Title

Estimation of genetic admixture proportions via haplotypes

Permalink

https://escholarship.org/uc/item/7kk6x6j8

Authors

Ko, Seyoon
Sobel, Eric M
Zhou, Hua
et al.

Publication Date

2024-12-01

DOI

10.1016/j.csbj.2024.11.043

Computational and Structural Biotechnology Journal 23 (2024) 4384–4395

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Research Article

Estimation of genetic admixture proportions via haplotypes

Seyoon Ko [a,b,c,*], Eric M. Sobel [a,d], Hua Zhou [a,b], Kenneth Lange [a,d,e]

[a] Department of Computational Medicine, University of California, Los Angeles, Los Angeles, CA 90095, USA
[b] Department of Biostatistics, University of California, Los Angeles, Los Angeles, CA 90095, USA
[c] Department of Mathematics, University of California, Los Angeles, Los Angeles, CA 90095, USA
[d] Department of Human Genetics, University of California, Los Angeles, Los Angeles, CA 90095, USA
[e] Department of Statistics, University of California, Los Angeles, Los Angeles, CA 90095, USA

ARTICLE INFO

Dataset link: https://github.com/OpenMendel/HaploADMIXTURE.jl, https://github.com/kose-y/HaploADMIXTURE-resources

Keywords:
Admixture
Ancestry informative marker
Sparse clustering
OpenMendel

ABSTRACT

Estimation of ancestral admixture is essential for creating personal genealogies, studying human history, and conducting genome-wide association studies (GWAS). The following three primary methods exist for estimating admixture coefficients. The frequentist approach directly maximizes the binomial loglikelihood. The Bayesian approach adds a reasonable prior and samples the posterior distribution. Finally, the nonparametric approach decomposes the genotype matrix algebraically. Each approach scales successfully to datasets with a million individuals and a million single nucleotide polymorphisms (SNPs). Despite their variety, all current approaches assume independence between SNPs. To achieve independence requires performing LD (linkage disequilibrium) filtering before analysis. Unfortunately, this tactic loses valuable information and usually retains many SNPs still in LD. The present paper explores the option of explicitly incorporating haplotypes in ancestry estimation. Our program, HaploADMIXTURE, operates on adjacent SNP pairs and jointly estimates their haplotype frequencies along with admixture coefficients. This more complex strategy takes advantage of the rich information available in haplotypes and ultimately yields better admixture estimates and better clustering of real populations in curated datasets.

1. Introduction

Estimation of genetic admixture is key to reconstructing personal genealogies and understanding population histories [1]. Adjusting for genetic ancestry is also a necessary prelude to genome-wide association studies (GWAS) for medical and anthropological traits [2]. Failure to account for ancestry can lead to false positives due to population stratification [3–5]. In these analyses, admixture coefficients serve as covariates adjusting for ancestry. Because admixture coefficients represent the proportions of a person's ancestry derived from different populations, they are more interpretable than principal components (PCs).

Admixture coefficients can be estimated simultaneously with allele frequencies in known or latent populations. ADMIXTURE [6] is the most widely-used likelihood-based software. It directly maximizes the binomial likelihood of the admixture coefficients and allele frequencies via alternating sequential quadratic programming [7]. Our recent Julia version, OpenADMIXTURE [8], incorporates time-saving software enhancements and AIM (ancestry informative markers) preselection via sparse K-means clustering [9]. STRUCTURE [10] and its extensions fastStructure [11] and TeraStructure [12] rely on Bayesian inference. SCOPE [13] replaces the genotype matrix by a low-rank matrix, which is delivered by alternating least squares and randomized linear algebra [14]. Each of the recent versions of these programs – OpenADMIXTURE, TeraStructure, and SCOPE – scales to biobank-size datasets of a million people and a million single nucleotide polymorphisms (SNPs).

A regrettable limitation of most of these programs is their assumption of independence for the alleles present at neighboring SNPs. To avoid this patently false assumption, SNPs are filtered to remove SNPs in linkage disequilibrium (LD). Filtering must reach a balance between LD elimination and the loss of valuable AIMs. The LD-aware program fineSTRUCTURE [15] scales poorly on large datasets [13], despite the clear advantage of using ancestry informative haplotypes over individual SNPs [16–18].

The current paper demonstrates the value of haplotypes in admixture estimation and population clustering. Given the combinatorial and computational complexities encountered, we consider only haplotypes formed from adjacent SNP pairs. Even with this limitation, haplotype models offer substantial improvements in estimation and clustering in

[*] Corresponding author at: Department of Mathematics, University of California, Los Angeles, Los Angeles, CA 90095, USA.
E-mail address: kos@ucla.edu (S. Ko).

https://doi.org/10.1016/j.csbj.2024.11.043
Received 1 August 2024; Received in revised form 26 November 2024; Accepted 26 November 2024

Available online 6 December 2024

simulated and real datasets on well-separated ancestral populations. Our new program, HaploADMIXTURE, builds on the high-performance computing (HPC) techniques pioneered in OpenADMIXTURE [8]. By leveraging the parallel processing capabilities of graphics processing units (GPUs), HaploADMIXTURE is able to run in reasonable time. In practice, only a minority of haplotypes are informative. To select ancestry informative haplotypes, we exploit unsupervised sparse K-means clustering via feature ranking [8,9]. This generic method, denoted by the acronym SKFR, selects the informative features (haplotypes) driving cluster formation. Our experience suggests that the SKFR-HaploADMIXTURE pipeline delivers the best admixture results currently available with reasonable computing times.

2. Methods

2.1. Admixture likelihoods

Consider a sample of I unrelated individuals, B haplotype blocks, and S SNPs per block. For our purposes S equals 1 or 2. Let $\boldsymbol{x}_{ib}$ denote the length-S genotype vector for haplotype block b of individual i. Each genotype of i counts the number of i's reference alleles present and is coded as a number from the set $\{0,1,2\}$. Haplotypes are coded as sequences of 0's and 1's, and every $\boldsymbol{x}_{ib} = \boldsymbol{h}_{ib1} + \boldsymbol{h}_{ib2}$ equals a sum of a maternal and paternal haplotypes. The blocks are taken to be contiguous, non-overlapping, and exhaustive. Haplotypes may be chosen through feature selection as discussed in Section 2.4. Let $p_{kb\boldsymbol{h}} > 0$ be the frequency of haplotype $\boldsymbol{h}$ of haplotype block b in population k, and let $q_{ki} > 0$ denote the fraction of i's genome coming from population k, where $1 \le k \le K$. The loglikelihood of the sample under a binomial distribution and independence of haplotype blocks is

$$\mathcal{L}(\boldsymbol{Q},\boldsymbol{P}) = \sum_{i=1}^{I}\sum_{b=1}^{B}\log\Bigg[\sum_{\boldsymbol{h}:\boldsymbol{h}\le\boldsymbol{x}_{ib}\text{ and }\boldsymbol{x}_{ib}-\boldsymbol{h}\le\boldsymbol{1}}\Big(\sum_{k=1}^{K} q_{ki}p_{kb\boldsymbol{h}}\Big)\Big(\sum_{k=1}^{K} q_{ki}p_{kb,\boldsymbol{x}_{ib}-\boldsymbol{h}}\Big)\Bigg], \quad (1)$$

where $\boldsymbol{x}_{ib}$ is the sum of the maternal haplotype $\boldsymbol{0} \le \boldsymbol{h} \le \boldsymbol{1}$ and the paternal haplotype $\boldsymbol{x}_{ib} - \boldsymbol{h}$. The matrix $\boldsymbol{Q}$ has dimension $K \times I$. Because there are 2^S possible haplotypes per block, $\boldsymbol{P}$ has dimension $K \times B \times 2^S$. The constraints $\sum_{k=1}^{K} q_{ki} = 1$ and $\sum_{\boldsymbol{h}} p_{kb\boldsymbol{h}} = 1$ hold for each i and combination (k,b). The loglikelihood (1) simplifies by symmetry if any entry of $\boldsymbol{x}_{ib}$ equals 1 (a heterozygous SNP). Because maternal and paternal haplotypes are interchangeable, the number of summands can be halved if the remaining sum of products is doubled. When $S = 1$ and i and b are fixed, the heterozygous genotype 1 has probability $2(\sum_k q_{ki}p_{kb0})(\sum_k q_{ki}p_{kb1})$, which the log function splits into a sum of logarithms. In fact, this simplification replicates the binomial likelihood employed in ADMIXTURE and STRUCTURE. When $S = 2$, the doubly heterozygous genotype has the probability $2\left[\sum_k q_{ki}p_{kb(00)}\right]\left[\sum_k q_{ki}p_{kb(11)}\right] + 2\left[\sum_k q_{ki}p_{kb(01)}\right]\left[\sum_k q_{ki}p_{kb(10)}\right]$, which no longer splits under the log function. In addition, there are cases where one of the genotypes is observed, but the other is missing. For example, if the first genotype is heterozygous and the other is missing, the probability equals $\left[\sum_k q_{ki}p_{kb(00)} + \sum_k q_{ki}p_{kb(01)}\right]\left[\sum_k q_{ki}p_{kb(10)} + \sum_k q_{ki}p_{kb(11)}\right]$, and the loglikelihood (1) should be adjusted accordingly. Nonetheless, as described in the next subsection, the whole loglikelihood is still amenable to maximization.

