Unsupervised discovery of ancestry-informative markers and genetic admixture proportions in biobank-scale datasets

Authors

Seyoon Ko, Benjamin B. Chu, Daniel Peterson, ..., Eric M. Sobel, Hua Zhou, Kenneth L. Lange

Correspondence

esobel@ucla.edu

Ko et al., 2023, The American Journal of Human Genetics *110*, 314–325
February 2, 2023
https://doi.org/10.1016/j.ajhg.2022.12.008

CellPress

ARTICLE

Unsupervised discovery of ancestry-informative markers and genetic admixture proportions in biobank-scale datasets

Seyoon Ko,[1,2] Benjamin B. Chu,[1,3] Daniel Peterson,[4] Chidera Okenwa,[5] Jeanette C. Papp,[6] David H. Alexander,[7] Eric M. Sobel,[1,6,*] Hua Zhou,[1,2] and Kenneth L. Lange[1,6,8]

Summary

Admixture estimation plays a crucial role in ancestry inference and genome-wide association studies (GWASs). Computer programs such as ADMIXTURE and STRUCTURE are commonly employed to estimate the admixture proportions of sample individuals. However, these programs can be overwhelmed by the computational burdens imposed by the 10^5 to 10^6 samples and millions of markers commonly found in modern biobanks. An attractive strategy is to run these programs on a set of ancestry-informative SNP markers (AIMs) that exhibit substantially different frequencies across populations. Unfortunately, existing methods for identifying AIMs require knowing ancestry labels for a subset of the sample. This supervised learning approach creates a chicken and the egg scenario. In this paper, we present an unsupervised, scalable framework that seamlessly carries out AIM selection and likelihood-based estimation of admixture proportions. Our simulated and real data examples show that this approach is scalable to modern biobank datasets. OpenADMIXTURE, our Julia implementation of the method, is open source and available for free.

Introduction

Discovery of ancestral groups or population structure by genetic means is of inherent interest for both private genealogies and public population histories.[1] In addition, genetic ancestry adjustment is a necessary prelude for genome-wide association studies (GWASs)[2] seeking DNA sites that influence medical or anthropomorphic traits. Without this safeguard, population stratification can lead to a false association between a trait and a SNP predictor.[3–5] Ancestry adjustment can be handled by adding a few principal components (PCs) of the SNP genotype matrix as trait predictors. Alternatively, one can substitute admixture proportions (coefficients) in place of PCs. Because admixture coefficients are the proportions of an individual's ancestry from each of several founding populations, they are usually more interpretable than PCs. In some cases, PCs have been shown to be inefficient for correcting biases.[6]

Estimation of ancestry coefficients is carried out simultaneously with maximum likelihood estimation of allele frequencies when the underlying populations are latent rather than explicit. ADMIXTURE[7] is a widely used likelihood-based software. ADMIXTURE directly maximizes the likelihood of the genotype matrix via alternating sequential quadratic programming with quasi-Newton acceleration.[8] Another popular package, STRUCTURE,[9] implements Bayesian inference; recent extensions of the Bayesian approach include fastStructure[10] and TeraStructure.[11] The EIGENSTRAT software[2] is the primary vehicle for PC extraction from the genotype matrix. One can also achieve dimensionality reduction by approximating a low-rank linear subspace of the row space of the admixture proportion matrix and then performing matrix decomposition via alternating least squares, as is implemented in the ALStructure software.[12] Matrix decomposition is further accelerated in the SCOPE software[13] by invoking randomized linear algebra and routines specifically designed for accessing compressed versions of the genotype matrix.

As the size of genomic data grows, these methods suffer. In particular, most of the methods fail on the UK Biobank data,[14] which contain $\{0, 1, 2\}$ genotypes on about half a million British individuals across millions of SNPs. The genotype matrix in PLINK format alone requires around 70 GB of storage. One of the few software programs that is capable of handling these data is SCOPE,[13] which avoids holding large intermediate matrices in memory. However, SCOPE's preprocessing of the genotype matrix to speed up matrix multiplication still requires at least 250 GB of RAM (random access memory).

One can make ancestry estimation more efficient by limiting analysis to ancestry-informative markers (AIMs).[15,16] Early AIM sets included tens to hundreds of AIMs.[17–20] Even at this crude scale, it is possible to recover admixture coefficients that correlate well (74%–92%) with those delivered by the full set of SNPs.[21] AIM-based methods exploit F statistics, absolute allele frequency differences, principal component loadings, and informativeness in

[1]Department of Computational Medicine, University of California, Los Angeles, Los Angeles, CA 90095, USA; [2]Department of Biostatistics, University of California, Los Angeles, Los Angeles, CA 90095, USA; [3]Department of Biomedical Data Science, Stanford University, Stanford, CA 94305, USA; [4]Department of Mathematics, Brigham Young University, Provo, UT 84602, USA; [5]Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720, USA; [6]Department of Human Genetics, University of California, Los Angeles, Los Angeles, CA 90095, USA; [7]X Development LLC, Mountain View, CA 94043, USA; [8]Department of Statistics, University of California, Los Angeles, Los Angeles, CA 90095, USA
*Correspondence: esobel@ucla.edu
https://doi.org/10.1016/j.ajhg.2022.12.008.

ancestry assignment.[22,23] To their disadvantage, most AIM selection methods are supervised and based on self-reported labels. In biobank-scale data, such labels should be viewed with suspicion.

In the current paper, we advocate first selecting AIMs in an unsupervised way through a sparse K-means clustering algorithm.[24] We will refer to this algorithm by the acronym SKFR (sparse K-means with feature ranking). SKFR performs hard clustering and feature selection jointly and is scalable to biobank data. Given the selected AIMs, we run a version of ADMIXTURE[7] that leverages the computational advances of the Julia programming language.[25] We call this package OpenADMIXTURE, in part because of its open-source status. OpenADMIXTURE incorporates both SKFR and admixture estimation, supports multithreading, and acts directly on the input genotype matrix. The maximum memory demand is less than 120% the size of the input genotype file. For example, our analysis of the UK Biobank data with 500,000 individuals and 600,000 SNPs requires only 73 GB of RAM versus the 250 GB required by SCOPE. OpenADMIXTURE also supports graphics processing unit (GPU) acceleration. Runtimes and results of OpenADMIXTURE are comparable to those of SCOPE but within the RAM limitations of more typical computers. Furthermore, OpenADMIXTURE retains the advantages of a likelihood-based analysis. SKFR is generally useful in feature selection across a wide variety of clustering applications beyond genetics. An independent Julia-based package to efficiently perform SKFR on general datasets is available.

Material and methods

Sparse K-means with feature ranking (SKFR)

SKFR selects and ranks a predetermined number of features that drive K-means clustering.[24] Feature selection and clustering are intertwined. In our case, features are standardized SNP genotypes displayed in an $I \times J$ matrix $\boldsymbol{X} = (x_{ij})$. Rows correspond to samples and columns to features. Standardization of columns to have mean 0 and variance 1 puts all features on the same footing. Given a fixed number of clusters K, the goal is to assign each individual i and its corresponding row $\boldsymbol{x}_i^T$ of $\boldsymbol{X}$ to the cluster C_k minimizing the loss

$$f(\boldsymbol{B}, \Theta) = \sum_{k=1}^{K} \sum_{i \in C_k} \|\boldsymbol{x}_i - \boldsymbol{\theta}_k\|_2^2 = \sum_{i=1}^{n} \min_{1 \le k \le K} \|\boldsymbol{x}_i - \boldsymbol{\theta}_k\|_2^2 = \|\boldsymbol{X} - \boldsymbol{B}\Theta\|_{\mathrm{F}}^2. \quad \text{(Equation 1)}$$

Here, the matrix $\boldsymbol{B} \in \{\boldsymbol{M} \in \{0,1\}^{I \times K} : \boldsymbol{M}\boldsymbol{1}_K = \boldsymbol{1}_I\}$ conveys cluster membership, the matrix $\Theta \in \mathbb{R}^{K \times J}$ conveys cluster centers, and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. The k-th row $\boldsymbol{\theta}_k^T \in \mathbb{R}^J$ of Θ is the center of cluster C_k. In SKFR with sparsity level S, at most S columns of Θ are allowed to be nonzero. The SKFR procedure (see Algorithm 1A in appendix A) cycles through the following three steps until convergence: (1) update the cluster centers, (2) rank and select features according to their contribution to the loss, and (3) re-assign samples to clusters according to the selected features. In Algorithm 1A, the information criterion h_j measures the drop in the loss when feature j is designated informative. The clusters are initialized by the K-means++ scheme.[26] Initial cluster centers emerge after steps 2 and 3 are performed on the standardized matrix $\boldsymbol{X}$. The section "missing genotypes" sketches a modification of the algorithm to handle missing data.

