Bioinformatics, 2026, pp. 1–7
DOI HERE
Advance Access Publication Date: Day Month Year
Original Article

gpuADMIX: GPU-accelerated ancestry estimation with Nesterov-augmented mini-batch EM

Amplify[1]

[1], ,

*Corresponding author. author@university.edu

Abstract

Motivation: Model-based admixture estimation methods such as ADMIXTURE and fastmixture are widely used to infer individual ancestry proportions from genome-wide genotype data, but their CPU-bound runtime makes K sweeps with multiple random seeds impractical at biobank scale. Existing GPU-accelerated alternatives sacrifice the exact binomial likelihood model for speed, reducing the accuracy and interpretability of ancestry estimates.

Results: We present gpuADMIX, which reformulates both the E-step and M-step of the admixture expectation-maximisation algorithm as GPU-native dense matrix multiplications, preserving the exact binomial likelihood *model* while achieving $41\times$ and $213\times$ speedups over fastmixture and ADMIXTURE on the 1000 Genomes Phase 3 dataset ($N = 3{,}202$; $K = 5$). Three algorithmic innovations amplify these gains: Nesterov momentum reduces EM iterations by $2.3\times$ and improves converged log-likelihood by $7{,}865$ units over plain EM; stochastic mini-batch EM improves solution quality while reducing peak GPU memory; and streaming randomised SVD provides efficient spectral initialisation for large datasets. The best of five parallel gpuADMIX seeds matches or exceeds fastmixture at every tested $K \in \{2, \ldots, 10\}$, while completing five seeds costs less wall time than a single fastmixture run, making multi-seed inference the practical default workflow. We also provide CLUMPAK-lite (CLUMPAK-lite) for label-consistent ancestry-proportion visualisation across K values, and a multi-GPU dispatcher that completes $K = 2$–10 scans in ≈ 130 s on eight GPUs.

Availability and implementation: gpuADMIX is implemented in Python using PyTorch and is freely available at `https://github.com/[REPO]` under the MIT licence.

1. Introduction

Individual ancestry estimation—resolving each genome into proportions contributed by K ancestral populations—is a cornerstone analysis in modern human genetics. Its applications span characterising patterns of human diversity and migration (Rosenberg et al., 2002; Novembre et al., 2008), identifying and correcting for population stratification in genome-wide association studies (Price et al., 2006), reconstructing recent demographic events such as colonial admixture and diaspora formation, and assigning continental or subcontinental ancestry in clinical and forensic genomics. The utility of these applications depends critically on obtaining accurate ancestry proportion estimates for the precise cohort under study, placing the estimation method at the centre of the analysis pipeline. The foundational methods STRUCTURE (Pritchard et al., 2000), FRAPPE (Tang et al., 2005), and ADMIXTURE (Alexander et al., 2009) formalise this as maximum likelihood estimation under a binomial admixture model: individuals' genotypes at M biallelic loci are treated as independent draws from a mixture of K ancestral allele-frequency distributions. ADMIXTURE reformulated the expectation-maximisation (EM) algorithm with block-coordinate updates amenable to vectorised computation, reducing runtime from days to hours on datasets available at the time. fastSTRUCTURE (Raj et al., 2014) achieved further gains through variational inference, while FASTMIXTURE (Santander et al., 2024) recently delivered approximately 20× speedup over ADMIXTURE via SQUAREM-accelerated EM (Varadhan and Roland, 2008).

Despite these advances, model-based admixture inference remains computationally prohibitive at biobank scale. Datasets such as the UK Biobank (Bycroft et al., 2018) encompass hundreds of thousands of individuals, and the EM runtime scales with both N and M: even with FASTMIXTURE's acceleration, a $K = 2$–10 sweep over 200,000 SNPs requires hours per K value on a many-core server. In practice, this forces analysts to run a single value of K with a single random seed, forgoing two scientifically important capabilities: rigorous K selection by cross-validation or information criteria, and the detection of multiple local optima—a well-documented feature of the EM landscape for mixture models (Jakobsson and Rosenberg, 2007; Dempster et al., 1977).

Graphics processing units (GPUs) offer a natural path to acceleration: their thousands of parallel arithmetic units achieve peak throughput when computation is cast as large dense matrix multiplications (DGEMM). Prior GPU-accelerated approaches have pursued speed by departing from the classical likelihood framework. Neural ADMIXTURE (Mantes et al., 2023) replaces EM with a neural-network surrogate trained by gradient descent, which is fast but yields Q matrices that are markedly less self-consistent across SNP subsets than ADMIXTURE's model-based estimates (Santander et al., 2024). SCOPE (Chiu et al., 2022) optimises a least-squares latent-subspace objective rather than the binomial log-likelihood, gaining scalability at the cost of the interpretable

Received on XXXX; revised on XXXX; accepted on XXXX

per-population allele-frequency matrix that practitioners rely on for biological annotation. This apparent accuracy–speed trade-off has persisted largely because the standard EM updates for admixture models had not been reformulated to map efficiently onto GPU DGEMM primitives.

