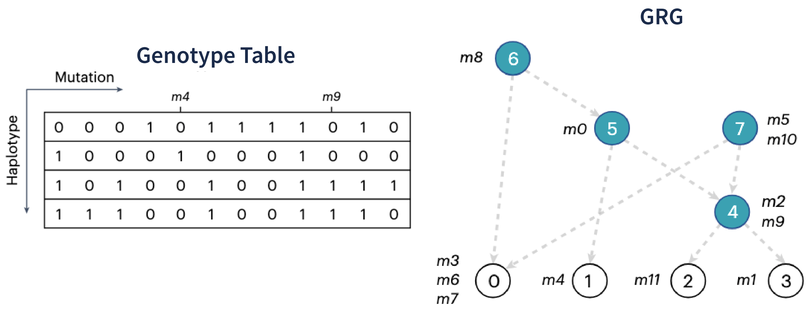
Faster Biobank-Scale Genomics through Sparse Linear Algebra on Genotype Representation Graph

Yifan Li, Qingyao Sun, Drew DeHaas, April Wei, Giulia Guidi (advisor)
Cornell University

Background

Whole-genome sequencing generates **massive genetic datasets**.
- All of Us v8 dataset has 410K individuals × 870M variants.
- Typically require **tens of TBs** of space.
- Computation is inefficient even when exploiting sparsity.

Genotype Representation Graph (GRG) addressed this by encoding shared ancestral structure in a multi-tree, replacing explicit matrix-vector multiplication with graph traversal, and achieving up to 25× space reduction comparing to gzipped formats.



Example of a genomic dataset in tabular format (left) and the correspnding GRG (right) [1].

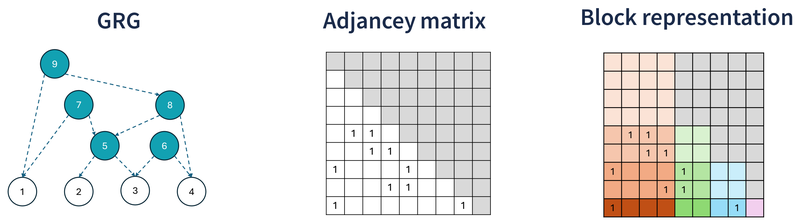
Motivation

GRG compresses biobanks and significantly reduces the amount of computation needed. However, existing GRG traversal algorithms are **single-threaded**.
No simple solution exists for **parallelizing the multi-tree traversal** problem, leaving performance bottlenecked.

Acknowledgement: This research used resources of NERSC, using NERSC award ASCR-ERCAP0030076. The material is based upon work supported by the National Science Foundation under Grant IIS-2435801.
This work is done with the support of Max Xiaohang Zhao, Adam R. Boyko, and Shaila A. Musharoff.
Reference: [1] DeHaas, D., Pan, Z. & Wei, X. Enabling efficient analysis of biobank-scale data with genotype representation graphs. Nat Comput Sci 5, 112–124 (2024).
[2] DeHaas, D., Adonizio, C., Pan, Z. & Wei, X. General, orders-of-magnitude faster whole-genome analysis with genotype representation graphs.

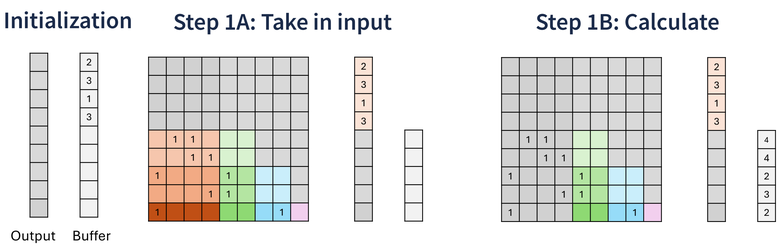
Methods

We reformulate multi-tree traversal as a **Sparse Triangular Solve (SpTRSV).**
- SpTrSV satisfies the exact dependency for multi-tree traversal.
- This enables the use of optimized kernels, such as cuSparse.