On convergence, the SKFR algorithm yields (1) a ranked list L of selected AIMs, (2) hard clustering assignments $\boldsymbol{B}$ of each sample to one and only one cluster, and (3) cluster centers Θ. A new set of PLINK files containing only the selected AIMs is generated via the SnpArrays Julia package.

Missing genotypes

Genotype data often include missing values. In practice, genotype imputation precedes GWAS. However, imputing genotypes at biobank scale is extremely resource and computation intensive, traditionally taking days to months on a cluster. Modern software[27,28] has reduced this bottleneck. Following Chi et al.,[29] we extend SKFR to incorporate missing data in a mathematically principled way. Let $\Omega \subset \{1,\ldots,I\} \times \{1,\ldots,J\}$ denote the subset of indices corresponding to the observed entries of $\boldsymbol{X}$. In this notation, the modified loss is

$$f_\Omega(\boldsymbol{B}, \Theta) = \|P_\Omega(\boldsymbol{X} - \boldsymbol{B}\Theta)\|_{\mathrm{F}}^2, \quad \text{(Equation 2)}$$

where $P_\Omega(\boldsymbol{M})$ zeros all entries of a matrix $\boldsymbol{M}$ not in Ω.

The quickest route to minimization of the loss passes through the majorization-minimization (MM) principle.[30–32] At iteration n of a search, we construct the surrogate function $g(\boldsymbol{B}, \Theta | \boldsymbol{B}_n, \Theta_n) = \|\boldsymbol{Y}_n - \boldsymbol{B}\Theta\|_{\mathrm{F}}^2$ majorizing the loss, where $\boldsymbol{Y}_n = P_\Omega(\boldsymbol{X}) + P_{\Omega^\perp}(\boldsymbol{B}_n\Theta_n)$ and $\Omega^\perp$ denotes the set of missing values. In other words, we leave observed values untouched and impute missing values by their predicted values on the basis of the centers of their current cluster assignments. The MM principle guarantees that minimizing the surrogate reduces the loss. This monotonic algorithm is summarized in Algorithm 1B in appendix A. The current code differs from the code presented in Zhang et al.[24] by standardizing the genotype matrix beforehand with the observed values rather than repeatedly standardizing on the fly with both the observed and imputed values. The current implementation is more efficient and still mathematically sound.

Estimation of admixture proportions

In contrast to hard clustering, soft clustering estimates the probability of a sample belonging to each of the K clusters. Soft clustering algorithms like ADMIXTURE[7] better account for ambiguities than hard clustering and in GWASs more realistically adjust for population structure. In this section, we describe a Julia implementation of ADMIXTURE that capitalizes on parallel processing and GPU support. Recall that ADMIXTURE simultaneously estimates a population-specific allele frequency matrix $\boldsymbol{P} \in \mathbb{R}^{K \times J}$ and an individual-specific admixture matrix $\boldsymbol{Q} \in \mathbb{R}^{K \times I}$ by maximizing the log likelihood

$$\ell(\boldsymbol{P}, \boldsymbol{Q}) = \sum_{i,j} \left[x_{ij} \log\left(\sum_k p_{kj} q_{ki} \right) + (2 - x_{ij}) \log\left(1 - \sum_k p_{kj} q_{ki} \right) \right]. \quad \text{(Equation 3)}$$

Here, each raw genotype x_{ij} follows a Binomial$\left(2, \sum_k p_{kj} q_{ki}\right)$ distribution, where the parameters satisfy the constraints $\sum_{k=1}^K q_{ki} = 1$ and $q_{ki}, p_{kj} \in [0,1]$. Maximization is carried out by block ascent, alternating updates of $\boldsymbol{P}$ and $\boldsymbol{Q}$ by sequential quadratic programming with quasi-Newton acceleration.[8]

Given an objective function $f(\boldsymbol{x})$, sequential quadratic programming finds the next iterate $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta$ by minimizing the second-order approximation

$$f(\boldsymbol{x}_n + \Delta) \approx f(\boldsymbol{x}_n) + df(\boldsymbol{x}_n)\Delta + \frac{1}{2}\Delta^T d^2 f(\boldsymbol{x}_n)\Delta$$

with respect to Δ subject to relevant constraints. Here, $df(\boldsymbol{x})$ is the first differential (transposed gradient) and $d^2 f(\boldsymbol{x})$ is the second differential (Hessian) of $f(\boldsymbol{x})$. With linear constraints and upper and lower bounds, one can exploit a standard pivoting strategy to solve this quadratic program.[33] ADMIXTURE accomplishes precisely this with the objective equal to the negative log likelihood. In the block ascent updates, the SNP-specific allele frequencies and individual-specific admixture proportions are parameters that can be separated. This results in an overall computational complexity of $O[(I+J)K^3]$ for the quadratic programs, which is negligible compared to the bottleneck of $O(IJK^2)$ in computing Hessians, as $K \ll I, J$. Admixture proportions are initialized by five iterations of FRAPPE's EM (expectation maximization) algorithm.[34]

OpenADMIXTURE leverages Julia to achieve higher performance. The core computations of sequential quadratic programming now exploit tiling to maximize locality and avoid cache misses. Users can choose to offload most computations to graphics processing units (GPUs) for further speedup. OpenADMIXTURE's default setting declares convergence when the relative change in loglikelihoods is less than 10^{-7}. Supervised inference is possible by fixing $\boldsymbol{P}$ and updating only $\boldsymbol{Q}$ or fixing $\boldsymbol{Q}$ and updating only $\boldsymbol{P}$. The former is pertinent when admixture proportions are sought and predefined allele frequencies from reference populations are used. The latter is pertinent when allele frequencies are sought for the reference populations whose admixture coefficients are fixed.

Software input and output

Our OpenADMIXTURE package internally runs a pipeline of SKFR and then admixture estimation. As mentioned above, SKFR is also available as a stand-alone package for use on general datasets (see "data and code availability"). It is also possible to run OpenADMIXTURE with all available SNPs and bypass AIM discovery through SKFR. Given these considerations the input and output conventions adopted by OpenADMIXTURE are the following.

For SKFR

Input: a single set (`bed`, `fam`, `bim`) or collection of PLINK binary files, the number of clusters, and the number of AIMs.

Output: a single set of PLINK binary files containing only the selected AIMs and a file containing hard clustering results, where each row indicates the cluster to which a sample is assigned. A filtered PLINK file containing only the AIMs is optional.

For admixture estimation

Input: a single set (`bed`, `fam`, `bim`) or collection of PLINK binary files, possibly filtered to contain only the selected AIMs under SKFR, and the number of clusters.

Output: a `P` file where each row indicates the cluster-specific allele frequencies of an AIM and a `Q` file where each row indicates the estimated admixture proportions of an individual.

Selection of the number of clusters K

To choose the number of clusters K, the gap statistic of Tibshirani et al.[35] is handy. As a permutation test, the gap statistic requires running SKFR with different values of K, and samples revised by randomly shuffling genotypes across each SNP.

Warm start for a path with different sparsity level S

It is often desirable to explore a variety of AIM sparsity levels S on the same dataset. This can be done efficiently by starting with the highest level $S_{\max}$ desired and gradually decreasing S. The results from a given S are then invoked to warm start computations at the next lower level of S. OpenADMIXTURE's ranking of AIMs facilitates this tactic.

Further computational tactics

We directly exploit the structure of the PLINK `bed` format[36] to reduce memory usage through OpenMendel's[37] package SnpArrays. Further tactics that improve computational efficiency, such as initialization, recursive tiling for cache efficiency,[38] multithreading, and GPU acceleration[39] are discussed in the supplemental methods, A.