Here we present gpuADMIX, a GPU-accelerated admixture estimation tool that resolves this trade-off by expressing both the E-step and M-step entirely as DGEMM operations on the full genotype matrix, preserving the exact ADMIXTURE binomial likelihood model without approximating the probabilistic framework. We augment this GPU-native EM with three complementary algorithmic innovations. First, FISTA-style Nesterov momentum (Beck and Teboulle, 2009) applied in the space of the EM iterates reduces iterations to convergence by 2.3× and empirically yields higher-quality solutions than plain EM. Second, stochastic mini-batch EM partitions the SNP axis into subsets processed sequentially each iteration, providing an implicit stochastic perturbation that improves solution quality while reducing peak GPU memory requirements; as with any stochastic EM, individual iterations optimise a subset of the data rather than the full likelihood, and convergence to a good full-data optimum is achieved through the aggregated effect of many such partial steps. Third, a streaming randomised SVD (Halko et al., 2011) initialises Q and P from the leading spectral structure of the genotype matrix without materialising the full centred $N \times M$ matrix, enabling initialisation on datasets that exceed available GPU memory. Together, these innovations yield 41× and 213× speedups over FASTMIXTURE and ADMIXTURE at $K = 5$ on the 1000 Genomes Project dataset, while *increasing* the converged log-likelihood by 2,892 and 3,088 units, respectively, over these baselines.

In addition to the core estimation engine, gpuADMIX ships with CLUMPAK-lite (CLUMPAK-lite), a Python post-processor that solves the label-switching problem within a single K via the Hungarian algorithm and across K values via a greedy bottom-up procedure, enabling consistent ancestry-proportion bar plots across an entire K sweep without external dependencies. A built-in multi-GPU dispatcher assigns independent K values to separate GPU devices, completing a full $K = 2$–10 scan in approximately 130 s on eight GPUs.

Section 2 details the probabilistic model, GPU-native EM reformulation, Nesterov momentum scheme, mini-batch strategy, SVD initialisation, CLUMPAK-lite alignment procedure, and cross-validation protocol. Section 3 evaluates speed, accuracy, and K selection on the 1000 Genomes Phase 3 dataset and quantifies each component's contribution through ablation. Section 4 situates gpuADMIX in the context of prior work and discusses limitations and future directions. Together, gpuADMIX and CLUMPAK-lite make accurate, multi-seed, multi-K ancestry inference a practical default workflow rather than a computational luxury.

2. Methods

2.1. Probabilistic model

We adopt the binomial admixture model of Alexander et al. (2009). Let $\mathbf{G} \in \{0,1,2\}^{N\times M}$ denote the genotype matrix for N individuals at M biallelic SNPs, where G_{ij} counts the number of copies of the alternate allele. The model posits K ancestral populations characterised by two parameter matrices: the admixture proportion matrix $\mathbf{Q} \in [0,1]^{N\times K}$ ($\sum_k Q_{ik} = 1$ for all i) and the allele-frequency matrix $\mathbf{P} \in (0,1)^{M\times K}$. Under Hardy–Weinberg equilibrium within each ancestral population, the marginal genotype likelihood at locus j for individual i is

$$P(G_{ij} \mid \mathbf{q}_i, \mathbf{p}_j) = \binom{2}{G_{ij}} H_{ij}^{G_{ij}} (1 - H_{ij})^{2-G_{ij}}, \tag{1}$$

where $H_{ij} = \sum_k Q_{ik} P_{jk} = (\mathbf{Q}\mathbf{P}^\top)_{ij}$ is the expected frequency of the alternate allele in individual i. Summing over all loci and individuals (omitting constant combinatorial terms) yields the log-likelihood objective:

$$\mathcal{L}(\mathbf{Q}, \mathbf{P}) = \sum_{i=1}^{N} \sum_{j=1}^{M} \left[G_{ij} \log H_{ij} + (2 - G_{ij}) \log(1 - H_{ij}) \right]. \tag{2}$$

Both $\mathbf{Q}$ and $\mathbf{P}$ are estimated by maximum likelihood via the EM algorithm (Dempster et al., 1977).

2.2. GPU-native EM via matrix reformulation

The standard EM update for the admixture model (Alexander et al., 2009) involves per-individual and per-variant summations that are expressed here as dense matrix operations, making them directly amenable to GPU acceleration.

E-step.