Example of a GRG G (left), its adjacency matrix representation A (middle), and height-group block representation (right).

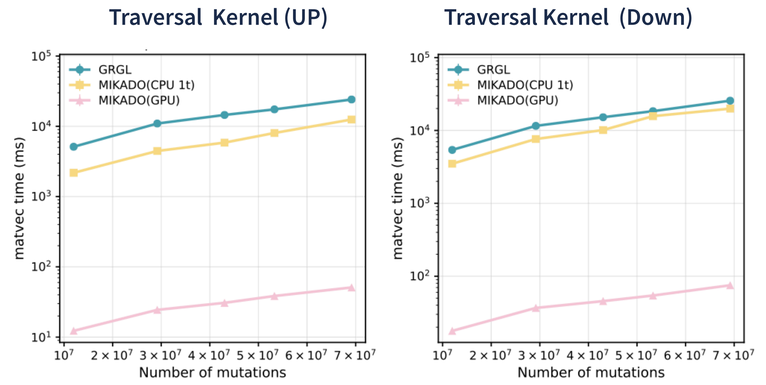
We further decompose the SpTRSV into **a height-ordered pipeline of Sparse Matrix-Vector multiplications** (SpMVs):
- Matrix representation of GRG has a block structure.
- SpMV primitives are widely implemented across hardware platforms.
- SpMV primitives are heavily optimized.



The SpTRSV is transformed into a sequence of SpMVs with the block formulation. Solving procedure of height-1 block is shown.

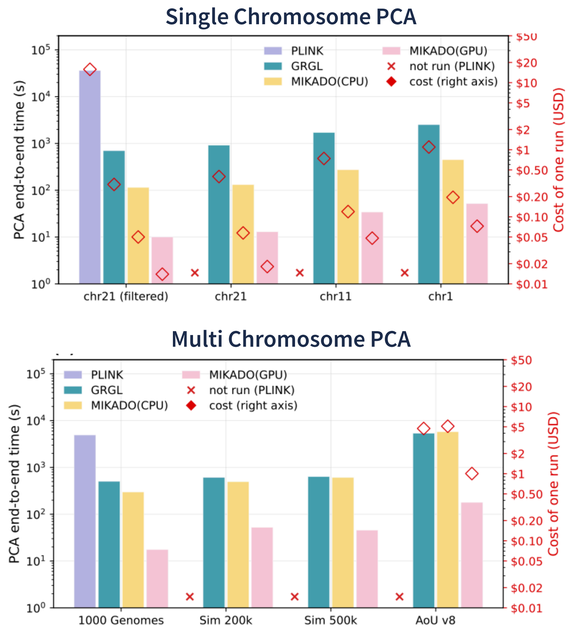
This reformulation of GRG traversal is **performant and portable** on different kind of hardware.
- Intel MKL for better CPU performance.
- cuSparse with streams and CUDA graph for GPU.
- Possibilities on novel architecture (Cerebras...)

Results



Kernel runtime for up and direction traversal on five chromosomes in the All of Us v8 dataset.

We conduct experiments on multiple datasets..
For a single traversal kernel, one A100 80GB SXM GPU can achieve **more than 400× speedup**. For end-to-end applications such as Principle Component Analysis (PCA), GPU is at least **20× faster** than CPU baseline.



PCA end-to-end runtime and cost. Experiments on AoU dataset are run on Researcher Workbench; all other experiments are run on Perlmutter Supercomputer (GPU nodes).

Conclusion Our reformulation of GRG traversal enables the use of highly optimized SpMV kernels on heterogeneous hardware. Experiments show a 20× end-to-end speedup for PCA, with larger gains expected for more compute-intensive applications. This substantially reduces the time and cost of biobank-scale computation and facilitates the development of novel algorithms.

Preprint on biorxiv