Performance evaluation

Permutation matching of clusters

Clustering results derived from two separate algorithms can be compared by various statistics. Any pertinent statistic should be invariant under permutation of cluster labels and match similar clusters. We carry out matching following the approach of Behr et al.[40] The similarity between cluster m of $\boldsymbol{Q}^1$ and cluster n of $\boldsymbol{Q}^2$ is quantified by

$$\mathcal{J}(q_m^1, q_n^2) = 1 - \sqrt{\frac{\sum_{i=1}^{I}\left(q_{mi}^1 - q_{ni}^2\right)^2}{2|N^*|}},$$

where $N^* = \{i : q_{mi}^1 + q_{ni}^2 > 0\}$. Cluster matching can be formulated as an assignment problem maximizing the criterion $\sum_{m,n} x_{mn}\mathcal{J}(q_m^1, q_n^2)$ subject to the constraints $x_{mn} \in \{0,1\}$ and $\sum_k^K x_{km} = \sum_k^K x_{nk} = 1$. In practice, the domain of x_{mn} is relaxed to the unit interval, and the problem is solved by linear programming via JuMP,[41] Julia's mathematical optimization package.

Visualization

We visualize estimates for admixture proportions as stacked bar plots. The clusters in each run are matched for easy comparison. To determine the order of samples, we rely on hierarchical clustering with complete linkage based on the OpenADMIXTURE $\boldsymbol{Q}$ estimates. The samples are ordered within each population, and the populations are ordered on the basis of hierarchical clustering of cluster centers. The same is done for superpopulations whenever applicable.

Real datasets

Our evaluation of OpenADMIXTURE relies on four independent datasets: 1000 Genomes Project (TGP),[42,43] Human Genome Diversity Project (HGDP),[44,45] Human Origins (HO),[46] and UK Biobank (UKB, retrieved under Project ID: 48152) in compliance with the data use agreements. The TGP dataset consists of the 2012-01-31 Omni Platform genotypes confined to unrelated individuals with at least a 95% genotyping success rate and SNPs with at least a 1% minor allele frequency (MAF). The filtered dataset contains 1,718 individuals and 1,854,622 SNPs. The original VCF-formatted data are converted to PLINK `bed` format. Samples are labeled as belonging to one of 26 populations, which are grouped into five superpopulations designated African, admixed American, East Asian, European, and South Asian. The HGDP dataset contains the individuals in the Stanford H952 dataset with greater than a 95% genotyping success rate and SNPs with at least a 1% MAF. The HGDP data contain 642,951 SNPs and 940 individuals across 32 populations, which are grouped into seven continental

Table 1. Accuracy of estimated admixture proportion on the simulated datasets

Number of samples	Number of SNPs	Number of populations	Percentage of SNPs selected as AIMs by SKFR								(baseline)	
			2.5%	5%	7.5%	10%	12.5%	15%	17.5%	20%	100%	SCOPE
1,000	10,000	5	0.0457	0.0365	0.0320	0.0292	0.0276	0.0263	0.0252	0.0244	0.0196*	0.0309
1,000	100,000	5	0.0154	0.0122	0.0110	0.0101	0.0098	0.0093	0.0092	0.0091	0.0089*	0.0104
1,000	1,000,000	5	0.0065	0.0063	0.0064	0.0064	0.0066	0.0065	0.0067	0.0067	0.0076	0.0056*
10,000	10,000	5	0.0482	0.0371	0.0332	0.0310	0.0296	0.0285	0.0277	0.0270	0.0231*	0.0333
10,000	100,000	5	0.0166	0.0135	0.0120	0.0111	0.0106	0.0103	0.0100	0.0098	0.0088*	0.0126
1,000	10,000	10	0.0539	0.0428	0.0369	0.0337	0.0315	0.0298	0.0287	0.0277	0.0226*	0.0307
1,000	100,000	10	0.0183	0.0148	0.0136	0.0131	0.0128*	0.0129	0.0130	0.0131	0.0176	0.0128*
1.000	1,000,000	10	0.0114	0.0113	0.0114	0.0117	0.0118	0.0120	0.0122	0.0124	0.0169	0.0098*
10,000	10,000	10	0.0529	0.0421	0.0373	0.0344	0.0324	0.0309	0.0298	0.0291	0.0234*	0.0320
10,000	100,000	10	0.0186	0.0148	0.0131	0.0121	0.0113	0.0108	0.0104	0.0101	0.0082*	0.0122

Accuracy is measured in terms of root-mean-square error. The best value in each row is denoted with an asterisk.

superpopulations: Europe, Middle East, Central South Asia, East Asia, Africa, America, and Oceania. The HO dataset is filtered to include only samples with at least a 99% genotype success rate and SNPs with at least a 5% MAF. The HO data contain 385,089 SNPs for 1,931 people across 163 populations. Continental population labels are not provided. For the UKB dataset,[14] we filtered bulk genotypes to include individuals with at least a 95% genotyping success rate and SNPs with at least a 1% MAF. The resulting data include 488,154 individuals and 610,741 SNPs.

Simulations

We simulated data following the Pritchard-Stephens-Donnelly (PSD) model[9] based on the software provided in the SCOPE package.[13] In the PSD model,

$$p_{kj} \sim \text{Beta}\left(\frac{1-F_{ST}}{F_{ST}}p_A, \frac{1-F_{ST}}{F_{ST}}(1-p_A)\right)$$
$$q_{\cdot i} \sim \text{Dirichlet}(\alpha \mathbf{1}_K).$$

The allele frequencies p_{kj} are sampled following the Balding-Nichols[47] model, a beta distribution characterized by the fixation index F_{ST} and the initial allele frequency p_A. We sample F_{ST} and p_A from their distributions in the TGP dataset, as illustrated in Figure S1. For admixture proportions q_{ki}, we sample a Dirichlet distribution with $\alpha = 0.2$, and for each genotype we sample from the binomial distribution

$$x_{ij} \sim \text{Binomial}\left(2, \sum_{k=1}^{K} p_{kj}q_{ki}\right).$$

Initial allele frequencies p_A outside the interval [0.005,0.995] are clipped to the closest endpoint. To simulate weak genetic structure, we also sampled F_{ST} uniformly from the range $(0, 0.01)$ rather than from the F_{ST} distribution found in the TGP data.

Results

Simulation studies

To determine a reasonable number of AIMs to choose using SKFR, we simulated independent datasets with various numbers of samples, numbers of SNPs, and numbers of populations. Table 1 records root-mean-square errors. Without filtering SNPs, OpenADMIXTURE shows better accuracy compared to SCOPE when the number of samples dominates the number of SNPs. SCOPE performs better in the reverse situation. In either case, selecting 15%–20% of the SNPs as AIMs via SKFR gives a good intermediate root-mean-square error between that of OpenADMIXTURE and SCOPE, where both use all the SNPs. When there are a million SNPs and 1,000 samples, similar to the TGP and HGDP datasets, using SKFR to select just 25,000 SNPs (2.5% of the SNPs) is enough for reasonable results. There is no evidence that selection by SKFR is biased in terms of either allele frequency or fixation index F_{ST}. Indeed, Figures S1 and S2 display no visible difference in either measure's distribution before and after SKFR selection.

We also examined a version of SKFR selecting a predefined number of AIMs per cluster, proposed as "SKFR2" in Zhang et al.[24] The results are displayed in Table S1, which is largely similar to Table 1. As the AIMs selected by each cluster may overlap, it is difficult to control the total number of AIMs selected under this strategy. To directly control the total number of AIMs, we use the version discussed in section "sparse K-means with feature ranking (SKFR)", selecting predefined number of AIMs across all the clusters ("SKFR1") for our analysis of real data. Our software supports both versions of SKFR.