2.2. Maximum likelihood estimation

Estimation in HaploADMIXTURE and OpenADMIXTURE are similar. The optimization machinery in both programs alternates estimation of the per-population haplotype frequencies $p_{kb\boldsymbol{h}}$ and the per-individual admixture coefficients q_{ki}. To allow easy parallelization with graphics processing units (GPUs), we invoke the minorization-maximization (MM) principle [19,20] to split sums appearing in the arguments to the logarithms of the haplotype loglikelihood (1). The operative inequality

$$\begin{aligned}\log(u+v) &\ge \frac{u^{(n)}}{u^{(n)}+v^{(n)}}\log u + \frac{v^{(n)}}{u^{(n)}+v^{(n)}}\log v \\ &\quad + \frac{u^{(n)}}{u^{(n)}+v^{(n)}}\log\frac{u^{(n)}+v^{(n)}}{u^{(n)}} + \frac{v^{(n)}}{u^{(n)}+v^{(n)}}\log\frac{u^{(n)}+v^{(n)}}{v^{(n)}} \\ &= \frac{u^{(n)}}{u^{(n)}+v^{(n)}}\log u + \frac{v^{(n)}}{u^{(n)}+v^{(n)}}\log v + c_n\end{aligned}$$

reduces to an equality when $u = u^{(n)}$ and $v = v^{(n)}$. Here the irrelevant constant c_n depends only on the current values $u^{(n)}$ and $v^{(n)}$ of u and v. The function $\frac{u^{(n)}}{u^{(n)}+v^{(n)}}\log u + \frac{v^{(n)}}{u^{(n)}+v^{(n)}}\log v$ becomes a surrogate for the function $\log(u+v)$ it replaces. For example, when $S = 2$ and i presents a doubly heterozygous genotype, we take $u = 2\left[\sum_k q_{ki}p_{kb(00)}\right]\left[\sum_k q_{ki}p_{kb(11)}\right]$ and $v = 2\left[\sum_k q_{ki}p_{kb(01)}\right]\left[\sum_k q_{ki}p_{kb(10)}\right]$. Most genotype probabilities (all homozygous and singly heterozygous genotypes) reduce to a single product where log splitting is unnecessary. For haplotypes involving more than two SNPs, phase combinations become more complex, code is harder to write, and computation slows. For these reasons we venture no further than two-SNP haplotypes. Maximization of the surrogate function created by minorization enjoys the ascent property of steadily increasing the loglikelihood. The ascent property is the essence of the MM (minorization-maximization) principle [19,20].

Minorization creates a surrogate function

$$\mathcal{G}(\boldsymbol{Q},\boldsymbol{P}\mid\boldsymbol{Q}^{(n)},\boldsymbol{P}^{(n)}) = \sum_{i=1}^{I}\sum_{b=1}^{B}\sum_{\boldsymbol{h}} w_{ikb\boldsymbol{h}}^{(n)}\log\Big(\sum_{k=1}^{K} q_{ki}p_{kb\boldsymbol{h}}\Big) \quad (2)$$

involving nonnegative weights $w_{ikb\boldsymbol{h}}^{(n)}$, many of which are 0 because they correspond to haplotypes incompatible with observed genotypes. Except for revising the weights $w_{ikb\boldsymbol{h}}^{(n)}$ at each iteration n, the surrogate loglikelihood (2) is simpler to deal with than the actual loglikelihood. Updating the admixture matrix $\boldsymbol{Q} = (q_{ki})$ can be done simultaneously over columns (individuals i). Updating the haplotype frequency tensor $\boldsymbol{P} = (p_{kb\boldsymbol{h}})$ can be done simultaneously over its middle columns (blocks b). Each such maximization must respect the nonnegativity constraints on the proportions and their sum to 1 constraints. Very simple multinomial updates of the $p_{kb\boldsymbol{h}}$ can be achieved by splitting the argument $\sum_{k=1}^{K} q_{ki}p_{kb\boldsymbol{h}}$ of the log function, but this second minorization slows convergence dramatically.

The parallel updates of $\boldsymbol{P}$ and $\boldsymbol{Q}$ are structured around functions of the form

$$\mathcal{G}(\boldsymbol{r}) = \sum_j w_j \log\Big(\sum_k c_{jk}r_k\Big)$$

subject to nonnegativity and sum to 1 constraints. The method of recursive quadratic programming involves replacing $\mathcal{G}(\boldsymbol{r})$ by its local quadratic approximation

$$\mathcal{G}(\boldsymbol{r}) \approx \mathcal{G}(\boldsymbol{r}^{(n)}) + \nabla\mathcal{G}(\boldsymbol{r}^{(n)})^\top(\boldsymbol{r}-\boldsymbol{r}^{(n)}) + \frac{1}{2}(\boldsymbol{r}-\boldsymbol{r}^{(n)})^\top d^2\mathcal{G}(\boldsymbol{r}^{(n)})(\boldsymbol{r}-\boldsymbol{r}^{(n)})$$

and maximizing this approximation subject to the constraints. The required gradient and Hessian are

$$\nabla\mathcal{G}(\boldsymbol{r}) = \sum_j \frac{w_j\boldsymbol{c}_j}{\sum_k c_{kj}r_k} \quad\text{and}\quad d^2\mathcal{G}(\boldsymbol{r}) = -\sum_j \frac{w_j\boldsymbol{c}_j\boldsymbol{c}_j^\top}{(\sum_k c_{jk}r_k)^2},$$

where $\boldsymbol{c}_j^\top$ is the jth row of the matrix $\boldsymbol{C}$ of nonnegative constants c_{jk}.

Given the structure of the problem, the Hessian is block diagonal. As a consequence, the computation of the gradients and Hessians of $\mathcal{G}(\boldsymbol{Q},\boldsymbol{P}\mid\boldsymbol{Q}^{(n)},\boldsymbol{P}^{(n)})$ with respect to $\boldsymbol{Q}$ has time complexity $O(2^S IBK^2)$ and space complexity $O(IK^2)$. Computation of the gradients and Hessians with respect to $\boldsymbol{P}$ again has time complexity $O(2^S IBK^2)$ but now space complexity $O(2^S BK^2)$. The quadratic programming cost of updating $\boldsymbol{Q}$ breaks down into I quadratic programs of size K with a single equality constraint. By design, solving these small quadratic programs in parallel circumvents the computation and storage of the massive Hessian of the full objective. The cost of solving one of these quadratic programs is polynomial in K. The quadratic programming cost of updating $\boldsymbol{P}$ breaks down into B quadratic programs of size $2^S K$ with an equality constraint for each population k. The cost of solving one of these quadratic programs has complexity polynomial in $2^S K$. In practice, when $S = 2$, the time needed for solving the quadratic programs for $\boldsymbol{Q}$ is negligible compared to the time proportional to IB for computing gradients and Hessians. In contrast, the time needed for solving the quadratic programs for $\boldsymbol{P}$ is comparable to the time needed for computing gradients and Hessians.

Our Julia implementation of HaploADMIXTURE allows users to invoke Nvidia graphics processing units (GPUs) to accelerate the evaluation of gradients and Hessians and to solve the various quadratic programs. Convergence criteria can be set by the user. The default setting for overall convergence mandates that the relative change in loglikelihoods falls below 10^{-7}.

2.3. *Selection of K*

We employ two devices to select the number of ancestral populations K. First, the cross-validation method introduced in ADMIXTURE [21] partitions the sample individuals into v folds. Each of the folds is held out as a validation set, and the model is fit on the remaining training individuals. Fitting on a training set is fast because it warm starts parameter values from the estimates garnered under the full dataset. Given the haplotype frequencies $\boldsymbol{P}_{train}$ estimated on the training set fixed, we estimate the admixture fractions $\boldsymbol{Q}_{test}$ on the validation set. This fitting step is also fast because it qualifies as a straightforward convex problem. Given $\boldsymbol{P}_{train}$ and $\boldsymbol{Q}_{test}$, we predict the genotype matrix of the validation individuals. The deviance residual under a binomial model yields the prediction error

$$d(\boldsymbol{x}, \boldsymbol{y}) = \sum_i \sum_j \left[x_{ij} \log \frac{x_{ij}}{y_{ij}} + (2 - x_{ij}) \log \frac{(2 - x_{ij})}{(2 - y_{ij})} \right],$$

where $\boldsymbol{x}$ is $I \times SB$ true genotype matrix, and $\boldsymbol{y}$ is the predicted genotype matrix. This error is then averaged across the different folds. We choose the most parsimonious model whose prediction error is no more than one standard error above the error of the best model (one standard error rule).

The second device for selecting K is the Akaike information criterion (AIC) [22]. In the current setting

$$\text{AIC} = 2\left[BK(2^S - 1) + I(K - 1) - \mathcal{L}(\hat{\boldsymbol{Q}}, \hat{\boldsymbol{P}})\right]$$

The term $BK(2^S - 1) + I(K - 1)$ counts the number of free parameters in the model with K ancestral populations. The loglikelihood is evaluated at the maximum likelihood estimates given K. We fit the model for several different values of K and choose the K with the lowest value of AIC. The virtue of AIC is that it requires less computation than full cross-validation.