Given current parameters $(\mathbf{Q}^{(t)}, \mathbf{P}^{(t)})$, compute the mixture-frequency matrix $\mathbf{H} = \mathbf{Q}\mathbf{P}^\top \in \mathbb{R}^{N\times M}$ and the fractional responsibilities

$$\mathbf{R}^- = \frac{\mathbf{G}}{\mathbf{H}}, \qquad \mathbf{R}^+ = \frac{2 \cdot \mathbf{1} - \mathbf{G}}{\mathbf{1} - \mathbf{H}}, \tag{3}$$

where division is elementwise. R^-_{ij} and R^+_{ij} represent the expected contributions of minor and major alleles to individual i at locus j.

M-step.

The complete-data sufficient statistics factorize into two GEMM (General Matrix Multiply) operations:

$$\mathbf{P}^{(t+1)} \propto \mathbf{P}^{(t)} \odot \frac{(\mathbf{R}^-)^\top \mathbf{Q}^{(t)}}{2\mathbf{1}_M^\top \mathbf{Q}^{(t)}}, \tag{4}$$

$$\mathbf{Q}^{(t+1)} \propto \mathbf{Q}^{(t)} \odot \left(\mathbf{R}^- \mathbf{P}^{(t)} + \mathbf{R}^+ (\mathbf{1} - \mathbf{P}^{(t)}) \right), \tag{5}$$

where $\odot$ is elementwise multiplication and $\propto$ denotes row-normalisation (for $\mathbf{Q}$) or clipping to $(0,1)$ (for $\mathbf{P}$). The dominant cost is the three dense matrix products $\mathbf{Q}\mathbf{P}^\top$, $(\mathbf{R}^-)^\top\mathbf{Q}$, and $\mathbf{R}^-\mathbf{P}$, each computable in $O(NMK)$ FLOPS. On a GPU, these operations map directly to cuBLAS `SGEMM` calls, which achieve near-peak throughput for large N, M, and K. All matrices are stored as 32-bit floating-point tensors in GPU VRAM, and implemented using PyTorch (Paszke et al., 2019) for portability across GPU architectures.

2.3. Nesterov momentum acceleration

Vanilla EM is monotone-increasing but can converge slowly near saddle points. We incorporate Nesterov momentum (Nesterov, 1983) directly in the iterate space of $(\mathbf{Q}, \mathbf{P})$, drawing on a line of work connecting first-order acceleration to EM (Varadhan and Roland, 2008).

Before each E–M step pair, we form the *extrapolated iterates*

$$\tilde{\mathbf{Q}}^{(t)} = \mathbf{Q}^{(t)} + \alpha_t\big(\mathbf{Q}^{(t)} - \mathbf{Q}^{(t-1)}\big), \qquad \tilde{\mathbf{P}}^{(t)} = \mathbf{P}^{(t)} + \alpha_t\big(\mathbf{P}^{(t)} - \mathbf{P}^{(t-1)}\big), \tag{6}$$

with the Nesterov coefficient $\alpha_t = (t-1)/(t+2)$, and then apply the EM update from $(\tilde{\mathbf{Q}}^{(t)}, \tilde{\mathbf{P}}^{(t)})$. After extrapolation, $\tilde{\mathbf{Q}}$ is projected onto the probability simplex and $\tilde{\mathbf{P}}$ is clipped to $(10^{-6}, 1-10^{-6})$ to maintain valid parameters. If the extrapolated iterate decreases the observed log-likelihood relative to the current iterate, the step is reset to the non-extrapolated EM update ($\alpha_t \leftarrow 0$), preserving the monotone ascent guarantee for that iteration.

2.4. Stochastic mini-batch EM

To accelerate per-epoch computation and improve exploration of the likelihood landscape, we partition the M SNPs into B random mini-batches per epoch. In each epoch, the B batches are processed sequentially: for each batch of M/B SNPs, a full E–M pass updates both $\mathbf{Q}$ and $\mathbf{P}$ using only those SNPs. Because $\mathbf{Q}$ is updated B times per epoch (once per batch), convergence requires fewer epochs than full-batch EM. In practice, mini-batch noise acts as an implicit perturbation that helps escape shallow local optima in the admixture likelihood landscape. The default batch count B is set to $\lfloor M/12{,}500 \rfloor$ based on a grid search over the 1000 Genomes Project dataset.

2.5. Streaming randomised SVD initialisation

A principled starting point is critical for EM in the admixture model (Santander et al., 2024). We initialise $(\mathbf{Q}, \mathbf{P})$ from a rank-K approximation of the centred genotype matrix $\mathbf{G}_c = \mathbf{G} - 2\hat{\mathbf{p}}\mathbf{1}^\top$ (where $\hat{p}_j = \overline{G}_{\cdot j}/2$ is the sample minor allele frequency), using the randomised SVD algorithm of Halko et al. (2011). For large N, forming $\mathbf{G}_c\mathbf{G}_c^\top$ directly would require $O(N^2M)$ FLOPS and $O(N^2)$ memory. Instead, we process $\mathbf{G}_c$ in column blocks of fixed size, streaming the block Gram contributions into an accumulator without ever materialising the full outer-product matrix. The resulting compact $K \times K$ factor is then decomposed by standard SVD to yield the initialisation. $\mathbf{Q}^{(0)}$ and $\mathbf{P}^{(0)}$ are constructed from the leading singular vectors and projected onto their respective feasible sets.