Selection of K

Table 2 shows the value of K chosen under different settings. The gap method consistently chooses K close to the true value during data generation, even with a relatively small number of selected AIMs. However, when the number of AIMs is less than 0.5% of the total number of SNPs, the limited information available causes the gap statistic to underestimate K. Choosing at least 10,000 to 100,000 AIMs works well in general. Table 2 suggests that SKFR's deletion of uninformative SNPs tends to improve clustering. Table S2 presents the values of K selected under our weak structure simulations with reduced F_{ST}. Cluster number estimation

Table 2. Number of clusters inferred by the gap statistics in SKFR

Number of samples	Number of SNPs	Number of populations	Percentage of SNPs selected as AIMs by SKFR									(baseline)
			0.1%	0.2%	0.5%	1%	2%	5%	10%	20%	50%	100%
1,000	10,000	5	2	3	3	4	4	4	4	4	4	4
1,000	100,000	5	3	4	4	4	4	4	4	4	4	4
1,000	1,000,000	5	4	4	4	4	4	4	4	4	4	4
10,000	10,000	5	4	3	3	4	4	4	4	5	5	4
10,000	100,000	5	3	4	4	4	5	5	5	5	5	5
1,000	10,000	10	3	4	6	8	8	10	10	9	8	8
1,000	100,000	10	10	10	9	9	9	9	9	9	8	8
1,000	1,000,000	10	8	8	10	10	10	9	9	9	9	7
10,000	10,000	10	2	4	6	8	8	9	10	10	10	10
10,000	100,000	10	8	8	10	10	10	10	10	10	10	13

Allele frequencies and fixation indexes, F_{ST}, are sampled from the values of SNPs in the TGP data. Admixture proportions are sampled from a Dirichlet distribution with $\alpha = 0.02$. See the text for simulation details.

is unstable, with estimated values of K usually too low but occasionally too high. Again, the gap statistic is stymied by the limited information available.

Large-scale real datasets

Hard clustering via SKFR

To evaluate the clustering performance of SKFR, we ran it on the TGP data with $K = 8$ clusters, the HGDP data with $K = 10$ clusters, and the HO data with $K = 14$ clusters, choices consistent with previous analyses of these data.[11–13] Recall that in the TGP data, each individual is labeled as coming from one of 26 populations and one of five superpopulations. We tried ten different initializations for sparsity level $S = 100,000$ and chose the best clustering according to the loss function of Equation 2. Then we successively decremented S to 80,000, 60,000, 40,000, 20,000, 10,000, and 5,000 by using the warm start tactic described in section "warm start for a path with different sparsity level S". We computed the adjusted Rand index (ARI)[48,49] and the normalized mutual information (NMI)[49] between our hard clusterings and the five superpopulation labels originally attributed to the sample individuals. Although these two metrics are rather opaque, they do allow the number of clusters to differ in each clustering. These measures were also computed for the baseline K-means clustering with all SNPs included. The baseline results also reflect ten different initializations.

Table 3 for the TGP data shows that SKFR's hard clusterings clearly outperform the baseline K-means results and that the SKFR results are stable across a wide range of selected SNPs. When we exclude the admixed American (AMR) superpopulation in our assessment, our clusters perfectly capture the four remaining superpopulations. It appears that including uninformative SNPs or admixed samples creates unwanted noise that obscures true clusters. For the HGDP and HO data (Tables S3 and S4, respectively), the ARI and NMI measures delivered by SKFR are comparable to but slightly worse than those of the baseline K-means. For the HGDP data this anomalous result may stem from the admixed nature of the HGDP superpopulations. It is also noteworthy that the HO data include 163 different population labels. For the HGDP and HO data, ARI and NMI decrease as we choose more AIMs, up to a total of 100,000 selected AIMs. This anomaly may have two sources. First, we are relying on possibly inaccurate self-reported population labels. Second, we are hard labeling individuals who may be admixed. Unlike the TGP data, where it is straightforward to distinguish admixed populations (one continental label is literally "admixed Americans"), it is much more difficult to isolate less admixed populations from continental labels in the HGDP data, as it intentionally collected samples with more diverse background. In the case of HO, no continental labels are provided to compare to the 163 population labels.

Admixture estimation

We recorded concordance with ancestry labels included in the datasets as a performance measure for soft clustering under OpenADMIXTURE. We also trained a softmax (multinomial logistic) classifier to predict superpopulation labels using TGP data with the inferred admixture proportions as predictors. Since the results are continuous proportions rather than hard clusters, cross-entropy is a reasonable measure of error. We additionally matched clusters as discussed in section "permutation matching of clusters" and computed root-mean-square error (RMSE) compared to the OpenADMIXTURE estimates with all SNPs included.

Tables 4, S5, and S6 display our complex findings for the TGP, HGDP, and HO datasets, respectively. The accuracy of OpenADMIXTURE classification with a limited number of AIMs is roughly comparable to that of SCOPE, which employs all SNPs. In general, cross-entropy decreases (improves) as we select more AIMs in OpenADMIXTURE's inference. In particular for the HO and TGP datasets, the

Table 3. Hard clustering performance on the TGP data with all samples

	All samples		Without AMR	
Number of AIMs	ARI	NMI	ARI	NMI
5,000	0.824	0.839	1.000	1.000
10,000	0.825*	0.840	1.000	1.000
20,000	0.825*	0.840	1.000	1.000
40,000	0.825*	0.841*	1.000	1.000
60,000	0.824	0.840	1.000	1.000
80,000	0.825*	0.840	1.000	1.000
100,000	0.822	0.837	1.000	1.000
All SNPs	0.575	0.726	0.726	0.802

Performance for "all samples" measured relative to the five superpopulation labels and also for the samples not including the admixed Americans ("without AMR") relative to the remaining four superpopulation labels. Performance is evaluated with the adjusted Rand index (ARI) and the normalized mutual information (NMI). The best value in each column is denoted with an asterisk, except when the maximum value of 1.0 is reached. The category "all SNPs" refers to baseline results under *K*-means.

OpenADMIXTURE estimates with 60,000 or more AIMs outperform SCOPE. The RMSEs of SCOPE and AIM-driven OpenADMIXTURE are also roughly comparable within each of the three datasets. SCOPE does somewhat better on the HO data, while OpenADMIXTURE does better on the HGDP and TGP data. Again, we stress that the admixed nature of the data may obscure the value of limiting analysis to AIMs and cloud the choice of the optimal number of AIMs.

The TGP data demonstrate the value of excluding samples known to be admixed. Table 4 shows that OpenADMIXTURE classification is perfect for the non-AMR individuals with at least 20,000 AIMs selected. The table also shows better cross-entropy for classifying non-admixed samples versus all samples. Table S7 reinforces these findings by omitting AMR samples during SKFR AIM selection prior to admixture analysis. Table S7 shows slightly better classification performance than that recorded in Table 4 with the same number of AIMs.

For the UKB data with $K = 4$ and $K = 15$ clusters, we computed the accuracy of the softmax classifier with three sets of labels. The first set (L1) uses all 22 raw labels. The second (L2) uses the eight labels, British, Irish, Indian, Pakistani, Bangladeshi, Caribbean, African, and Chinese, for roughly homogeneous populations and removes mixed or uncertain labels such as "mixed" or "other." The third set (L3) merges L2's populations into four continental groupings, British-Irish, Indian-Pakistani-Bangladeshi, Caribbean-African, and Chinese. Table S8 reports this classification accuracy for OpenADMIXTURE with $S = 100,000$ AIMs for $K = 4$ clusters and with $S = 80,000$ AIMs for $K = 15$ clusters and for SCOPE with all SNPs included. SCOPE failed to run with $K = 15$ clusters, giving not-a-number (NaN) internal errors. Note that our preprocessing is simpler than that of Chiu et al.[13]

We checked whether OpenADMIXTURE can capture regional structure in the historically British subset of the UKB data used to compute PCs.[14] The subset consists of 147,604 typed SNPs on 430,815 subjects who self-identify their ethnicity as British. As dictated by the gap statistic, we set the number of populations to $K = 9$ and trained a softmax classifier to predict the assessment region of each subject. The 22 assessment centers across UK can be grouped into five regions: North England, South England, North Wales, South Wales, and Scotland. There is no center in North Ireland. The training accuracy with OpenADMIXTURE is 67.7%. SCOPE with $K = 9$ exhibits a training accuracy of 64.9%. We also trained the softmax classifier with eight PCs to match the number of free parameters under clustering.[50] The training accuracy with principal-component analysis (PCA) is 67.5%, very similar to OpenADMIXTURE's 67.7%. Our results are displayed in Table S9. This type of analysis is limited by the imperfect relationship between assessment centers visited and ancestry.