2.4. *Sparse K-means with feature ranking for haplotypes*

To select AIMs, sparse K-means with feature ranking (SKFR) [8,9] has proved ideal. SKFR ranks and selects a predetermined number of features (sparsity level) s based on their importance in K-means clustering. HaploADMIXTURE requires input blocks of SNPs rather than individual SNPs. The center for cluster j is a vector $\boldsymbol{c}_j = (c_{jg})$. The loss in K-means clustering is $\sum_{j \in J} \sum_{i \in C_j} \|\boldsymbol{c}_j - \boldsymbol{x}_i\|^2$, where C_j denotes the set of individuals belonging to cluster j, and each raw feature vector $\boldsymbol{x}_i = \boldsymbol{h}_m + \boldsymbol{h}_p$ is a sum of unknown haplotypes. (If everyone is haplotyped, then SKFR should operate on haplotypes.) In practice, the $\boldsymbol{x}_i$ are standardized to have a mean of 0 for each feature across the entire sample. A missing genotype x_{ig} in $\boldsymbol{x}_i$ is imputed by the center coordinate c_{jg} when i is assigned to cluster j [23]. To model haplotypes, the feature vector $\boldsymbol{x}_i$ is broken into vector blocks $\boldsymbol{x}_{ib}$. In identifying AIMs, Lloyd's algorithm [24] alternates updating cluster centers and reassigning feature vectors to clusters. At each iteration of the SKFR algorithm, the s blocks giving the largest reduction in the loss are selected based on the decomposition $\|\boldsymbol{c}_j - \boldsymbol{x}_i\|^2 = \sum_b \|\boldsymbol{c}_{jb} - \boldsymbol{x}_{ib}\|^2$. The mean for a selected block is cluster-specific. The mean for a non-selected block is taken to be $\mathbf{0}$. Our sparsity inducing version of Lloyd's algorithm converges when the cluster centers and ancestry informative blocks stabilize.

2.5. *Supervised inference of population*

Given the population haplotype frequencies $\hat{\boldsymbol{P}}$, we can estimate population structure $\hat{\boldsymbol{Q}}$ by fixing $\boldsymbol{P}$ and only updating $\boldsymbol{Q}$. The problem becomes convex and can be efficiently solved. This technique is used for cross-validation and for our large-scale analysis of the UK Biobank dataset.

2.6. *Computational tactics*

Most of the computational tactics introduced in OpenADMIXTURE carry over to HaploADMIXTURE. For instance, HaploADMIXTURE significantly reduces memory demands by directly converting the bit genotypes stored in PLINK BED format [25] into numbers through the OpenMendel [26] package SnpArrays. Multithreading is employed throughout HaploADMIXTURE. Multithreading not only promotes parallelism, but also reduces memory usage by tiling the computation of gradients and Hessians. CUDA GPU kernels are implemented for EM updates and computing gradients and Hessians. When running SKFR for multiple sparsity levels s, we start with the highest level of s and warm start Lloyd's algorithm at the current level by its converged value at the previous higher level. We refer the readers to Ko et al. [8] for further details.

2.7. *Performance evaluation*

2.7.1. *Permutation matching of clusters*

A promising similarity metric proposed by Behr et al. [27] is effective in matching clusters defined by two admixture matrices $\boldsymbol{Q}^1$ and $\boldsymbol{Q}^2$. This metric faithfully matches similar clusters and is invariant when cluster labels are permuted. The metric quantifies the similarity between cluster m in $\boldsymbol{Q}^1$ and cluster n in $\boldsymbol{Q}^2$ as

$$\mathcal{J}(\boldsymbol{q}_m^1, \boldsymbol{q}_n^2) = 1 - \sqrt{\frac{\sum_{i=1}^I (q_{mi}^1 - q_{ni}^2)^2}{2|N|}},$$

where N is the set of indices i for which $q_{mi}^1 + q_{ni}^2 > 0$, and $|N|$ is the cardinality of N. To match the clusters delivered by two algorithms, we solve the assignment problem that maximizes the criterion $\sum_m \sum_n y_{mn} \mathcal{J}(\boldsymbol{q}_m^1, \boldsymbol{q}_n^2)$, subject to the constraints $y_{mn} \in \{0, 1\}$ and $\sum_k^K y_{km} = \sum_k^K y_{nk} = 1$, where K is the number of clusters. In practice, we relax the domain of y_{mn} to the unit interval and solve the simplified problem using linear programming via JuMP [28], a mathematical optimization package in Julia.

2.7.2. *Silhouette coefficient*

The silhouette index s_i is a measure of how similar object i is to its own cluster (cohesion) compared to other clusters (separation) [29]. If i belongs to cluster C_k, then the index s_i reflects the two averages

$$a_i = \frac{\sum_{j \in C_k \setminus \{i\}} \text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{|C_k| - 1}$$

$$b_i = \min_{l \neq k} \frac{\sum_{j \in C_l} \text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{|C_l|},$$

where a_i is the average distance of sample i from the other points in C_k, and b_i is the minimum average distance of sample i from the other clusters. Given these values we define

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}.$$

Note that s_i ranges from −1 to 1; the higher s_i is, the better separated the clusters are. Thus, the average silhouette value serves as a sensitive measure of clustering quality.

2.7.3. Visualization

Stacked bar plots allow easy visualization of estimated admixture proportions when clusters are matched consistently across computer runs. Matching is accomplished by hierarchical clustering with complete linkage based on the HaploADMIXTURE $\boldsymbol{Q}$ estimates. Hierarchical clustering determines the order of samples within a population. One can also apply hierarchical clustering to the set of populations and to the set of continents. In the former case, clustering operates on cluster centers, and in the latter case, on averages of cluster centers.

2.8. Real datasets

To evaluate its performance, we applied HaploADMIXTURE to four different real-world datasets: the 1000 Genomes Project (TGP) [30,31], the Human Genome Diversity Project (HGDP) [32,33], the Human Origins (HO) [34] project, and the UK Biobank data (UKB) [35]. (We adhered to compliance agreements in each case.) The TGP dataset includes genotypes from the 2012-01-31 Omni Platform after filtering to exclude related individuals, individuals with less than a 95% genotyping success rate, and variants with minor allele frequency (MAF) less than 1%. The filtered dataset contains 1718 unrelated individuals and 1,854,622 SNPs. The self-reported ancestry labels range over 26 different populations grouped into continental populations of African (AFR), Admixed American (AMR), East Asian (EAS), European (EUR), and South Asian (SAS) descent. The HGDP dataset contains 940 individuals across 32 self-reported populations and 642,950 SNPs after filtering by the same criteria applied to the TGP data. The self-reported population labels are further grouped into seven continental labels: Europe, Middle East, Central South Asia, East Asia, Africa, America, and Oceania. The HO dataset includes 1931 individuals across 163 populations and 385,088 SNPs. Here, filtering excludes individuals with less than 99% genotyping success rate and SNPs with MAF less than 5%. No continental population labels are provided for HO. Our discussion of results focuses on the TGP dataset. Corresponding results for HGDP and HO appear in the Supplementary Materials. For the UK Biobank data, we select 488,154 individuals with a 95% or better genotyping rate and 178,734 SNPs shared with the TGP dataset and having at least 1% MAF.

2.9. Simulations

The model for simulating genetic admixture is a variant of the Pritchard-Stephens-Donnelly (PSD) model [10], with allele frequencies sampled from the Balding-Nicolas model [36] that follows a beta distribution:

$$
\begin{aligned}
p_{kb1} &\overset{\text{iid}}{\sim} \text{Beta}\left(\frac{1-F_{ST}}{F_{ST}}p_A, \frac{1-F_{ST}}{F_{ST}}(1-p_A)\right)\\
\boldsymbol{s}_t &\overset{\text{iid}}{\sim} \text{Dirichlet}(\alpha\mathbf{1}_K), \quad \text{regional centers}\\
\boldsymbol{q}_{\cdot i} &\overset{\text{iid}}{\sim} \text{Dirichlet}(\gamma\boldsymbol{s}_t),
\end{aligned}
$$

where p_A is the allele frequency and F_{ST} is the fixation index. Chiu et al. [13] introduced the second line's extra level of Dirichlet sampling to simulate populations gathered around regional centers. This is accomplished by first sampling T regional centers $\boldsymbol{s}_t$ from the Dirichlet$(\alpha\mathbf{1}_K)$ distribution. Then for each regional center, I/T of the admixture vectors $\boldsymbol{q}_{\cdot i}$ are sampled around the center $\boldsymbol{s}_t$, with a high value of the parameter $\gamma = 50$.

To model SNPs in linkage disequilibrium, we sample two haplotypes separately by sequential Bernoulli sampling, instead of sampling them independently. Haplotypes h_{ij1} and h_{ij2} are sampled from the conditional Bernoulli distribution given $h_{i(j-1)1}$ and $h_{i(j-1)2}$, respectively, so that the Pearson correlation coefficient between h_{ijm} and $h_{i(j-1)m}$ is a constant ρ and the marginal distribution of h_{ijm} follows Bernoulli$(\sum_k q_{ki}p_{kj})$. To specify the underlying parameters p_A and F_{ST}, we randomly sampled SNPs from chromosome 1 of the TGP dataset and used their minor allele frequencies and the estimated fixation indexes. If any minor allele frequency fell below 0.005, we clamped it to 0.005.