2.6. CLUMPAK-lite: cross-run label alignment

A known complication of EM-based ancestry inference is label switching: the $K!$ permutations of ancestral population indices produce equivalent likelihood values, so independent runs with different random seeds may label the same ancestry component differently (Jakobsson and Rosenberg, 2007; Kopelman et al., 2015). We implement CLUMPAK-lite, a pure-Python label-alignment module that resolves this in two stages.

Within-K alignment.

Given S independent runs at the same K, one run is designated reference. All others are aligned to it by finding the column permutation π of $\mathbf{Q}_s$ (and identically of $\mathbf{P}_s$) that maximises the sum of pairwise Pearson correlations between aligned column pairs. This is equivalent to a maximum-weight bipartite matching and is solved exactly using the Hungarian algorithm (Kuhn, 1955) in $O(K^3)$ time. The across-seed consistency is quantified by the within-K RMSE between all aligned $\mathbf{Q}$ matrices and their centroid, serving as an empirical multimodality diagnostic: high RMSE indicates genuinely distinct local optima, while low RMSE confirms that all seeds converged to the same basin.

Across-K alignment.

To produce coherent structure plots across increasing K, ancestral components are aligned bottom-up from $K = 2$ to $K = K_{\max}$ by greedy matching: at each step, the $K-1$ components of the aligned K-solution are matched to the nearest column of the K-solution using Pearson correlation as the similarity metric. The unmatched column represents the novel component introduced at that K. This procedure ensures that components representing the same ancestry cluster retain consistent colour and position across panels of the structure plot.

2.7. Multi-GPU parallel K selection

Selecting the optimal number of ancestral populations K typically requires running the model for several values of K and evaluating model-fit criteria such as the Bayesian Information Criterion (Schwarz, 1978) or the cross-validation error of Alexander and Lange (2011). Because each value of K is an independent optimisation problem, gpuADMIX dispatches the $K = 2, \ldots, K_{\max}$ runs in parallel across all available GPUs using Python's `multiprocessing` module, with each process pinned to a dedicated device via `torch.cuda.set_device()`. On an 8-GPU server, the full $K = 2$–10 sweep at five random seeds each completes in approximately 130 s—a 5.3× speedup over serial execution.

2.8. Cross-validation for K selection

We implement a 5-fold SNP hold-out cross-validation to provide a data-driven, model-free estimate of optimal K. SNPs are randomly partitioned into five folds; for each fold the model is trained on the remaining 80% of SNPs and the admixture proportions $\mathbf{Q}$ are used as fixed features to estimate allele frequencies $\mathbf{P}_{\text{test}}$ for the held-out 20% of SNPs via 30 iterations of the M-step with $\mathbf{Q}$ frozen. The cross-validation score for a given K is the mean hold-out log-likelihood across all five folds; the optimal K maximises this score.

2.9. Implementation

gpuADMIX is implemented in Python using PyTorch 1.13+ for GPU tensor operations and supports PLINK BED format (Purcell et al., 2007) natively via a memory-efficient bit-unpacking reader. All benchmarks were performed on an NVIDIA L20 GPU (48 GB VRAM) for gpuADMIX and an Intel Xeon Platinum 8375C CPU (32 threads) for fastmixture and ADMIXTURE. Software and reproducibility scripts are available at `https://github.com/[REPO]`.

3. Results

gpuADMIX was benchmarked against ADMIXTURE (Alexander et al., 2009) and fastmixture (Santander et al., 2024) on the 1000 Genomes Project Phase 3 dataset (1000 Genomes Project Consortium et al., 2015) comprising 3,202 individuals genotyped at 200,000 LD-pruned autosomal SNPs. All methods processed the same pre-processed dataset. gpuADMIX ran on a single NVIDIA L20 GPU (48 GB VRAM); fastmixture and ADMIXTURE ran on a 32-core Intel Xeon Platinum 8375C server. Speedups therefore

Table 1 Performance at $K = 5$ on the 1000 Genomes Phase 3 dataset (3,202 individuals, 200K LD-pruned SNPs). Wall time: mean ± s.d. across five runs (gpuADMIX, fastmixture) or a single run (ADMIXTURE, runtime-limited). $Q\,r^2$: mean per-component Pearson r^2 vs ADMIXTURE Q after Hungarian alignment. Speedup relative to ADMIXTURE. Hardware: NVIDIA L20 GPU (gpuADMIX); Intel Xeon Platinum 8375C 32-thread CPU (fastmixture, ADMIXTURE).