Visualization

Figures 1, 2, and 3 depict the inferred admixture proportions for the TGP, HGDP, and HO datasets. Each figure includes three graphs: first, the results from OpenADMIXTURE with all SNPs; second, the results from OpenADMIXTURE with 100,000 AIMs; and third, the results from SCOPE.

Computation times and maximum memory requirements

Most of our numerical experiments were run on Amazon Web Services (AWS). Table S10 lists the hardware instances invoked for computation. For our GPU experiments, we used two types of GPUs. The first, Nvidia A10G in a g5.4xlarge instance, is a moderate-grade GPU designed for low-cost performance. The second, Nvidia V100 in a p3.2xlarge instance, is specialized for scientific computing. The main difference between the two is double precision performance. By design, double precision performance on Nvidia A10G is 32 times slower than single precision, while double precision on Nvidia V100 is only twice as slow as single precision.

Table 4. Performance comparison of OpenADMIXTURE and SCOPE on the TGP dataset

	All samples		Without AMR		
Number of AIMs	Accuracy	Cross-entropy	Accuracy	Cross-entropy	RMSE from Baseline
5,000	0.941	313	1.000	37.4	0.234
10,000	0.947	295	0.998	34.9	0.185
20,000	0.947	284	1.000	33.4	0.152
40,000	0.953	274	1.000	31.7	0.164
60,000	0.971	242	1.000	29.0	0.043
80,000	0.968	241	1.000	30.5	0.033
100,000	0.969	241	1.000	30.1	0.027*
All SNPs	0.980*	232*	1.000	27.3*	–
SCOPE	0.979	248	1.000	32.1	0.044

Performance is measured by training accuracy and cross-entropy with the five (four without admixed Americans ["without AMR"]) continental labels delivered by the trained softmax classifier. Root-mean-square error (RMSE) from baseline compares estimated admixture coefficients to those with all SNPs included and regular ADMIXTURE run without prior SKFR AIM selection ("all SNPs"). The best value in each column is denoted with an asterisk; accuracy has a maximum value of 1.0.

Comparison to SCOPE

It is instructive to compare the runtimes of OpenADMIXTURE (pipelining SKFR followed by admixture estimation) to those of SCOPE. For our pipeline on the TGP data, when all 16 available threads are used in a g5.4xlarge instance, a single SKFR run takes 1 min 36 s. Filtering takes less than a minute. The subsequent run of admixture estimation takes less than 5 min with 100,000 or fewer SNPs. Cumulatively, the pipeline takes fewer than 7 min. On the other hand, a SCOPE run on the TGP data takes slightly over 16 min on the same hardware.

For the UKB data with $K = 4$, a single run of SKFR takes 44 min on a compute-optimized c6i.16xlarge instance with 128 GB memory. The maximum memory footprint is 73.2 GB. Creating the PLINK files containing only the selected AIMs takes less than 10 min. The subsequent admixture estimation run with 100,000 selected SNPs takes 29 min on a V100 GPU. Thus, the total pipeline runtime, invoked by a single OpenADMIXTURE call, was under 83 min. SCOPE took a similar 91 min to run on the UKB dataset. Since SCOPE's memory requirement for this dataset is 250 GB, it had to be run on a more expensive memory-optimized r6i.16xlarge instance with 512 GB memory. In summary, total computation times are comparable, but OpenADMIXTURE is clearly less memory intensive than SCOPE.

Runtime improvement versus the original ADMIXTURE software

Although it invokes the same statistical model and optimization strategy, OpenADMIXTURE delivers better performance than the original ADMIXTURE software. Table S11 records the per-iteration times of various admixture estimation routines on the TGP data with 100,000 AIMs. 16-thread CPU and A10G GPU experiments were performed on an AWS g5.4xlarge hardware instance; V100 GPU experiments were performed on an AWS p3.2xlarge hardware instance. OpenADMIXTURE software, when restricted to CPUs, is 2.8 times faster on a single thread, and 8 times faster in a 16-thread run, compared to the original ADMIXTURE. When a GPU is available, OpenADMIXTURE accelerates computation by another factor of 2–4, depending of course on the GPU hardware and the floating-point precision invoked.

Discussion

This paper presents a biobank-scalable, unsupervised pipeline for AIM selection and admixture estimation. Our procedures provide both interpretable admixture coefficients and population-specific allele frequencies. Our Julia package OpenADMIXTURE implements the entire pipeline. The SKFR (sparse *K*-means with feature ranking) component of the pipeline is highly parallelized and effective in AIM selection. SKFR's unsupervised clustering is insensitive to a small fraction of labeling errors and admixed samples. SKFR also delivers an explicit ranking of AIMs. Our experiments suggest that 10,000–100,000 AIMs deliver better clusters than full biobank-scale SNP sets. Uninformative SNPs simply constitute noise that slows clustering and obscures subpopulations.

The second component of the pipeline, estimation of admixture proportions, is an open-source re-implementation in the Julia programming language of our previous package ADMIXTURE. The original ADMIXTURE[7] is widely used, with over 5,400 Google citations. OpenADMIXTURE is up to 8 times faster than ADMIXTURE on CPUs with multithreading and even faster on computers with GPUs. Total computation time is comparable to SCOPE, another method currently scalable to biobank data. We have shown that both OpenADMIXTURE and SCOPE can analyze a dataset with 500,000 individuals and 600,000 SNPs in well under

