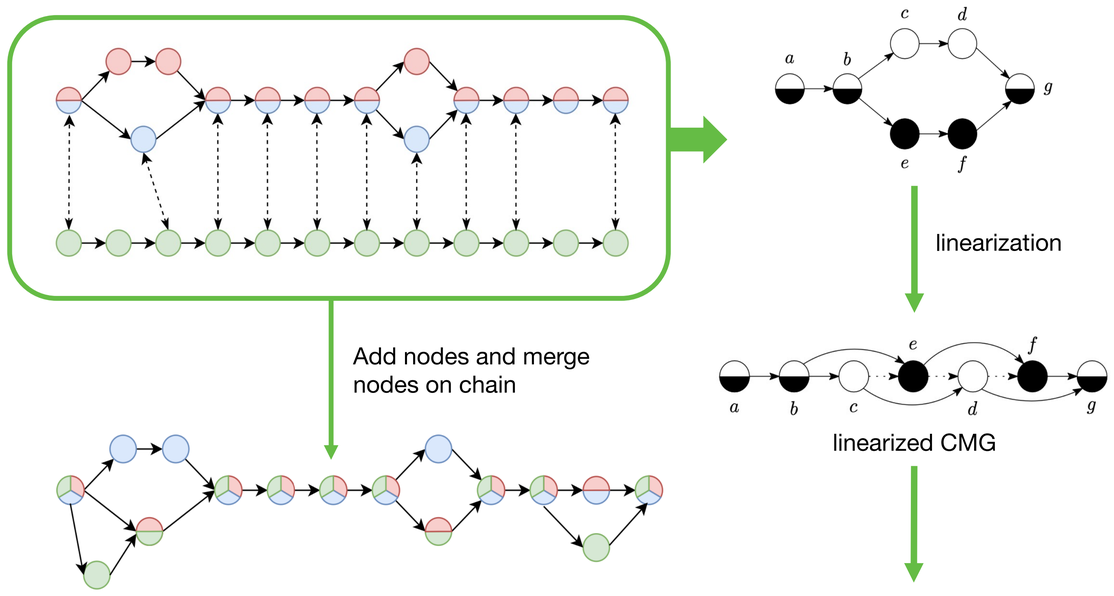
- an efficient graph index
  Use **minimizers;** Serializable graph format
- relatively fast to create
  Co-linear chaining and **RMQ**
- able to map reads efficiently and accurately
  Co-linear chaining with **banded DP**
  Good **linearization algorithm**

CMG is *not*

- minimizer de Bruijn graph: CMG don't collapse identical k-mers
- minigraph: CMG stores individual k-mers with subsampled reference positions, not reference sequences or unitig paths

Reference to Graph Chaining

The graph is constructed through **iterative reference-to-graph chaining**. At each iteration, we chain a new reference sequence onto the current graph and merge k-mer nodes to create multi-colored nodes if necessary. Reference-to-graph chaining uses a linear cost function and can be solved efficiently using RMQ.



Read to Graph Chaining

Read mapping is done using banded dynamic programming similar to minimap2 and uses the topologically linearized coordinate for the gap function, instead of the more costly DAG chaining or partial order alignment. We also check for color consistency during chaining.

$$f'(j) = \max \left\{ \max_{\substack{1 \le i < j \\ L(a_i) < L(a_j) \\ b_i < b_j}} \{f'(i) + A' - g(i,j)\}, 0 \right\}$$

use linearized coord

$$g(i,j) = \begin{cases} B'|(L(x_j) - L(x_i)) - (y_j - y_i)| & \text{if } x_i \preceq x_j \\ \infty & \text{otherwise} \end{cases}$$

ensures path coherence

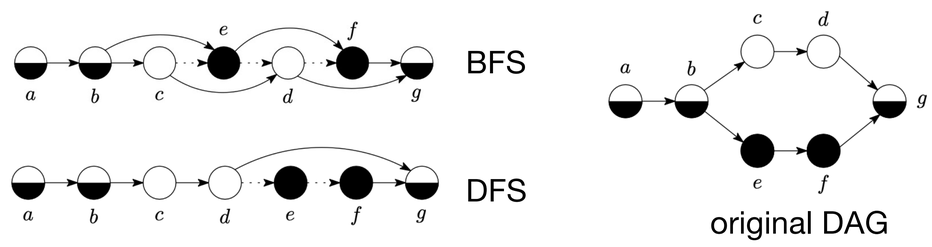
Topological Linearization

- **Spatiality**: nucleotides/k-mers that are physically close together within a genome should have similar coordinates
- **Monotonicity**: the genome graph coordinates of successive nucleotides within a genome should be increasing

Classical graph algorithms for generating topological linearization:

DFS: finish all vertices in current branch before exploring next branch
BFS: explore each branch before moving to next vertex

Conclusion: BFS is more desirable because it interleaves vertices to prevent large insertion.



Alternative Distance and Bubble Detection

Topological linearization fails to provide a reasonable estimate of the graph topology around regions with **unbalanced superbubbles**. To remedy this issue, we first identify unbalanced superbubbles and then score the chain using sampled reference positions at neighboring vertices.