Table 1
Root-mean-square errors of the estimated admixture proportions on the simulated datasets, 10,000 SNPs, $\alpha = 0.02$. Root-mean-square error checks the accuracy of the estimated admixture coefficients; the lower, the better. Five populations were used for the simulation, with 1000 individuals and 10,000 SNPs for various values of ρ, the correlation between two nearby SNPs. Each value is averaged over five simulation runs. The best value for each ρ is in italics.

AIMs	HaploADMIXTURE	OpenADMIXTURE	SCOPE	TeraStructure
$\rho = 0.75$				
10,000	*0.0108*	0.0415	0.0575	0.0737
2000	0.0127	0.0419		
1500	0.0134	0.0429		
1000	0.0150	0.0447		
500	0.0203	0.0486		
$\rho = 0.5$				
10,000	*0.0087*	0.0285	0.0476	0.0579
2000	0.0119	0.0308		
1500	0.0131	0.0320		
1000	0.0152	0.0338		
500	0.0214	0.0381		
$\rho = 0.25$				
10,000	*0.0098*	0.0199	0.0415	0.0434
2000	0.0150	0.0228		
1500	0.0168	0.0239		
1000	0.0194	0.0256		
500	0.0259	0.0295		
$\rho = 0$				
10,000	*0.0141*	0.0167	0.0393	0.0294
2000	0.0193	0.0198		
1500	0.0212	0.0209		
1000	0.0237	0.0226		
500	0.0293	0.0266		

3. Results

3.1. Simulation studies

We simulated datasets with different numbers of SNPs, values of the concentration parameter $\alpha \in \{0.02, 0.05, 0.1\}$, and correlation between nearby SNPs $\rho \in \{0, 0.25, 0.5, 0.75\}$ as described in Section 2.9. Tables 1, 2, and 3 display root-mean-square errors (RMSE) for $\alpha = 0.02$ and 10,000, 100,000, and 1,000,000 simulated SNPs, respectively. Results with different values of α are available in the Supplemental Materials in Tables S2-S7. RMSE is estimated by

$$
\widehat{\text{RMSE}}(\hat{\boldsymbol{Q}}) = \sqrt{\frac{1}{IK}\sum_{i,k}(q_{ik} - \hat{q}_{ik})},
$$

where $\boldsymbol{Q}$ are the true values and $\hat{\boldsymbol{Q}}$ are the estimates. When the populations are easily distinguishable with $\alpha = 0.02$, and $\rho = 0.75$, HaploADMIXTURE performs better than OpenADMIXTURE, SCOPE, and TeraStructure, as HaploADMIXTURE accounts for LD. OpenADMIXTURE performs better for 1,000,000 SNPs with lower LD with the correlation coefficients of 0, 0.25, and 0.5. When lower numbers of AIMs selected by sparse K-means are used, HaploADMIXTURE and OpenADMIXTURE both maintain their performance reasonably well, sometimes even improving on the results found with all of the SNPs. As populations get harder to distinguish with higher value of α, HaploADMIXTURE begins to struggle. For all of the cases evaluated, the AIC correctly selects $K = 5$.

Table 2

Root-mean-square errors of the estimated admixture proportions on the simulated datasets, 100,000 SNPs, $\alpha = 0.02$**.** Root-mean-square error checks the accuracy of the estimated admixture coefficients; the lower, the better. Five populations were used for the simulation, with 1000 individuals and 100,000 SNPs for various values of ρ, the correlation between two nearby SNPs. Each value is averaged over five simulation runs. The best value for each α is in italics.

AIMs	HaploADMIXTURE	OpenADMIXTURE	SCOPE	TeraStructure
$\rho = 0.75$				
100,000	*0.0089*	0.0195	0.0258	0.0403
20,000	0.0092	0.0183		
15,000	0.0092	0.0185		
10,000	0.0090	0.0192		
5000	0.0090	0.0207		
$\rho = 0.5$				
100,000	*0.0061*	0.0285	0.0200	0.0195
20,000	0.0068	0.0112		
15,000	0.0068	0.0118		
10,000	0.0070	0.0123		
5000	0.0074	0.0137		
$\rho = 0.25$				
100,000	*0.0035*	0.0056	0.0162	0.0158
20,000	0.0043	0.0070		
15,000	0.0045	0.0076		
10,000	0.0050	0.0082		
5000	0.0061	0.0094		
$\rho = 0$				
100,000	*0.0034*	0.0042	0.149	0.0085
20,000	0.0050	0.0058		
15,000	0.0055	0.0062		
10,000	0.0062	0.0067		
5000	0.0078	0.0079		

Table 3

Root-mean-square errors of the estimated admixture proportions on the simulated datasets, 1,000,000 SNPs, $\alpha = 0.02$**.** Root-mean-square error checks the accuracy of the estimated admixture coefficients; the lower, the better. Five populations were used for the simulation, with 1000 individuals and 1,000,000 SNPs for various values of ρ, the correlation between two nearby SNPs. Each value is averaged over five simulation runs. The best value for each ρ is in italics.

AIMs	HaploADMIXTURE	OpenADMIXTURE	SCOPE	TeraStructure
$\rho = 0.75$				
1,000,000	0.0087	0.0154	0.0119	0.0119
200,000	0.0089	0.0123		
150,000	0.0088	0.0110		
100,000	0.0085	0.0097		
50,000	*0.0081*	0.0094		
$\rho = 0.5$				
1,000,000	0.0058	0.0052	0.0079	0.0097
200,000	0.0063	*0.0040*		
150,000	0.0064	*0.0040*		
100,000	0.0063	0.0042		
50,000	0.0063	0.0049		
$\rho = 0.25$				
1,000,000	0.0029	*0.0014*	0.0058	0.0042
200,000	0.0093	0.0028		
150,000	0.0094	0.0029		
100,000	0.0096	0.0031		
50,000	0.0099	0.0034		
$\rho = 0$				
1,000,000	0.0045	*0.0009*	0.0051	0.0045
200,000	0.0020	0.0015		
150,000	0.0021	0.0017		
100,000	0.0022	0.0019		
50,000	0.0025	0.0023		

Table 4

Entropy per SNP per individual of $\boldsymbol{P}$ for TGP, HGDP, and HO.

Software	TGP	HGDP	HO
HaploADMIXTURE	0.303	0.344	0.360
OpenADMIXTURE	0.347	0.414	0.444
SCOPE	0.347	0.339	0.422
TeraStructure	0.393	0.432	0.474

3.2. Real-world datasets

3.2.1. Selection of K

To assess the performance of HaploADMIXTURE, we computed AIC values and performed cross-validation to select the best K for the real-world datasets. For TGP, both AIC and cross-validation select $K = 7$, while TeraStructure selects $K = 8$. For HGDP, AIC selects $K = 7$, but cross-validation and TeraStructure select $K = 10$. For HO, AIC selects $K = 12$, cross-validation selects $K = 10$, and TeraStructure selects $K = 14$. On balance, we prefer AIC because of its computational efficiency and parsimony. This preference is bolstered by the notable differences observed in the graphs between TeraStructure and OpenADMIXTURE covered in Section 3.2.2. In the following sections, we use the same values of K across all the tools we compare. We use $K = 7$ for TGP and HGDP, and $K = 12$ for HO, as selected by AIC. For the UK Biobank dataset, we choose $K = 7$ as in the TGP dataset, as we perform supervised inference based on the result from the TGP data. In total, HaploADMIXTURE estimates 25,976,734 parameters for TGP, 9,007,880 parameters for HGDP, 18,507,396 parameters for HO, and 8,421,630 parameters for UKB.

3.2.2. Visualization

Figs. 1, S2, and S3 illustrate the admixture proportions inferred from the TGP, HGDP, and HO datasets by HaploADMIXTURE, OpenADMIXTURE, SCOPE, and TeraStructure. The general structure seems similar across the programs, with some differences. For example, TeraStructure tends to rely on a single European (EUR) population in TGP, while the other programs tend to rely on two. Section 3.2.4 summarizes the ability of the programs to separate self-identified populations. Previous publications of Chiu et al. [13] and Ko et al. [8] incorrectly match individuals to populations because of a data reading error. Figure S2 fixes this error and clearly separates the different continental populations.

Figs. 2, S4, and S6 show the structures inferred by HaploADMIXTURE operating on AIMs chosen through sparse K-means clustering. Figs. 3, S5, and S7 display the structures inferred by OpenADMIXTURE in the same circumstances. Evidently, HaploADMIXTURE faithfully reproduces the general structure with fewer AIMs than OpenADMIXTURE. In particular, OpenADMIXTURE fails to distinguish European populations from Middle-Eastern populations on the HGDP data. Figs. 4, S8, and S9 display the structure inferred by HaploADMIXTURE for different numbers of populations K as discussed in Section 3.2.1.