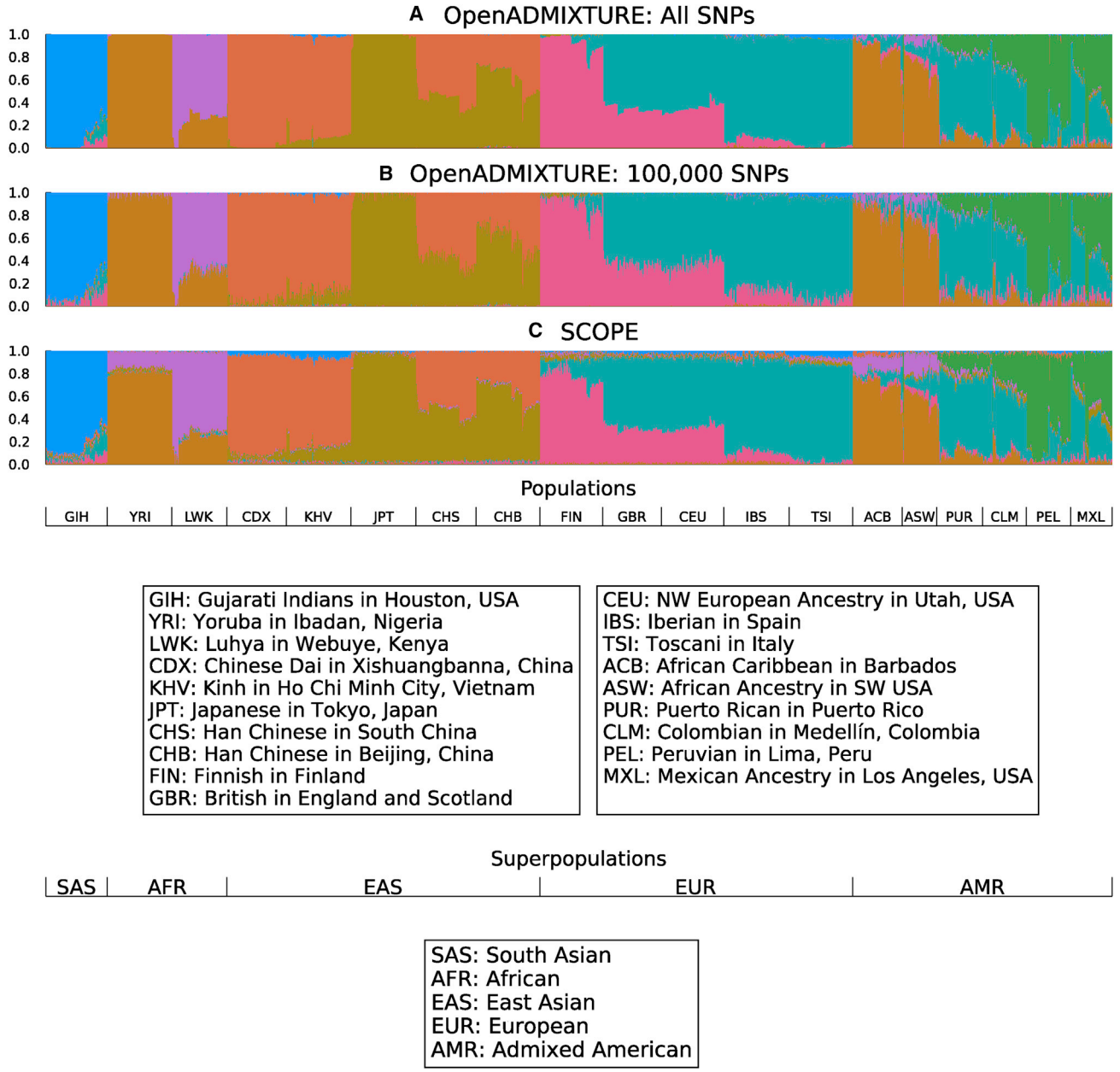

Figure 1. Estimated ancestry of TGP data samples
(A–C) Using OpenADMIXTURE with all SNPs, using OpenADMIXTURE with 100,000 AIMs (B), and using SCOPE (C). These are stacked bar plots with the y axis indicating the proportion of total ancestry. The x axis runs over all samples; the population labels originally assigned to these samples within the dataset are provided in the lower sections of the figure.

2 h. The paper[13] introducing SCOPE took about 24 h to analyze the same data. However, the current version of SCOPE is more efficient than the original version, and computers are more powerful.

The memory demands of OpenADMIXTURE are exceptionally light as a result of its systematic exploitation of PLINK's binary format for both computation and genotype storage. OpenADMIXTURE's peak memory footprint is less than 120% of the size of the genotype input file. Overall, OpenADMIXTURE's memory footprint is less than 30% of that of SCOPE. Specifically, to analyze the above biobank dataset, SCOPE requires 250 GB of RAM, while OpenADMIXTURE needs under 75 GB. OpenADMIXTURE is also based on a likelihood model that incorporates basic population genetics concepts.

The computational complexity $O(IJK^2)$ of Hessian computation is a bottleneck for OpenADMIXTURE in dealing with $K > 20$ populations. Limiting analysis to a small number of AIMs reduces runtimes but does not eliminate the K^2 dependence. If it is found desirable to tackle problems with large K, then gradient ascent might be helpful. Unfortunately, gradient ascent subject to constraints tends to be slow unless one can determine a nearly optimal step size. Line searches along the gradient direction require repeated log likelihood

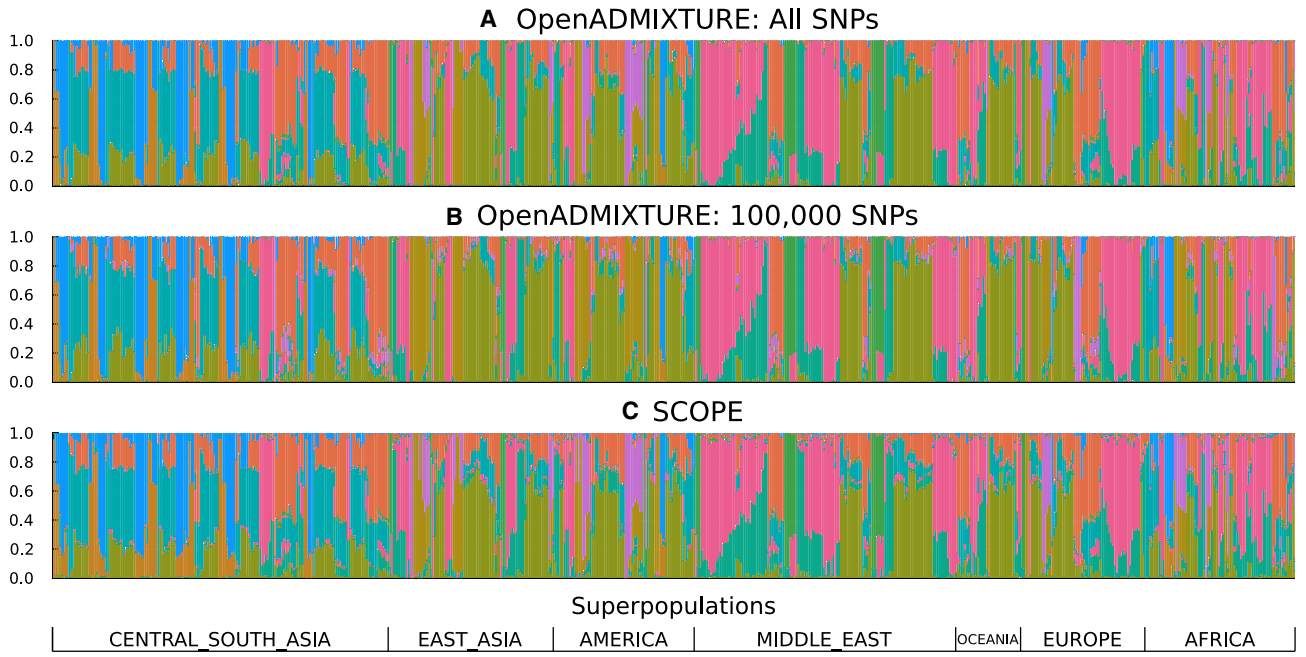

Figure 2. Estimated ancestry of HGDP data samples
(A–C) Using OpenADMIXTURE with all SNPs (A), using OpenADMIXTURE with 100,000 AIMs (B), and using SCOPE (C). These are stacked bar plots, with the y axis indicating the proportion of total ancestry. The x axis runs over all samples; the population labels originally assigned to these samples within the dataset are provided in the lower section of the figure.

evaluations and are expensive. We defer resolution of this issue to future research.

We have ignored the possible biological insights offered by the AIMs selected by SKFR. The genomic locations of AIMs and their relation to the ancestral populations of the samples warrant further research. The version of the SKFR algorithm that selects different AIM sets for different clusters may potentially improve biological interpretability. This issue also warrants further research. Selecting the number of clusters K and the sparsity level S is a third issue. Methods based on cross-validation require repeated runs of the pipeline and may be impractical on biobank-scale data. Because OpenADMIXTURE relies on a likelihood model, determination of K is possible on the basis of the Akaike information criterion

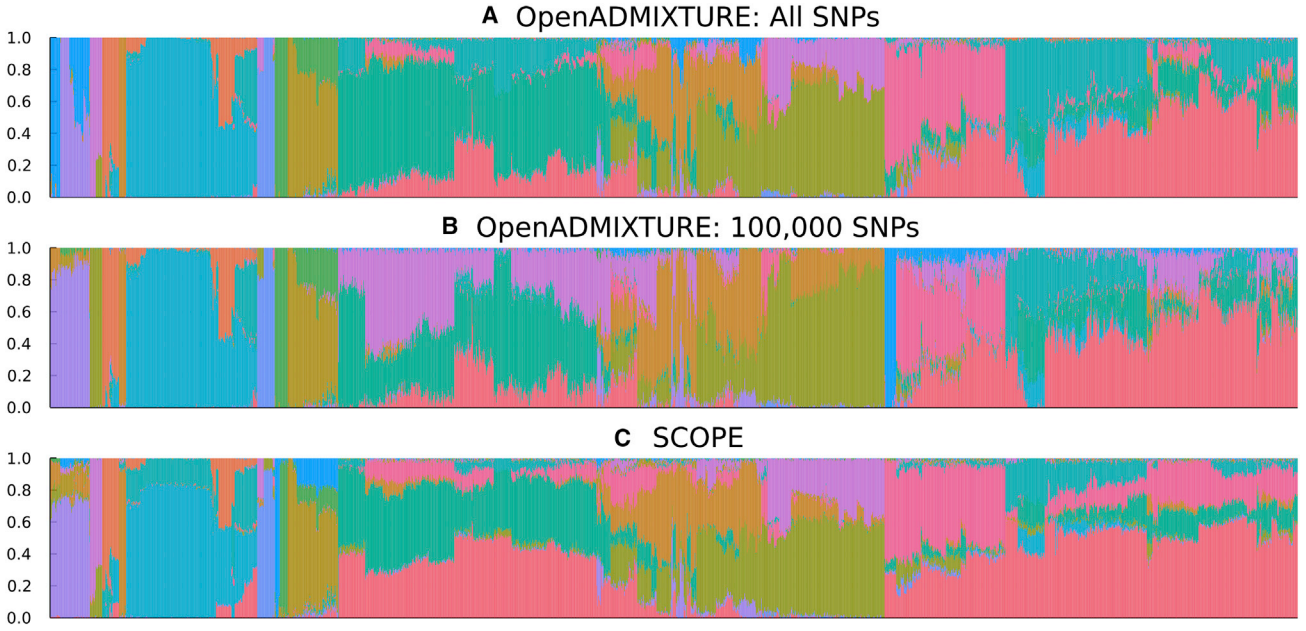

Figure 3. Estimated ancestry of HO data samples
(A–C) Using OpenADMIXTURE with all SNPs (A), using OpenADMIXTURE with 100,000 AIMs (B), and using SCOPE (C). These are stacked bar plots, with the y axis indicating the proportion of total ancestry. The x axis runs over all samples.