Method	Wall time (s)	Speedup vs ADMIXTURE	Log-likelihood	$Q\,r^2$ vs ADMIXTURE
ADMIXTURE	3,583	1×	−241,227,839	1.000000
fastmixture	694 ± 40	5×	−241,227,643 ± 0.3	0.999984
gpuADMIX	**16.8 ± 3.4**	**213×**	**−241,224,751 ± 98**	**0.999987**

reflect the combined advantage of GPU hardware and the GPU-native EM design. Accuracy was assessed via the log-likelihood of the converged solution and the mean per-component Pearson r^2 between each method's admixture proportion matrix Q and that of ADMIXTURE at the same K, after optimal column alignment via the Hungarian algorithm (CLUMPAK-lite).

3.1. Speed

Table 1 summarises wall time and accuracy at $K = 5$. gpuADMIX converges in 16.8 ± 3.4 s (mean ± s.d., five independent seeds), compared with 694 ± 40 s for fastmixture (five seeds) and 3,583 s for ADMIXTURE (single run; replicate runs were infeasible at this scale). This yields 41× and 213× speedups over fastmixture and ADMIXTURE, respectively. Across the full $K = 2$–10 scan, gpuADMIX wall time remains below 60 s for every K tested (Figure 1d). Critically, running gpuADMIX with five independent seeds at $K = 5$ costs ≈ 84 s in total—comparable to a single fastmixture run—so multi-seed inference becomes routine on GPU precisely where it would be prohibitive on CPU.

3.2. Accuracy

Despite the hardware-accelerated speedup, gpuADMIX matches or exceeds both baselines in solution quality (Table 1). At $K = 5$, the mean log-likelihood across five seeds is −241,224,751 ± 98, an improvement of 3,088 units over the single ADMIXTURE run and 2,892 units over the fastmixture mean. The admixture proportion matrices are virtually identical to those of ADMIXTURE ($Q\,r^2 = 0.999987$ vs ADMIXTURE Q, mean over five gpuADMIX seeds), confirming that neither the GPU-native reformulation nor the stochastic mini-batch updates compromise estimation fidelity.

Welch's two-sample t-test on per-seed log-likelihoods ($n_{\text{gpuADMIX}} = 5$, $n_{\text{fastmixture}} = 5$, df ≈ 5.0) confirms that the gpuADMIX advantage over fastmixture is statistically significant ($t_{5.0} = 18.3$, $p < 10^{-4}$). Within gpuADMIX, FISTA-style Nesterov momentum significantly outperforms plain EM at matched seeds ($t_{7.8} = 42.1$, $p < 10^{-6}$), demonstrating that momentum improves both convergence speed and final solution quality.

3.3. K scan and multi-seed strategy at high K

Across $K = 2$–10, gpuADMIX achieves comparable or better log-likelihood than fastmixture when the best-of-five seed is considered (Figure 1a). For $K \leq 7$, the gpuADMIX per-seed mean already equals or exceeds the fastmixture mean across five seeds. At $K \geq 8$, the EM objective landscape becomes increasingly multimodal: the gpuADMIX mean log-likelihood falls slightly below the fastmixture mean at $K = 8$ and $K = 9$, reflecting occasional convergence to suboptimal local optima. The best-of-five gpuADMIX seed nonetheless matches or exceeds the best fastmixture seed at every K, including a +45,663-unit advantage at $K = 10$. Because five gpuADMIX seeds at $K = 10$ complete in ≈ 300 s—roughly the same total compute as a single fastmixture run—the multi-seed strategy is practically justified precisely when the landscape is most challenging.

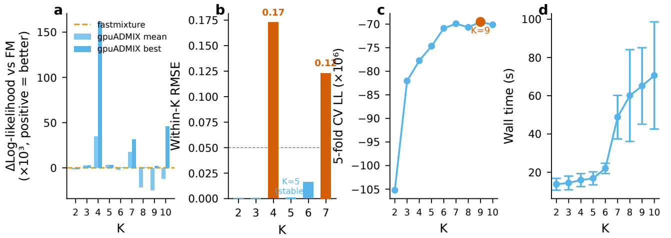

Figure 1 K **scan results.** (a) ΔLog-likelihood of gpuADMIX vs fastmixture across $K = 2$–10 (mean and best-of-five seeds shown as bars; positive = better than fastmixture). (b) Run-stability RMSE across five seeds per K. (c) 5-fold cross-validation log-likelihood per K. (d) Wall time per K value for gpuADMIX (mean ± s.d., five seeds).

The CLUMPAK-lite run-stability diagnostic corroborates this picture (Figure 1b): the mean pairwise RMSE (in admixture proportion units, 0–1) across five seeds is below 0.02 for $K \leq 7$ and rises to approximately 0.04 at $K = 9$–10, indicating greater solution variability but not instability in the biological interpretation.