3.2.3. Loglikelihood and entropy

Table S8 displays the likelihood of the fitted models. Since the binomial model of OpenADMIXTURE is a submodel of the model of HaploADMIXTURE, the maximum loglikelihood of the former is always less than the maximum loglikelihood of the latter based on the same SNP set. Table 4 shows the entropy of $\boldsymbol{P}$, the array of genotype/haplotype frequencies for each dataset. The entropy decrease in HaploADMIXTURE compared to OpenADMIXTURE quantifies the additional information available in haplotypes. The entropy of $\boldsymbol{P}$ using HaploADMIXTURE for TGP, HGDP, and HO show 12.7%, 16.9%, and 18.9% reductions, respectively, compared to OpenADMIXTURE. TeraStructure has higher entropy than OpenADMIXTURE, and SCOPE has entropy similar to OpenADMIXTURE on TGP and HO datasets. On HGDP, SCOPE has a similar entropy to HaploADMIXTURE. Note that the SCOPE model does not directly optimize the binomial loglikelihood model.

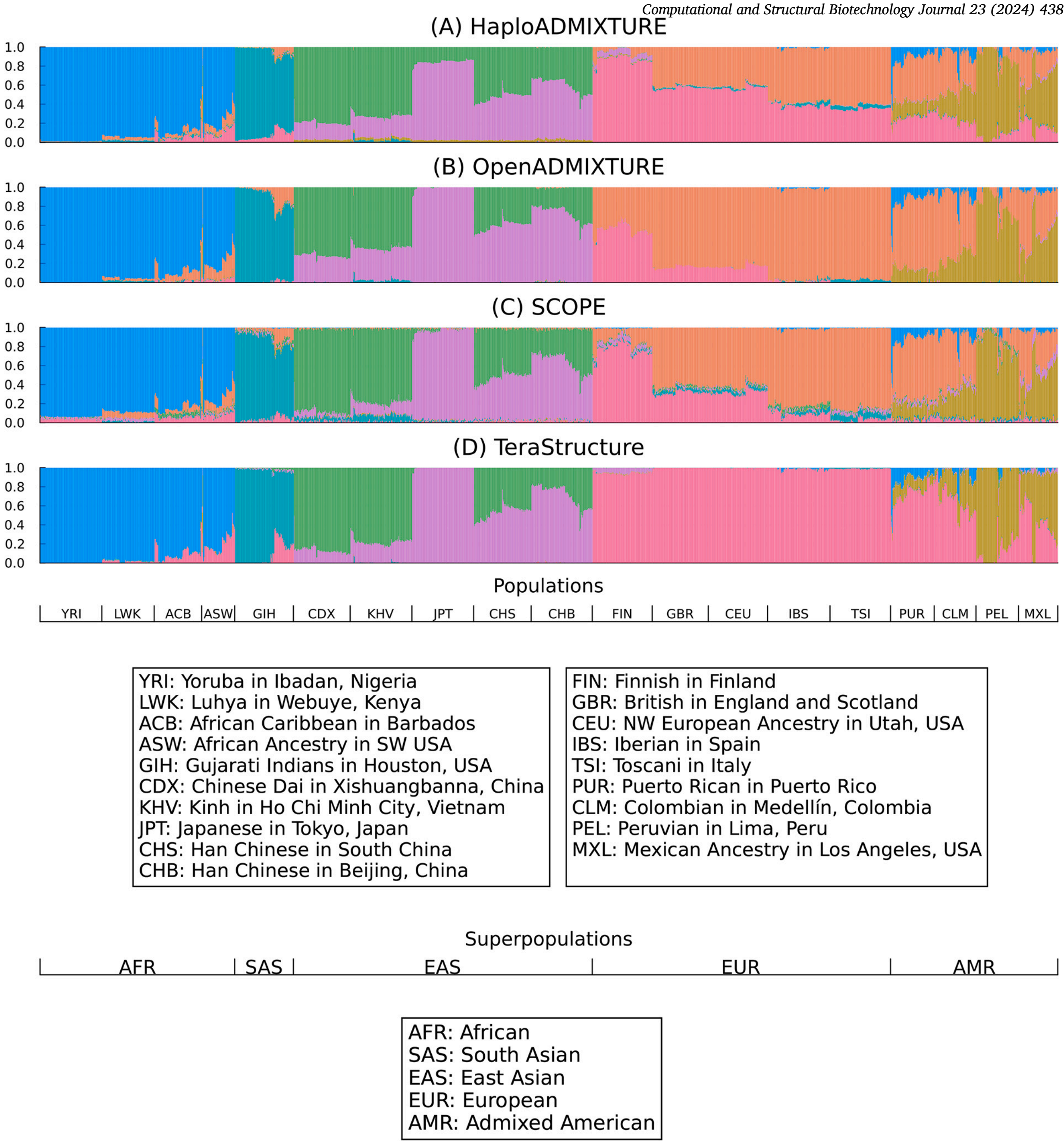

Fig. 1. Ancestry estimation of TGP data samples. (a) Using HaploADMIXTURE with all SNPs, (b) OpenADMIXTURE with all SNPs, (c) SCOPE, and (d) TeraStructure. The results are presented in stacked bar plots, where the y-axis indicates the proportion of total ancestry. The x-axis shows all samples arranged by population labels.

3.2.4. *Evaluation of estimated admixture*

Silhouette coefficients offer another way of quantifying performance. These are based on the ancestry labels implicit in the estimated $\boldsymbol{Q}$ matrix. The average silhouette coefficient is preferable to the training errors of linear classifiers and their cross-entropies [13,8] because training error is discrete, and a single individual can unduly influence cross-entropy. We additionally matched clusters as discussed in Section 2.7.1 and computed root-mean-square error (RMSE) from the SKFR clusters derived from all SNPs.

Tables 6, S10, and S11 display mean silhouette coefficients for HaploADMIXTURE, OpenADMIXTURE, SCOPE, and TeraStructure. Since one of the continental populations is known to be admixed Americans, we also provide the result without them in Table S9. Tables S12-S14 show continent-by-continent mean silhouette coefficients, and Tables S15-S18 show region-by-region mean silhouette coefficients. HaploADMIXTURE generally performs well in grouping populations by both continent and region. OpenADMIXTURE performs equally well in grouping by continent but in grouping regional labels, HaploADMIXTURE shows consistently higher value of the mean silhouette. For TGP and HGDP, TeraStructure is the best at distinguishing continental labels but falters in distinguishing regional labels, particularly in the TGP data where Middle-Eastern and European populations are lumped. For HGDP, SCOPE is the best at distinguishing the 32 regional labels but struggles compared to HaploADMIXTURE and OpenADMIXTURE in distinguishing African continental populations from each other. For the HO dataset, HaploADMIXTURE and OpenADMIXTURE perform similarly in distinguishing the 163 regional labels, followed by SCOPE and TeraStructure.

When the analysis is based on AIMs, HaploADMIXTURE usually performs better than OpenADMIXTURE. In the single instance of 5000 AIMs for TGP, HaploADMIXTURE suffers in distinguishing regional subpop-