(AIC) or the Bayesian information criterion (BIC). The current paper relies on the standard gap statistic for choosing K.[35] Alternatively, one can run SKFR with a variety of K values and then check AIC or BIC with the selected clusters and AIMs with OpenADMIXTURE. For selecting the sparsity level S, a variant of gap statistics may be helpful as well.[24] If genetic structure is weak, K may be overestimated, particularly when too few AIMs are chosen. We recommend a minimum of 10,000 AIMs. Again, the optimal method for choosing K is a question for future research.

OpenADMIXTURE offers the option of inferring admixture proportions on the basis of the population allele frequencies available in reference panels such as TGP. This approach fixes the allele frequency matrix $\boldsymbol{P}$ and only updates the admixture coefficient matrix $\boldsymbol{Q}$. Sequential quadratic programming easily solves this simplified convex problem, which is parameter separated across samples. Thus, OpenADMIXTURE can be readily applied to sample collections ranging from small to biobank scale.

In summary, OpenADMIXTURE is a substantial upgrade of ADMIXTURE. Although the full panoply of options already available in ADMIXTURE has not yet been implemented, the ADMIXTURE community will surely welcome an open-source version that can be cooperatively developed further. The OpenMendel tools that OpenADMIXTURE already exploits provide a clear path to further improvement. We also expect Julia's parallelization ecosystem to expand over time. We solicit the suggestions and assistance of committed users in the ADMIXTURE community in our efforts.

Appendix A

Algorithm 1A. SKFR Algorithm

Input: Standardized genotype matrix $\boldsymbol{X} \in \boldsymbol{R}^{I \times J}$, number of clusters K, sparsity level S, and initial clusters C_k.
repeat
  for all cluster k: **do**
    $\boldsymbol{\theta}_k = \frac{1}{|C_k|} \sum_{i \in C_k} \boldsymbol{x}_i$
  end for
  for all feature j: **do**
    Rank j by criterion $h_j = \sum_k |C_k| \theta_{kj}^2$
  end for
  Let L be the set of S features with the highest h_j
  for all sample i: **do**
    Assign sample i to the cluster C_k that minimizes $\sum_{j \in L} (x_{ij} - \theta_{kj})^2 + \sum_{j \notin L} x_{ij}^2$
  end for
  for all feature $j \notin L$: **do**
    Put j-th column of Θ to zero
  end for
until convergence
return Cluster assignments $\boldsymbol{B}$ ($b_{ik} = 1_{i \in C_k}$), Θ, and ranked list L.

Algorithm 1B. SKFR Algorithm Incorporating Missing Genotypes

Input: Standardized genotype matrix $\boldsymbol{X} \in \mathbb{R}^{I \times J}$, number of clusters K, sparsity level S, initial clusters C_k, and iteration number n.
Initialize $\Theta_0 = m \boldsymbol{1}_K \boldsymbol{1}_J^T$, where m is the mean of non-missing entries of $\boldsymbol{X}$
repeat
  $n = n + 1$
  $\boldsymbol{Y}_n = P_{\mathcal{Q}}(\boldsymbol{X}) + P_{\mathcal{Q}^\perp}(\boldsymbol{B}_{n-1} \Theta_{n-1})$
  Run the standard SKFR algorithm on $\boldsymbol{Y}_n$ to obtain $\boldsymbol{B}_n$ and Θ_n, and the ranked list of AIMs L
until convergence
return $\boldsymbol{B}_n$, Θ_n, and L.

Data and code availability

The stand-alone SKFR package can be found at https://github.com/kose-y/SKFR.jl. The OpenADMIXTURE package can be found at https://github.com/OpenMendel/OpenADMIXTURE.jl. The code for the experiments, and instructions to download publicly available data, can be found at https://github.com/kose-y/OpenADMIXTURE-resources. One exception is the UK Biobank data, which are available via application at https://www.ukbiobank.ac.uk. The UK Biobank data were retrieved under Project ID: 48152.

Supplemental information

Supplemental information can be found online at https://doi.org/10.1016/j.ajhg.2022.12.008.

Acknowledgments

This research was partially funded by grants from the National Research Foundation of Korea (NRF) (Basic Science Research Program, 2020R1A6A3A03037675, S.K.), the National Institute of General Medical Sciences (R35GM141798, E.M.S., H.Z., and K.L.L.), the National Human Genome Research Institute (R01HG006139, J.C.P., E.M.S., H.Z., and K.L.L.), and the National Science Foundation (DMS-2054253 and IIS-2205441, H.Z.).

Declaration of interests

The authors declare no competing interests.

Received: July 12, 2022
Accepted: December 12, 2022
Published: January 6, 2023

Web resources

SCOPE, https://github.com/sriramlab/SCOPE
SnpArrays, https://github.com/OpenMendel/SnpArrays.jl

References

1. Novembre, J., Johnson, T., Bryc, K., Kutalik, Z., Boyko, A.R., Auton, A., Indap, A., King, K.S., Bergmann, S., Nelson, M.R.,