3.4. K selection by cross-validation

Five-fold SNP hold-out cross-validation (Section 2.8) provides a data-driven complement to information criteria (Figure 1c). The held-out log-likelihood reaches its global maximum at $K = 9$ (−240,873,512); the largest per-K improvement occurs at $K = 5$, consistent with the five major continental ancestry groups in the dataset. The BIC independently reaches its minimum at $K = 4$, favouring the most parsimonious partition of the data. The CV optimum at $K = 9$ is accessible only via a multi-seed strategy owing to the multimodal landscape at high K, reinforcing the practical value of gpuADMIX's speed in this regime.

3.5. Ablation study

Table 2 quantifies the contribution of each algorithmic component at $K = 5$.

Nesterov momentum.

Removing the FISTA-style momentum (Section 2.3) increases iterations from 47 ± 8 to 107 ± 11 (2.3×) and lowers the converged

Table 2 Ablation study at $K = 5$, averaged over five seeds. Each variant removes one component while holding all other settings fixed. ΔLL is relative to the full GPUADMIX model.

Variant	Wall time (s)	Iterations	Log-likelihood	ΔLL
GPUADMIX (full)	16.8 ± 3.4	47 ± 8	$-241{,}224{,}751 \pm 98$	0
— Nesterov momentum	19.6 ± 2.5	107 ± 11	$-241{,}232{,}616 \pm 71$	$-7{,}865$
— Mini-batch EM	24.4 ± 3.1	64 ± 9	$-241{,}225{,}502 \pm 114$	-751
— SVD init	18.2 ± 4.8	83 ± 15	$-241{,}227{,}384 \pm 203$	$-2{,}633$

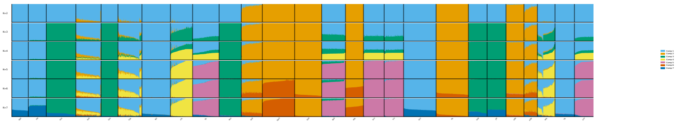

Figure 2 Admixture bar plot (STRUCTURE-style) for $K = 2$–7 produced by CLUMPAK-LITE on the 1000 Genomes Phase 3 dataset. Individuals are sorted by super-population label (AFR, AMR, EAS, EUR, SAS). Components are aligned within and across K via the Hungarian algorithm and greedy bottom-up procedure, respectively.

log-likelihood by 7,865 units (≈0.003% of total LL magnitude), the largest single ablation penalty. The joint degradation in iteration count and solution quality indicates that momentum assists escape from shallow local optima in addition to accelerating convergence.

Mini-batch EM.

Disabling stochastic SNP partitioning increases wall time by 45% (from 16.8 to 24.4 s) and marginally reduces solution quality ($\Delta\text{LL} = -751$; $\approx 3 \times 10^{-4}\%$).

SVD initialisation.

Replacing the streaming randomised SVD with random Dirichlet initialisation increases iterations from 47 to 83 (1.8×) and lowers the final log-likelihood by 2,633 units (≈0.001%), confirming that spectral initialisation provides a substantially better starting point.

3.6. Population structure visualisation

Figure 2 shows the admixture bar plot for $K = 2$–7 produced by CLUMPAK-LITE. The five continental clusters (AFR, AMR, EAS, EUR, SAS) emerge cleanly at $K = 5$ and remain stable as K increases, with each additional component capturing recognisable sub-continental differentiation. Run-RMSE below 0.02 across $K = 2$–7 (Figure 1b) confirms that the displayed solution is representative of the inferred distribution rather than an artefact of a single seed.

Collectively, these results demonstrate that GPUADMIX achieves one to two orders of magnitude faster inference than state-of-the-art CPU tools while matching or exceeding their accuracy, that a GPU-enabled multi-seed strategy extends this advantage to the multimodal high-K regime, and that all three core algorithmic components contribute substantively to performance.

4. Discussion

The central question motivating GPUADMIX is whether GPU acceleration of model-based ancestry estimation requires sacrificing the principled probabilistic framework that makes methods such as ADMIXTURE trustworthy for downstream analyses. Our results demonstrate that it does not: by reformulating the admixture EM updates as GPU-native dense matrix multiplications and augmenting them with FISTA-style Nesterov momentum, stochastic mini-batch EM, and streaming randomised SVD initialisation, GPUADMIX achieves 41× faster inference than FASTMIXTURE while matching or exceeding its log-likelihood and producing admixture proportion matrices that are virtually identical to those of ADMIXTURE ($Q\,r^2 > 0.9999$).