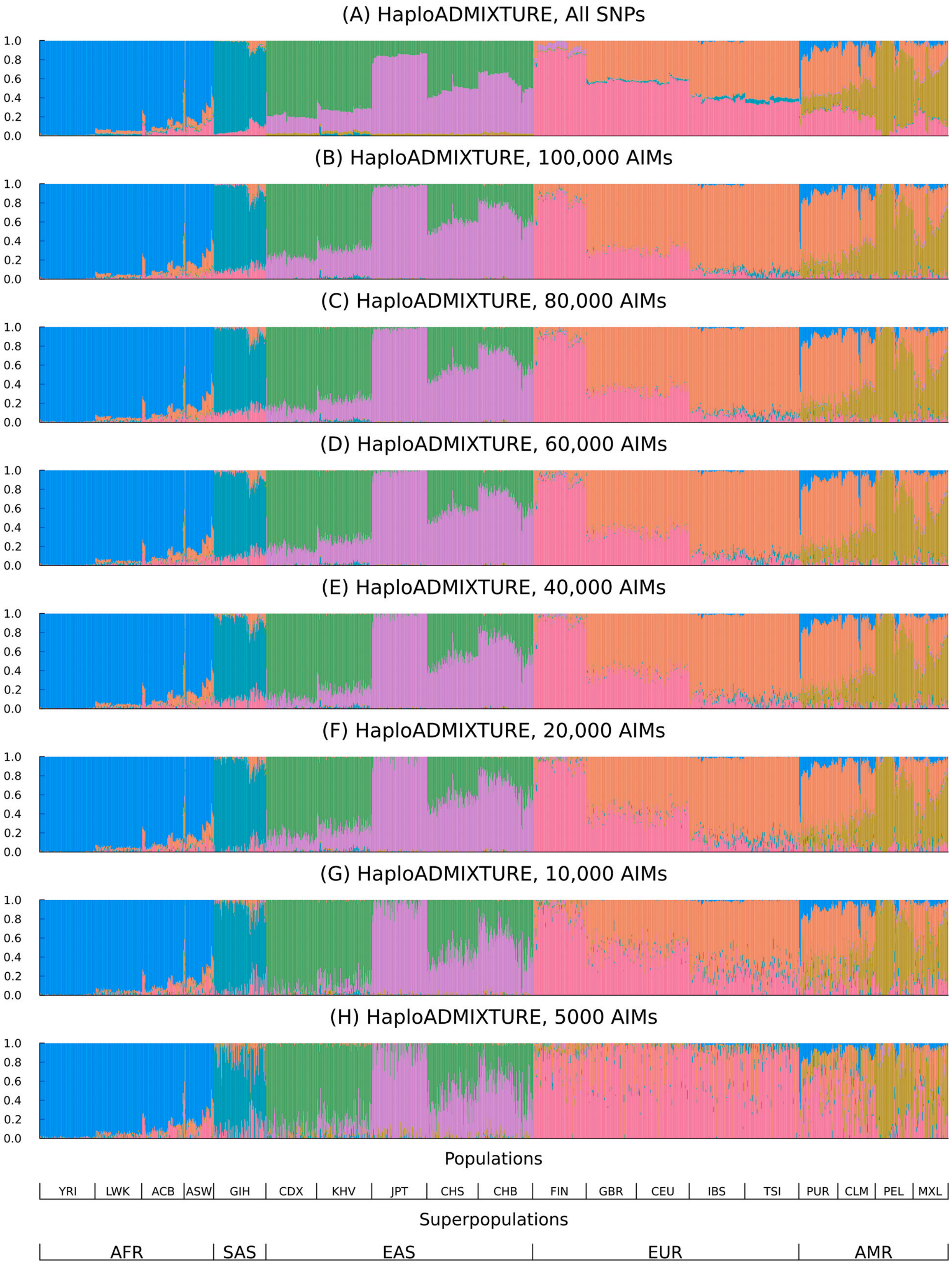

Fig. 2. Ancestry estimation of TGP samples using different numbers of AIMs with HaploADMIXTURE. The results are presented in stacked bar plots, where the y-axis indicates the proportion of total ancestry. The x-axis shows all samples arranged by population labels.

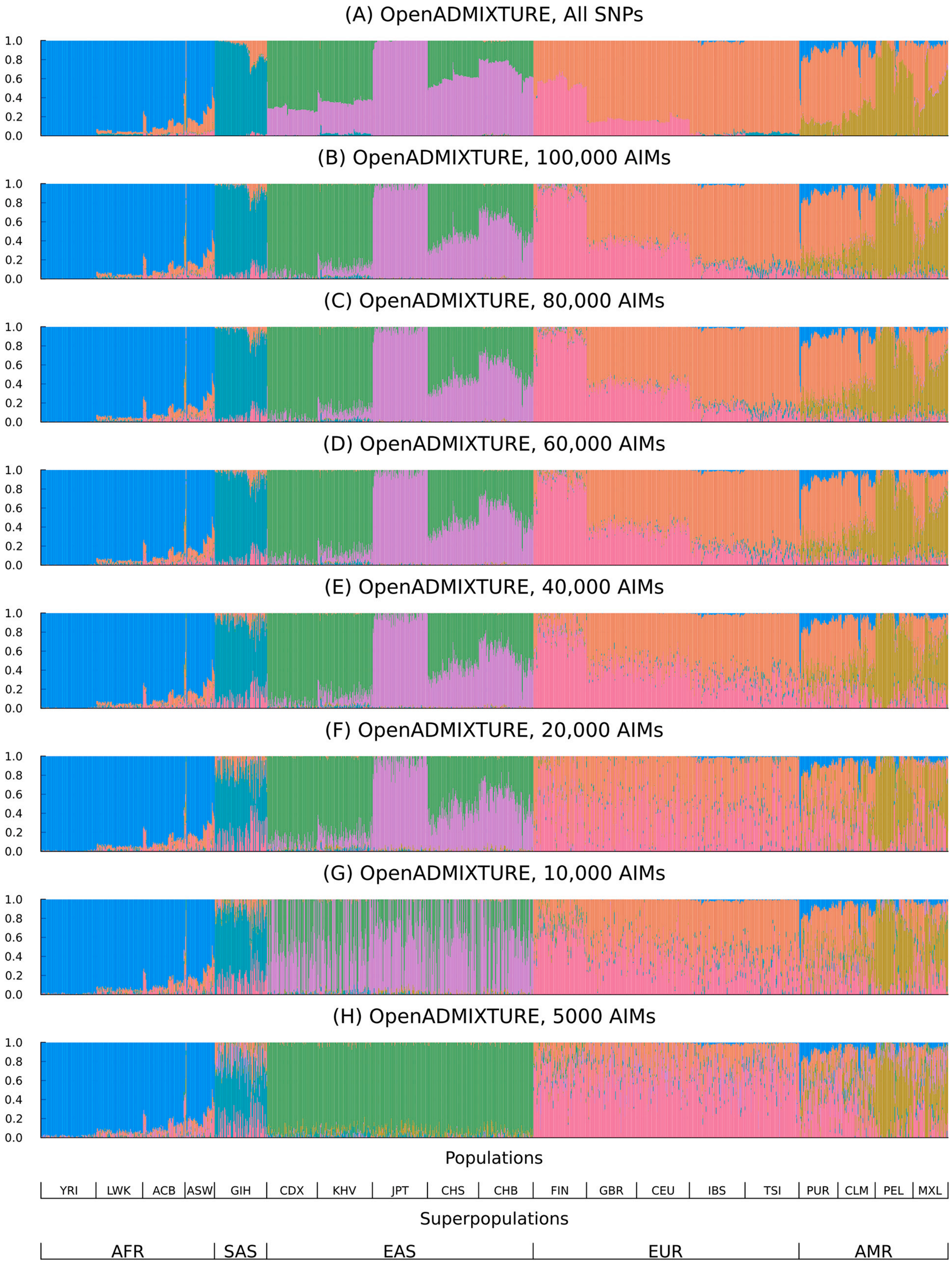

Fig. 3. Ancestry estimation of TGP data samples using different numbers of AIMs with OpenADMIXTURE. The results are presented in stacked bar plots, where the y-axis indicates the proportion of total ancestry. The x-axis shows all samples arranged by population labels.

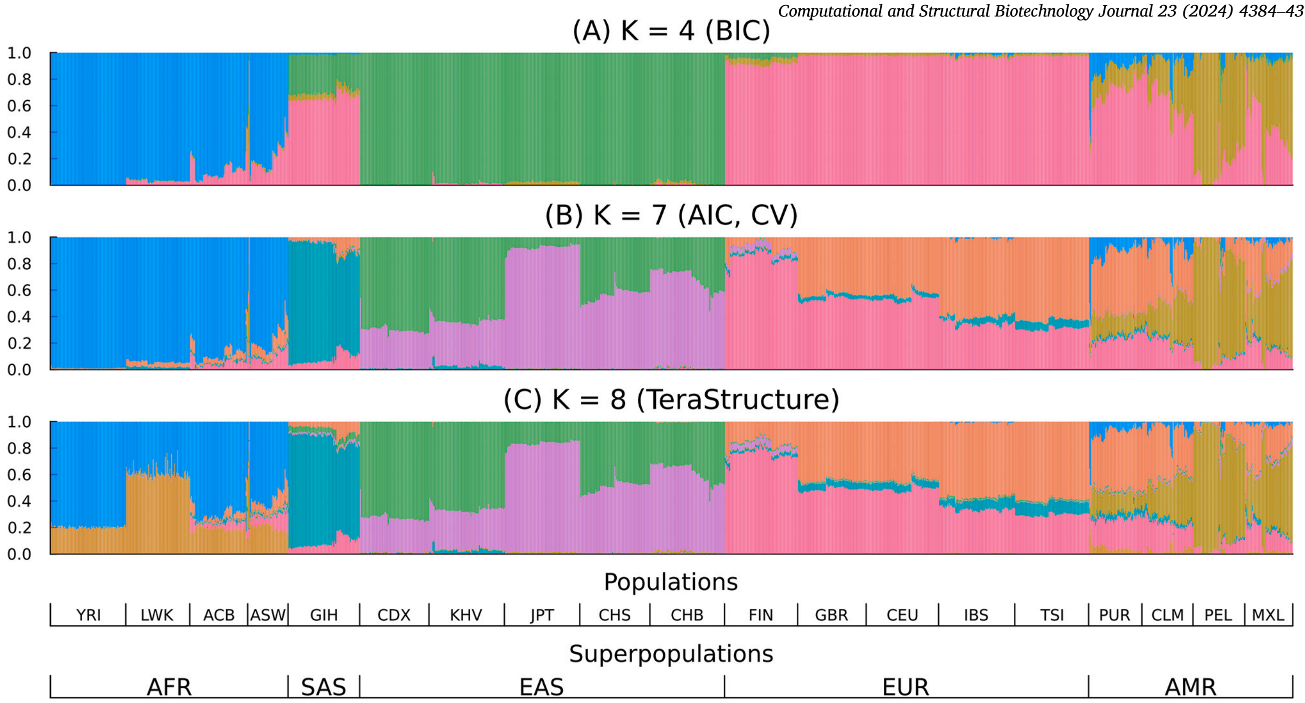

Fig. 4. Structure inferred for TGP data samples using HaploADMIXTURE for different K. (a) $K = 4$ as selected by Bayesian information criterion, (b) $K = 7$ as selected by cross-validation and Akaike information criterion, (c) $K = 8$ as selected by the validation likelihood method in TeraStructure. The results are presented in stacked bar plots where the y-axis indicates the proportion of total ancestry. The x-axis shows all samples arranged by population labels.