et al. (2008). Genes mirror geography within Europe. Nature *456*, 98–101.
2. Price, A.L., Patterson, N.J., Plenge, R.M., Weinblatt, M.E., Shadick, N.A., and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. Nat. Genet. *38*, 904–909.
3. Li, C.C. (1969). Population subdivision with respect to multiple alleles. Ann. Hum. Genet. *33*, 23–29.
4. Knowler, W.C., Williams, R.C., Pettitt, D.J., and Steinberg, A.G. (1988). Gm3; 5, 13, 14 and type 2 diabetes mellitus: an association in American Indians with genetic admixture. Am. J. Hum. Genet. *43*, 520–526.
5. Marchini, J., Cardon, L.R., Phillips, M.S., and Donnelly, P. (2004). The effects of human population structure on large genetic association studies. Nat. Genet. *36*, 512–517.
6. Lawson, D.J., Davies, N.M., Haworth, S., Ashraf, B., Howe, L., Crawford, A., Hemani, G., Davey Smith, G., and Timpson, N.J. (2020). Is population structure in the genetic biobank era irrelevant, a challenge, or an opportunity? Hum. Genet. *139*, 23–41.
7. Alexander, D.H., Novembre, J., and Lange, K. (2009). Fast model-based estimation of ancestry in unrelated individuals. Genome Res. *19*, 1655–1664.
8. Zhou, H., Alexander, D., and Lange, K. (2011). A quasi-Newton acceleration for high-dimensional optimization algorithms. Stat. Comput. *21*, 261–273.
9. Pritchard, J.K., Stephens, M., and Donnelly, P. (2000). Inference of population structure using multilocus genotype data. Genetics *155*, 945–959.
10. Raj, A., Stephens, M., and Pritchard, J.K. (2014). fastSTRUCTURE: variational inference of population structure in large SNP data sets. Genetics *197*, 573–589.
11. Gopalan, P., Hao, W., Blei, D.M., and Storey, J.D. (2016). Scaling probabilistic models of genetic variation to millions of humans. Nat. Genet. *48*, 1587–1590.
12. Cabreros, I., and Storey, J.D. (2019). A likelihood-free estimator of population structure bridging admixture models and principal components analysis. Genetics *212*, 1009–1029.
13. Chiu, A.M., Molloy, E.K., Tan, Z., Talwalkar, A., and Sankararaman, S. (2022). Inferring population structure in biobank-scale genomic data. Am. J. Hum. Genet. *109*, 727–737.
14. Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L.T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., et al. (2018). The UK Biobank resource with deep phenotyping and genomic data. Nature *562*, 203–209.
15. Shriver, M.D., Parra, E.J., Dios, S., Bonilla, C., Norton, H., Jovel, C., Pfaff, C., Jones, C., Massac, A., Cameron, N., et al. (2003). Skin pigmentation, biogeographical ancestry and admixture mapping. Hum. Genet. *112*, 387–399.
16. Brown, R., and Pasaniuc, B. (2014). Enhanced methods for local ancestry assignment in sequenced admixed individuals. PLoS Comput. Biol. *10*, e1003555.
17. Sinsheimer, J.S., Plaisier, C.L., Huertas-Vazquez, A., Aguilar-Salinas, C., Tusie-Luna, T., Pajukanta, P., and Lange, K. (2008). Estimating ethnic admixture from pedigree data. Am. J. Hum. Genet. *82*, 748–755.
18. Li, C.-X., Pakstis, A.J., Jiang, L., Wei, Y.-L., Sun, Q.-F., Wu, H., Bulbul, O., Wang, P., Kang, L.-L., Kidd, J.R., and Kidd, K.K. (2016). A panel of 74 AISNPs: improved ancestry inference within eastern Asia. Forensic Sci. Int. Genet. *23*, 101–110.
19. Zeng, X., Chakraborty, R., King, J.L., LaRue, B., Moura-Neto, R.S., and Budowle, B. (2016). Selection of highly informative SNP markers for population affiliation of major US populations. Int. J. Legal Med. *130*, 341–352.
20. Pfaffelhuber, P., Grundner-Culemann, F., Lipphardt, V., and Baumdicker, F. (2020). How to choose sets of ancestry informative markers: A supervised feature selection approach. Forensic Sci. Int. Genet. *46*, 102259.
21. Santos, H.C., Horimoto, A.V.R., Tarazona-Santos, E., Rodrigues-Soares, F., Barreto, M.L., Horta, B.L., Lima-Costa, M.F., Gouveia, M.H., Machado, M., Silva, T.M., et al. (2016). A minimum set of ancestry informative markers for determining admixture proportions in a mixed American population: the Brazilian set. Eur. J. Hum. Genet. *24*, 725–731.
22. Ding, L., Wiener, H., Abebe, T., Altaye, M., Go, R.C.P., Kercsmar, C., Grabowski, G., Martin, L.J., Khurana Hershey, G.K., Chakorborty, R., and Baye, T.M. (2011). Comparison of measures of marker informativeness for ancestry and admixture mapping. BMC Genom. *12*, 622–718.
23. Lee, S., Epstein, M.P., Duncan, R., and Lin, X. (2012). Sparse principal component analysis for identifying ancestry-informative markers in genome-wide association studies. Genet. Epidemiol. *36*, 293–302.
24. Zhang, Z., Lange, K., and Xu, J. (2020). Simple and scalable sparse k-means clustering via feature ranking. In Adv. Neural Inf. Process. Syst., H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds. (Curran Associates, Inc), pp. 10148–10160.
25. Bezanson, J., Edelman, A., Karpinski, S., and Shah, V.B. (2017). Julia: A fresh approach to numerical computing. SIAM Rev. *59*, 65–98.
26. Arthur, D., and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms pages 1027–1035 (Society for Industrial and Applied Mathematics).
27. Chu, B.B., Sobel, E.M., Wasiolek, R., Ko, S., Sinsheimer, J.S., Zhou, H., and Lange, K. (2021). A fast data-driven method for genotype imputation, phasing and local ancestry inference: MendelImpute. Bioinformatics *37*, 4756–4763.
28. Das, S., Forer, L., Schönherr, S., Sidore, C., Locke, A.E., Kwong, A., Vrieze, S.I., Chew, E.Y., Levy, S., McGue, M., et al. (2016). Next-generation genotype imputation service and methods. Nat. Genet. *48*, 1284–1287.
29. Chi, J.T., Chi, E.C., and Baraniuk, R.G. (2016). k-pod: A method for k-means clustering of missing data. Am. Statistician *70*, 91–99.
30. Lange, K., Hunter, D.R., and Yang, I. (2000). Optimization transfer using surrogate objective functions. J. Comput. Graph Stat. *9*, 1–20.
31. Hunter, D.R., and Lange, K. (2004). A tutorial on MM algorithms. Am. Statistician *58*, 30–37.
32. Lange, K. (2016). MM Optimization Algorithms (SIAM).
33. Gallant, A.R., and Gerig, T.M. (1978). Proceedings of the Computer Science and Statistics: Eleventh Annual Symposium on the INTERFACE, held at North Carolina State University, March 6 and 7, 1978 (Technical report North Carolina State University. Dept. of Statistics).
34. Tang, H., Peng, J., Wang, P., and Risch, N.J. (2005). Estimation of individual admixture: analytical and study design considerations. Genet. Epidemiol. *28*, 289–301.
35. Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. J. Roy. Stat. Soc. B *63*, 411–423.

36. Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M.A.R., Bender, D., Maller, J., Sklar, P., de Bakker, P.I.W., Daly, M.J., and Sham, P.C. (2007). PLINK: a tool set for whole-genome association and population-based linkage analyses. Am. J. Hum. Genet. *81*, 559–575.
37. Zhou, H., Sinsheimer, J.S., Bates, D.M., Chu, B.B., German, C.A., Ji, S.S., Keys, K.L., Kim, J., Ko, S., Mosher, G.D., et al. (2020). OpenMendel: a cooperative programming project for statistical genetics. Hum. Genet. *139*, 61–71.
38. Frigo, M., Leiserson, C.E., Prokop, H., and Ramachandran, S. (1999). Cache-oblivious algorithms. In In 40th Annual Symposium on Foundations of Computer Science (IEEE), pp. 285–297. Cat. No. 99CB37039.
39. Besard, T., Foket, C., and De Sutter, B. (2019). Effective extensible programming: unleashing Julia on GPUs. IEEE Trans. Parallel Distrib. Syst. *30*, 827–841.
40. Behr, A.A., Liu, K.Z., Liu-Fang, G., Nakka, P., and Ramachandran, S. (2016). Pong: fast analysis and visualization of latent clusters in population genetic data. Bioinformatics *32*, 2817–2823.
41. Dunning, I., Huchette, J., and Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. SIAM Rev. *59*, 295–320.
42. 1000 Genomes Project Consortium, Abecasis, G.R., Auton, A., Brooks, L.D., DePristo, M.A., Durbin, R.M., Handsaker, R.E., Kang, H.M., Marth, G.T., and McVean, G.A. (2012). An integrated map of genetic variation from 1, 092 human genomes. Nature *491*, 56–65.
43. Genomes Project Consortium and others (2015). A global reference for human genetic variation. Nature *526*, 68.
44. Cann, H.M., De Toma, C., Cazes, L., Legrand, M.-F., Morel, V., Piouffre, L., Bodmer, J., Bodmer, W.F., Bonne-Tamir, B., Cambon-Thomsen, A., et al. (2002). A human genome diversity cell line panel. Science *296*, 261–262.
45. Cavalli-Sforza, L.L. (2005). The human genome diversity project: past, present and future. Nat. Rev. Genet. *6*, 333–340.
46. Lazaridis, I., Patterson, N., Mittnik, A., Renaud, G., Mallick, S., Kirsanow, K., Sudmant, P.H., Schraiber, J.G., Castellano, S., Lipson, M., et al. (2014). Ancient human genomes suggest three ancestral populations for present-day Europeans. Nature *513*, 409–413.
47. Balding, D.J., and Nichols, R.A. (1995). A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. Genetica *96*, 3–12.
48. Rand, W.M. (1971). Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. *66*, 846–850.
49. Vinh, N.X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn. Res. *11*, 2837–2854.
50. Abraham, G., Qiu, Y., and Inouye, M. (2017). FlashPCA2: principal component analysis of biobank-scale genotype datasets. Bioinformatics *33*, 2776–2778.