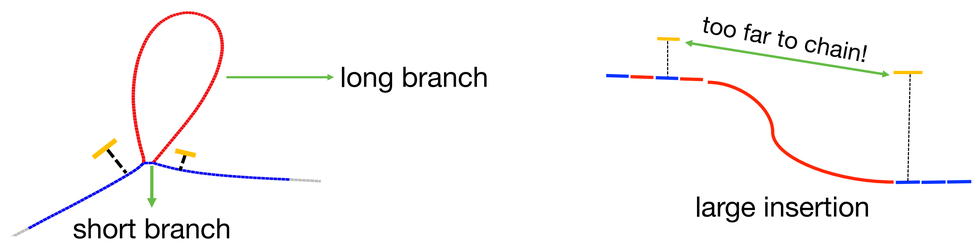


Figure: Chaining on topologically linearized coordinate incorrectly penalizes anchors that straddle an unbalanced bubble

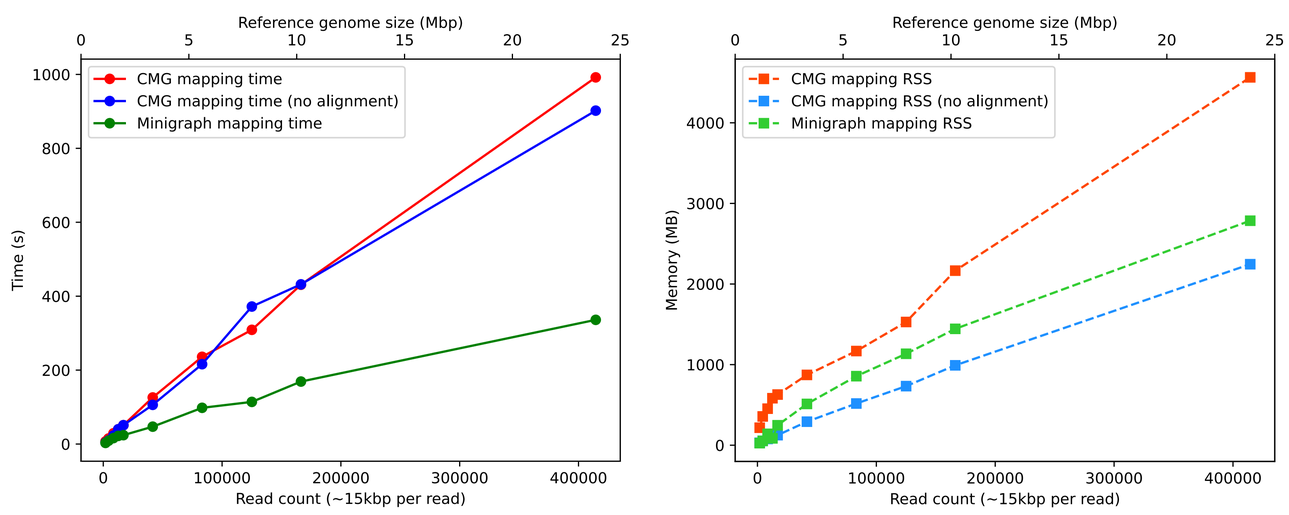
Results

We use ChromMiniGraph to construct graphs for human chromosomes 11 (2.3GB) and 20 (1.1GB) using 16 reference genomes. The serialized file is a binary file that can be read directly by ChromMiniGraph for fast read mapping. The tool also creates a much smaller **simplified graph** which stores the graph as a list of bubbles where unitigs are merged into continuous paths. It is useful for visualization and conversion to GFA format.

Chromosome	# references	Generation time (min)	Peak RSS (GB)	Serialized file size (GB)	Index only, compressed (MB)
Chr 20 (~65 Mbp)	16	51	9.15	1.2 (29MB simplified)	212
Chr 11 (~135 Mbp)	16	104	14.73	2.3 (60MB simplified)	415

* Experiments performed on HPC clusters running Ubuntu 20.04.5 LTS. Read mappings using CMG and minigraph are performed using 10 threads

Fully indexed graph files take about the same storage space as the starting references but can support efficient read mapping.



We simulated **10 sets of references** of different lengths, each consisting of **10 haplotypes**. We then generated long reads with **read depth 25x** using Badread with the error profile comparable to that of Nanopore R10.4.1. For each reference length, time and memory consumption (residence set size) for mapping all reads to the corresponding reference graph are measured and reported in the figure above.

We also evaluated the accuracy and quality of the mapping. This is measured by calculating the percentage of reads that are correctly and unambiguously assigned to the individual reference from which the reads are sequenced. The average accuracy for long reads (~15 kbps) is consistently above 90% (i.e. 90% of the reads are assigned a *correct* and *unique* haplotype).

References

H. Li, "Minimap2: pairwise alignment for nucleotide sequences," *Bioinformatics*, vol. 34, pp. 3094–3100, Sept. 2018.

C. Lee, C. Grasso, and M. F. Sharlow, "Multiple sequence alignment using partial order graphs," *Bioinformatics*, vol. 18, pp. 452–464, Mar. 2002.

J. Ma, M. Cáceres, L. Salmela, V. Mäkinen, and A. I. Tomescu, "GraphChainer: Co- linear Chaining for Accurate Alignment of Long Reads to Variation Graphs," preprint, *Bioinformatics*, Jan. 2022.

F. Gärtner and P. F. Stadler, "Direct Superbubble Detection," *Algorithms*, vol. 12, no. 4, p. 81, Apr. 2019

R. Wick. Badread: simulation of error-prone long reads. *Journal of Open Source Software*, vol. 4, no. 36, pp. 1316, Apr. 2019

gaojunxuan/
chrom_mini_graph