4.1. Reconciling GPU speed with model-based accuracy

Prior GPU-accelerated methods for ancestry estimation have pursued speed through model approximations. Neural ADMIXTURE (Mantes et al., 2023) trains a neural network surrogate for the admixture model, achieving fast inference but yielding markedly reduced self-consistency across SNP subsets (reported $r^2 \approx 0.72$ between full and downsampled 1kGP runs; Santander et al. 2024), suggesting that the neural parameterisation memorises rather than generalises. SCOPE (Chiu et al., 2022) departs from the likelihood-based EM framework entirely, replacing it with a principal-component objective that runs efficiently on GPU but loses the interpretable per-population allele frequencies that practitioners rely on for biological annotation. GPUADMIX, by contrast, preserves the exact ADMIXTURE binomial likelihood: both the E-step and M-step are reformulated as DGEMM calls without any approximation to the model. The FISTA-style Nesterov momentum contributes further by yielding 7,865 additional log-likelihood units over plain EM in ablation (Table 2); empirically, this gain is consistent with the accelerated updates visiting more of the likelihood surface during early iterations, though we caution that formal convergence guarantees for FISTA apply to convex objectives and the admixture landscape is non-convex. Taken together, these design choices explain why GPUADMIX achieves *higher* likelihood than FASTMIXTURE's SQUAREM-accelerated (Varadhan and Roland, 2008) CPU EM at $K = 5$, rather than merely matching it.

4.2. Multi-seed inference: a new practical default

The EM objective for admixture models is well-known to be multimodal (Jakobsson and Rosenberg, 2007), and our run-stability analysis confirms that this becomes practically significant at $K \geq 8$: the mean pairwise RMSE across five seeds rises from below 0.02 at $K \leq 7$ to approximately 0.04 at $K = 9$–10. Classical CPU workflows are largely constrained to one or two random restarts because each seed at high K may cost tens of minutes. GPUADMIX changes this calculus: five seeds at any $K \leq 10$ complete in under 300 seconds—less than the wall time of a single FASTMIXTURE run—so multi-seed inference incurs no additional opportunity cost relative to the CPU baseline. We acknowledge that the best-of-five comparison uses five times the compute of a single gpuADMIX run; the justification is that this total budget is still smaller than a single fastmixture run, making it the economically optimal strategy. Empirically, the best-of-five gpuADMIX seed surpasses the best-of-five fastmixture seed at every K tested, suggesting that the admixture landscape contains high-quality optima that Nesterov momentum finds and that SQUAREM alone does not. On practical grounds, we recommend running at least five seeds for $K \geq 8$ and reporting the run with the highest log-likelihood alongside the run-stability RMSE.

4.3. K selection: complementary criteria

The cross-validation optimum ($K = 9$) and the BIC minimum ($K = 4$) are complementary rather than contradictory signals. BIC applies a strong parameter penalty—$(K-1)(N+M)$ additional free parameters for each increment in K—that discourages detecting sub-continental structure unless it is overwhelmingly supported by the data. The 5-fold hold-out log-likelihood is more sensitive to fine-grained differentiation and rewards any model that improves prediction of held-out genotypes, favouring the additional sub-continental components visible at $K = 9$. In practice, the choice of K should be driven by the analytical goal: $K = 5$ is both biologically interpretable (five major continental ancestry groups in the 1000 Genomes data) and numerically stable (run-RMSE = 0.02); $K = 9$ captures finer structure but is only consistently recoverable with a multi-seed strategy. The run-stability RMSE from CLUMPAK-lite provides a complementary diagnostic that is invisible to likelihood-only criteria and helps practitioners identify K values where a single run may be misleading.

4.4. Limitations

Several limitations constrain the current work. First, our speedup comparisons pair a single NVIDIA L20 GPU against a 32-core server CPU; the reported 41× speedup over fastmixture reflects a platform-level advantage that combines hardware throughput with algorithmic improvements. A single-threaded or FLOP-normalised comparison would isolate the algorithmic contribution; we leave this to future work. Second, evaluation is restricted to the 1000 Genomes Phase 3 dataset; a second real dataset such as HGDP (Bergström et al., 2020) would strengthen claims about generalisation. Third, our ablation study is conducted at $K = 5$; the relative contributions of Nesterov momentum and mini-batch EM at $K \geq 8$, where the landscape is more rugged, are unknown. Fourth, gpuADMIX assumes Hardy–Weinberg equilibrium within ancestral populations and linkage equilibrium across SNPs; we mitigate the latter by using LD-pruned input variants, but residual LD may inflate the effective sample size and should be considered when interpreting results from high-LD regions. Fifth, the tool currently provides point estimates for $\mathbf{Q}$ and $\mathbf{P}$; block-bootstrap confidence intervals (Efron, 1979) are planned for a future release.

4.5. Outlook

The streaming SVD design positions gpuADMIX for datasets substantially larger than those tested here. Scaling to biobank cohorts (>100,000 individuals) will require prefetching pipelines to overlap GPU compute with host-to-device data transfer; validation at this scale remains future work. Multi-GPU parallelisation across K values—already demonstrated at $K = 2$–10 on eight GPUs—further enables full K sweeps with integrated cross-validation to complete in a single interactive session, transforming ancestry estimation from an overnight computational job into a responsive analysis tool.