Table 5
Root-mean-square error of sparse K-means (SKFR) from the baseline for HaploADMIXTURE and OpenADMIXTURE on the TGP dataset. Root-mean-square error (RMSE) from baseline compares estimated admixture coefficients of SKFR to those estimated using all SNPs; the lower, the better.

SNPs	HaploADMIXTURE	OpenADMIXTURE
100,000	0.093	0.089
80,000	0.087	0.091
60,000	0.085	0.093
40,000	0.082	0.113
20,000	0.077	0.162
10,000	0.065	0.166
5000	0.132	0.181

ulations. In the case of HGDP under AIM selection, OpenADMIXTURE has trouble distinguishing between Middle-Eastern and European populations and adds a population to Africa. This anomaly is visible in Figure S4. HaploADMIXTURE with AIMs retains the power to distinguish the Middle-Eastern and European populations. For the HO dataset, HaploADMIXTURE performance with AIMs better mimics its performance with all SNPs than OpenADMIXTURE does in the same comparison. Tables 5, S19, and S20 display RMSE from the baseline of all SNPs for the TGP, HGDP, and HO datasets, respectively. For the TGP dataset, as we choose fewer AIMs, the mean silhouette tends to decrease, except for 10,000 and 5000 SNPs in OpenADMIXTURE. However, these exceptional cases yield poorer separation of populations than HaploADMIXTURE with all SNPs. This suggests that parsimony alone is an imperfect criterion for judging admixture estimation.

3.2.5. *Computational efficiency*

Given the computational improvements incorporated in HaploADMIXTURE, the analyses reported here finish in a reasonable amount of time. HaploADMIXTURE's cost per iteration with $S = 2$ SNPs per haplotype block is less than eight times that of OpenADMIXTURE. Given that the number of frequency parameters quadruples, it takes four times longer to compute gradients and Hessians. While the time for solving quadratic programs is still negligible for $\boldsymbol{Q}$, quadratic programming for $\boldsymbol{P}$ takes longer, comparable to the time needed to compute gradients and Hessians on a GPU. Since 16-threaded ADMIXTURE was 16 times slower than GPU-accelerated OpenADMIXTURE [8], HaploADMIXTURE's per-iteration performance is still faster than that of ADMIXTURE. Balanced against these gains is the fact that the number of iterations until convergence increases. This reflects the greater complexity of the likelihood, the increased number of parameters, and the cost of parameter splitting by the MM principle.

Table S21 shows the average runtime using five random initial points for the TGP, HGDP, and HO datasets ignoring AIMs. Despite requiring more iterations to converge, HaploADMIXTURE takes less than 16 times longer than OpenADMIXTURE. Because runtime is proportional to the number of blocks B of SNPs employed, preprocessing with AIM selection to reduce B is recommended if speed is critical. For example, on the TGP data, it takes 2 minutes for sparse K-means to select 100,000 AIMs, and then another 12 minutes to run HaploADMIXTURE on the filtered dataset, for a total of just 14 minutes. Even so, running on AIMs yields admixture coefficients comparable to running on the full set of 1.8 million SNPs. The latter more onerous computations take 2 hours and 8 minutes. If one opts to preselect AIMs by sparse K-means, the time needed for SKFR in HaploADMIXTURE is not much different from that for OpenADMIXTURE. Indeed, the speed of the SKFR algorithm is minimally affected by the switch to haplotypes. SKFR and HaploADMIXTURE directly operate on PLINK BED-formatted data, so the total memory footprint of each is less than twice the size of the BED file.

3.2.6. *Large-scale analysis of the UK biobank data*

For the 488,154 individuals selected from the UKB dataset, we undertook supervised inference of population structure using the haplotype frequencies $\hat{\boldsymbol{P}}$ obtained from the TGP dataset. Our analysis is limited to $K = 7$ subpopulations and the SNPs shared by the TGP and UKB datasets.

Clustering performance is based on three sets of labels. The first set (L1) uses 20 raw self-identified ancestry labels, excluding "do not know" and "prefer not to answer." The second set (L2) uses 8 of the 20 labels:

Table 6

Performance comparison of HaploADMIXTURE, OpenADMIXTURE, SCOPE, and TeraStructure on the TGP dataset. Performance is measured by the mean silhouette coefficient of the population labels on the space of estimated admixture coefficients, $\boldsymbol{Q}$; the higher, the better. The best value in the mean silhouette is in *italics*; these range over $[-1, 1]$.

SNPs	HaploADMIXTURE	OpenADMIXTURE	SCOPE	TeraStructure
	Continental labels			
1,854,622	0.606	0.591	0.528	*0.671*
100,000	0.558	0.524		
80,000	0.540	0.525		
60,000	0.541	0.526		
40,000	0.526	0.533		
20,000	0.528	0.481		
10,000	0.522	0.500		
5000	0.521	0.626		
	Regional labels			
1,854,622	*0.423*	0.413	0.418	0.335
100,000	0.360	0.347		
80,000	0.353	0.329		
60,000	0.335	0.296		
40,000	0.317	0.225		
20,000	0.277	0.147		
10,000	0.206	0.035		
5000	0.083	0.025		

Table 7

Performance comparison of HaploADMIXTURE, OpenADMIXTURE, and SCOPE on the UKB dataset. Performance is measured by the mean silhoutte coefficient of the population labels on the space of estimated admixture coefficients, $\boldsymbol{Q}$; the higher, the better. The best value in the mean silhouette is in *italics*; these range over [-1, 1]. TeraStructure does not run within 24 hours.

HaploADMIXTURE Unsupervised	HaploADMIXTURE Supervised	OpenADMIXTURE	SCOPE
L3 - Continental labels			
0.540	*0.991*	0.471	0.257
L2 - Regional labels			
0.037	-0.284	0.030	0.019
L1 - Detailed labels			
-0.013	-0.303	-0.064	-0.086

British, Irish, Indian, Pakistani, Bangladeshi, Caribbean, African, and Chinese, removing mixed and uncertain population labels. Finally, for the third label set (L3), the 8 groups are merged by continent and reduce to British-Irish, Indian-Pakistani-Bangladeshi, Caribbean-African, and Chinese.

Table 7 shows the clustering performance of the resulting admixture coefficients. In unsupervised inference, HaploADMIXTURE consistently separates the different sets of ancestry labels the best, followed by OpenADMIXTURE, and then SCOPE. Supervised HaploADMIXTURE run using the $\hat{P}$ of the TGP performs significantly better on continental labels (L3) because TGP contains a substantial amount of relevant continental information. However, because haplotype frequency estimates rely on only 1718 individuals, supervised HaploADMIXTURE falters in distinguishing fine-grained populations compared to unsupervised HaploADMIXTURE.

Supervised inference is advantageous in that it takes much less time, namely 4 hours on an Nvidia L4 GPU with 24 GB memory. In contrast unsupervised inference takes around 11 hours. Unsupervised OpenADMIXTURE takes 6 hours. To its credit, SCOPE's randomized linear algebra takes just 1 hour and 10 minutes on a 72-core CPU instance.

4. Discussion

This paper introduces a technique for global ancestry estimation that converts linkage disequilibrium from a liability to an asset. Our program HaploADMIXTURE exploits multithreading, GPU acceleration, and sparse K-means clustering to identify ancestry informative haplotypes. Although these advances also appear in OpenADMIXTURE, our earlier upgrade of the ADMIXTURE [6] software, they require substantial modification to handle haplotypes. For instance, in the construction of AIMs, sparse K-means must now operate on haplotypes rather than SNPs. Likelihood calculation becomes more complicated because of increased phase ambiguity. Nevertheless, these technical hurdles can be overcome with computational speed and memory demands on a par with or better than that of the original ADMIXTURE. Computation times scale linearly in the number of haplotype blocks. To keep computational costs in check, our haplotypes span just two SNPs. Even with this limitation, we see substantial gains in ancestry estimation precision. Extending haplotype blocks to include more than two SNPs is theoretically possible and would further increase information content, particularly for those regions of the human genomes showing little recombination. However, this extension would quickly hit a combinatorial wall in computing the 2^S haplotype frequencies given S SNPs per block. The greater phase ambiguity encountered would complicate computer code and slow the convergence of recursive quadratic programming, the optimization engine in HaploADMIXTURE.

The admixture coefficients delivered by HaploADMIXTURE demonstrate a good separation of populations at the continental and regional levels in both real and simulated datasets. The other admixture programs tested often perform well on one level and poorly on the other. The admixture estimates from HaploADMIXTURE are more accurate than the competition as measured by mean square prediction error. In our experience, cross-validation and AIC produce reasonable estimates of the number of ancestral populations K. AIC is much faster than cross-validation. It will be interesting to see whether Bayesian or algebraic methods can be adapted to exploit haplotypes. The algebraic program SCOPE relies on alternating least squares, so its adaptation would require passing from matrix to tensor decompositions.

Estimation of admixture proportions given known populations and known haplotype frequencies is possible with HaploADMIXTURE, as shown in Section 3.2.6. One simply fixes $\boldsymbol{P}$ and updates only $\boldsymbol{Q}$. This simplification is also invoked in the time-consuming process of cross-validation. For best results, partial maximization requires curating the most informative pairs of SNPs in large population panels. Partial maximization is a parameter-separated convex problem that is easily solved on biobank-scale data.

Of course, estimation of human ancestry is fraught with interpretation pitfalls, errors in assumptions, and, for ancient populations, lack of relevant data. The issues are carefully covered in Pritchard's online book [37]. See especially Chapters 3.1 and 3.2. For modern populations, readers should keep in mind the utility of long conserved haplotype blocks in assigning ancestry. Chromosome painting identifies these blocks and can be accomplished rapidly as part of haplotying [38].

Advances in technology and the rapid expansion of human biobanks have pushed software development to the top of the agenda in genomics. The "All of Us" [39] databank contains more than 400,000 of individuals, of whom 250,000 are whole genome sequenced. The UK Biobank contains more than 488,000 genotyped individuals. Accurate and scalable adjustment for ancestry is a supremely important task in understanding these data.

Modeling haplotypes adds vital information in ancestry analysis, yields more precise estimates of admixture coefficients, and distinguishes subpopulations better. Our GPU-accelerated implementation, HaploADMIXTURE, maintains computational efficiency while improving accuracy of admixture coefficients and distinguishing subtle population variation better. HaploADMIXTURE is a thoughtful extension of OpenADMIXTURE, the open-source upgrade of the widely-used ADMIXTURE software. HaploADMIXTURE builds on Julia's high-performance computing environment and leverages potent OpenMendel tools. As HaploADMIXTURE is expanded and improved over time, we hope that it will ultimately receive the wide acceptance already enjoyed by ADMIXTURE.

Web resources

- OpenADMIXTURE, https://github.com/OpenMendel/OpenADMIXTURE.jl.
- Sparse K-means with Feature Ranking, https://github.com/kose-y/SparseKmeansFeatureRanking.jl.
- SnpArrays, https://github.com/OpenMendel/SnpArrays.jl.
- SCOPE, https://github.com/sriramlab/SCOPE.

CRediT authorship contribution statement

Seyoon Ko: Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis. **Eric M. Sobel:** Writing – review & editing, Writing – original draft, Validation, Funding acquisition. **Hua Zhou:** Writing – review & editing, Writing – original draft, Resources, Funding acquisition. **Kenneth Lange:** Writing – review & editing, Writing – original draft, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was partially funded by grants from the National Institute of General Medical Sciences (R35GM141798, EMS, HZ, and KL), the National Human Genome Research Institute (R01HG006139, EMS, HZ, and KL), and the National Science Foundation (DMS-2054253 and IIS-2205441, HZ).

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.csbj.2024.11.043.

Data availability

The HaploADMIXTURE package can be found at https://github.com/OpenMendel/HaploADMIXTURE.jl. The code for the experiments and instructions to download publicly available data can be found at https://github.com/kose-y/HaploADMIXTURE-resources.

References

[1] Novembre J, Johnson T, Bryc K, Kutalik Z, Boyko AR, Auton A, et al. Genes mirror geography within Europe. Nature 2008;456:98–101.

[2] Price AL, Patterson NJ, Plenge RM, Weinblatt ME, Shadick NA, Reich D. Principal components analysis corrects for stratification in genome-wide association studies. Nat Genet 2006;38:904–9.

[3] Li C. Population subdivision with respect to multiple alleles. Ann Hum Genet 1969;33:23–9.

[4] Knowler WC, Williams R, Pettitt D, Steinberg A. Gm3; 5, 13, 14 and type 2 diabetes mellitus: an association in American Indians with genetic admixture. Am J Hum Genet 1988;43:520.

[5] Marchini J, Cardon LR, Phillips MS, Donnelly P. The effects of human population structure on large genetic association studies. Nat Genet 2004;36:512–7.

[6] Alexander DH, Novembre J, Lange K. Fast model-based estimation of ancestry in unrelated individuals. Genome Res 2009;19:1655–64.

[7] Zhou H, Alexander D, Lange K. A quasi-Newton acceleration for high-dimensional optimization algorithms. Stat Comput 2011;21:261–73.

[8] Ko S, Chu BB, Peterson D, Okenwa C, Papp JC, Alexander DH, et al. Unsupervised discovery of ancestry-informative markers and genetic admixture proportions in biobank-scale datasets. Am J Hum Genet 2023;110:314–25.

[9] Zhang Z, Lange K, Xu J. Simple and scalable sparse k-means clustering via feature ranking. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H, editors. Advances in Neural Information Processing Systems, vol. 33. Curran Associates, Inc.; 2020. p. 10148–60.

[10] Pritchard JK, Stephens M, Donnelly P. Inference of population structure using multilocus genotype data. Genetics 2000;155:945–59.

[11] Raj A, Stephens M, Pritchard JK. fastSTRUCTURE: variational inference of population structure in large SNP data sets. Genetics 2014;197:573–89.

[12] Gopalan P, Hao W, Blei DM, Storey JD. Scaling probabilistic models of genetic variation to millions of humans. Nat Genet 2016;48:1587–90.

[13] Chiu AM, Molloy EK, Tan Z, Talwalkar A, Sankararaman S. Inferring population structure in biobank-scale genomic data. Am J Hum Genet 2022;109:727–37.

[14] Cabreros I, Storey JD. A likelihood-free estimator of population structure bridging admixture models and principal components analysis. Genetics 2019;212:1009–29.

[15] Lawson DJ, Hellenthal G, Myers S, Falush D. Inference of population structure using dense haplotype data. PLoS Genet 2012;8:e1002453.

[16] Brown R, Pasaniuc B. Enhanced methods for local ancestry assignment in sequenced admixed individuals. PLoS Comput Biol 2014;10:e1003555.

[17] Guan Y. Detecting structure of haplotypes and local ancestry. Genetics 2014;196:625–42.

[18] Pakstis AJ, Fang R, Furtado MR, Kidd JR, Kidd KK. Mini-haplotypes as lineage informative SNPs and ancestry inference SNPs. Eur J Hum Genet 2012;20:1148–54.

[19] Hunter DR, Lange K. A tutorial on MM algorithms. Am Stat 2004;58:30–7.

[20] Lange K. MM optimization algorithms. SIAM; 2016.

[21] Alexander DH, Lange K. Enhancements to the admixture algorithm for individual ancestry estimation. BMC Bioinform 2011;12:1–6.

[22] Akaike H. Information theory and an extension of the maximum likelihood principle. In: Selected papers of Hirotugu Akaike. Springer; 1998. p. 199–213.

[23] Chi JT, Chi EC, Baraniuk RG. k-POD: a method for k-means clustering of missing data. Am Stat 2016;70:91–9.

[24] Lloyd S. Least squares quantization in PCM. IEEE Trans Inf Theory 1982;28:129–37.

[25] Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MA, Bender D, et al. PLINK: a tool set for whole-genome association and population-based linkage analyses. Am J Hum Genet 2007;81:559–75.

[26] Zhou H, Sinsheimer JS, Bates DM, Chu BB, German CA, Ji SS, et al. OpenMendel: a cooperative programming project for statistical genetics. Hum Genet 2020;139:61–71.

[27] Behr AA, Liu KZ, Liu-Fang G, Nakka P, Ramachandran S. Pong: fast analysis and visualization of latent clusters in population genetic data. Bioinformatics 2016;32:2817–23.

[28] Dunning I, Huchette J, Lubin M. JuMP: a modeling language for mathematical optimization. SIAM Rev 2017;59:295–320.

[29] Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 1987;20:53–65.

[30] 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. Nature 2012;491:56–65.

[31] 1000 Genomes Project Consortium, et al. A global reference for human genetic variation. Nature 2015;526:68.

[32] Cann HM, De Toma C, Cazes L, Legrand M-F, Morel V, Piouffre L, et al. A human genome diversity cell line panel. Science 2002;296:261–2.

[33] Cavalli-Sforza LL. The Human Genome Diversity Project: past, present and future. Nat Rev Genet 2005;6:333–40.

[34] Lazaridis I, Patterson N, Mittnik A, Renaud G, Mallick S, Kirsanow K, et al. Ancient human genomes suggest three ancestral populations for present-day europeans. Nature 2014;513:409–13.

[35] Bycroft C, Freeman C, Petkova D, Band G, Elliott LT, Sharp K, et al. The UK Biobank resource with deep phenotyping and genomic data. Nature 2018;562:203–9.

[36] Balding DJ, Nichols RA. A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. Genetica 1995;96:3–12.

[37] Pritchard J. An owner's guide to the human genome: an introduction to human population genetics, variation and disease. https://web.stanford.edu/group/pritchardlab/HGbook.html, 2024. [Accessed 25 November 2024].

[38] Chu BB, Sobel EM, Wasiolek R, Ko S, Sinsheimer JS, Zhou H, et al. A fast data-driven method for genotype imputation, phasing and local ancestry inference: MendelImpute.jl. Bioinformatics 2021;37:4756–63.

[39] All of Us Research Program Investigators. The "All of Us" research program. N Engl J Med 2019;381:668–76.