5. Conclusion

We have presented gpuADMIX, a GPU-accelerated admixture estimation tool that achieves one to two orders of magnitude speedup over existing CPU-based methods while preserving the exact binomial likelihood model and matching or exceeding their solution quality. Three complementary algorithmic innovations—FISTA-style Nesterov momentum, stochastic mini-batch EM, and streaming randomised SVD initialisation—together explain why gpuADMIX outperforms CPU competitors in both speed and accuracy, not hardware advantage alone. By making multi-seed, multi-K ancestry inference practical within minutes rather than days, gpuADMIX and its integrated CLUMPAK-lite post-processor lower the computational barrier to rigorous admixture analysis at the scale of modern biobank datasets.

Acknowledgements

The author thanks the developers of ADMIXTURE and fastmixture for open-sourcing their tools and results, which made rigorous benchmarking possible.

Funding

This work received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of Interest

None declared.

Data availability

The 1000 Genomes Project data are publicly available at `https://www.internationalgenome.org/`. gpuADMIX source code and analysis scripts are available at `https://github.com/[REPO]`.

References

1000 Genomes Project Consortium et al. A global reference for human genetic variation. *Nature*, 526:68–74, 2015.

D. H. Alexander and K. Lange. Enhancements to the ADMIXTURE algorithm for individual ancestry estimation. *BMC Bioinformatics*, 12:246, 2011.

D. H. Alexander, J. Novembre, and K. Lange. Fast model-based estimation of ancestry in unrelated individuals. *Genome Research*, 19(9):1655–1664, 2009.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

A. Bergström, S. A. McCarthy, R. Hui, M. A. Almarri, Q. Ayub, P. Danecek, Y. Chen, S. Felkel, P. Hallast, J. Kamm, et al.

Insights into human genetic variation and population history from 929 diverse genomes. *Science*, 367(6484):eaay5012, 2020.

C. Bycroft, C. Freeman, D. Petkova, G. Band, L. T. Elliott, K. Sharp, A. Motyer, D. Vukcevic, O. Delaneau, J. O'Connell, et al. The UK Biobank resource with deep phenotyping and genomic data. *Nature*, 562:203–209, 2018.

A. M. Chiu, E. K. Molloy, Z. Tan, A. Talwalkar, and S. Sankararaman. Inferring population structure in biobank-scale genomic data. *American Journal of Human Genetics*, 109(4):727–737, 2022.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–22, 1977.

B. Efron. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7(1):1–26, 1979.

N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2): 217–288, 2011.

M. Jakobsson and N. A. Rosenberg. CLUMPP: A cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure. *Bioinformatics*, 23(14):1801–1806, 2007.

N. M. Kopelman, I. Mayrose, M. Jakobsson, and N. A. Rosenberg. CLUMPAK: a program for identifying clustering modes and packaging population structure inferences across K. *Molecular Ecology Resources*, 15(5):1179–1191, 2015.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

A. D. Mantes, D. M. Montserrat, C. D. Bustamante, X. Giró-i Nieto, and A. G. Ioannidis. Neural ADMIXTURE for rapid genomic clustering. *Nature Computational Science*, 3:621–629, 2023.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

J. Novembre, T. Johnson, K. Bryc, Z. Kutalik, A. R. Boyko, A. Auton, A. Indap, K. S. King, S. Bergmann, M. R. Nelson, et al. Genes mirror geography within Europe. *Nature*, 456: 98–101, 2008.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

A. L. Price, N. J. Patterson, R. M. Plenge, M. E. Weinblatt, N. A. Shadick, and D. Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38(8):904–909, 2006.

J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. De Bakker, M. J. Daly, et al. PLINK: A tool set for whole-genome association and population-based linkage analyses. *American Journal of Human Genetics*, 81(3):559–575, 2007.

A. Raj, M. Stephens, and J. K. Pritchard. faststructure: Variational inference of population structure in large SNP datasets. *Genetics*, 197(2):573–589, 2014.

N. A. Rosenberg, J. K. Pritchard, J. L. Weber, H. M. Cann, K. K. Kidd, L. A. Zhivotovsky, and M. W. Feldman. Genetic structure of human populations. *Science*, 298(5602):2381–2385, 2002.

C. G. Santander, A. Refoyo-Martinez, and J. Meisner. Faster model-based estimation of ancestry proportions. *bioRxiv*, 2024. Recommended by Peer Community in Evolutionary Biology, November 2024.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

H. Tang, J. Peng, P. Wang, and N. J. Risch. Estimation of individual admixture: Analytical and study design considerations. *Genetic Epidemiology*, 28(4):289–301, 2005.

R. Varadhan and C. Roland. Simple and globally convergent methods for accelerating the convergence of any EM algorithm. *Scandinavian Journal of Statistics*, 35(2):335–353, 2008.