UNIVERSITY OF CALIFORNIA

Los Angeles

Methods for detecting structure
in large-scale genomic data

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Bioinformatics

by

Alec Matthew Chiu

2022

ABSTRACT OF THE DISSERTATION

Methods for detecting structure
in large-scale genomic data

by

Alec Matthew Chiu
Doctor of Philosophy in Bioinformatics
University of California, Los Angeles, 2022
Professor Sriram Sankararaman, Chair

Large-scale repositories of genomic data are providing opportunities for researchers to answer biological questions at unprecedented resolution. Uncovering the structure underlying these datasets is a fundamental task where the structure can correspond to biological signals of interest or to confounders such as ancestry and batch effects that must be accounted for to prevent spurious findings. While discovering structure is a challenging problem, the growing size of genomic datasets leads to computational bottlenecks that further complicate their analysis. Here, we propose three scalable approaches for detecting structure in genomic data. We present ProPCA, a probabilistic principal component analysis method for large-scale genomic data. We also introduce SCOPE, a method for inferring admixture proportions from biobank-scale data. Both these methods utilize randomized eigendecomposition and the unique structure of the genotype matrix to perform scalable population structure inference. We apply these methods to simulations to reveal that they remain accurate while improving on runtime compared to existing methods. We applied both methods on the UK Biobank, a dataset containing half a million individuals, to uncover fine-scale structure within the United Kingdom. We subsequently introduce a statistical testing framework for detecting variance and covariance differences by extending eigengene analysis through a set of transformations and randomized eigendecomposition. We use RNA-seq data from individuals with psychiatric disease to reveal several (co)variance differences; highlighting the need to look beyond mean effects. With the increasing availability of large biological datasets, our work enables researchers to efficiently discover and test for structure and perform downstream analyses.

The dissertation of Alec Matthew Chiu is approved.

Bogdan Pasaniuc

Eran Halperin

Jingyi Li

Sriram Sankararaman, Committee Chair

University of California, Los Angeles

2022

To my family,

who have always been an endless supply of love and support

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

There are a countless number of people who have helped and guided me on this journey. I am grateful to everyone that I have encountered along the way.

I would like to acknowledge the faculty and staff who have shaped my academic career. Hilary Coller kickstarted my journey by providing me my first opportunity to conduct research in bioinformatics despite my limited knowledge of computational skills. Mithun Mitra supervised my first project in bioinformatics and helped me learn a lot about cluster analysis and bioinformatics. Together, the three of us published my first, first-author publication. I would not be pursuing a PhD if it were not for Eleazar Eskin. Eleazar was always a listening ear and provided me with opportunities that made me love bioinformatics more through his classes and programs such as Bruins-In-Genomics and the Computational Genomics Summer Institute. My committee, Bogdan Pasaniuc, Eran Halperin, and Jessica Li, have also been with me since the earliest days of graduate school. Bogdan and Eran have always been heavily involved in collaborations with our lab and have felt like secondary faculty advisors to me. Jessica is my favorite instructor at UCLA (along with Sriram!) and has taught me practically everything I know about statistics; knowledge that I will forever treasure through the future jobs and projects that I tackle. I would also like to thank Noah Zaitlen for also acting as a secondary mentor to me, especially through the COVID-19 pandemic. It was an honor to work with someone so passionate about the field. Lastly, I would like to thank my faculty advisor, Sriram Sankararaman. Sriram has taught me everything I know about machine learning and computing. I remember struggling through his class during my first quarter of graduate school, but learning so much. I am glad that he accepted me into the lab despite my limited computational knowledge. Since joining the lab, I have picked up so much knowledge about statistics, machine learning, and computing. When I first entered graduate school, I barely knew what PCA was, but now I have developed multiple methods that utilize the technique. I am immensely grateful for the support and mentorship given to me the past five years.

I would like to thank my colleagues in the Sankararaman lab, Halperin lab, Bogdan lab, and Zaitlen lab. In particular, I would like to thank Chris Robles, Ruthie Johnson, Ariel Wu, Boyang Fu, Albert Xue, Arun Durvasula, and Erin Molloy from the Sankararaman lab. Chris and Arun were like big brothers to me in the lab and played a large role in why I fell in love with the Sankararaman lab. Erin supported me tremendously on my projects and as a friend throughout the COVID-19 pandemic.

I would like to thank the friends and colleagues that I have met along the way. My cohort (Leah Briscoe, Jesse Garcia, Brandon Jew, Juan de la Hoz, Sarah Spendlove, Mike Thompson, Tommer Schwarz, Ha Vu, Kofi Amoah, Christa Caggiano, Ruthie Johnson, Kodi Collins) has been an anchor point for me throughout the years and I am extremely glad that we were able to maintain our connections throughout our time in graduate school. Mike, Brandon, Leah, and Ruthie have been with me on this journey even before graduate school had even started. Jesse and Kofi have become some of my closest friends throughout graduate school. I am glad to have all of you throughout these past years. I would also like to thank Andrew Lopez, Estelle Han, Kikuye Koyano, Lisa Gai, and Mukund Sudarshan for their support as friends and fellow scientists.

I would like to thank my family and extended family for supporting since the first day I was on this earth. My mother and father have shown me that there is no love that can rival a parent's love for their children. My brother, Andrew, inspired me to become a scientist and shaped a large portion of my values and personality. My sister, Ashley, has always given me support through an endless supply of hugs and been a role model for my professional career. My aunts, Lennie and Leeza, and my uncles, Ferdinand and Tommy, have always provided me with love and support while my cousins Barry, Bobby, Daniel, and Dana have always provided me with inspiration.

Lastly, I would like to thank my partner, Leah, for her constant and unwavering support throughout the past six years. She has been with me through some of the highest and lowest points of my academic career and life, and I could not hope for a better person to traverse them with.

Chapter Two of this dissertation is a version of Aman Agrawal*, Alec M. Chiu*, Minh Le, Eran Halperin, Sriram Sankararaman. “Scalable probabilistic PCA for large-scale genetic variation data.” PLOS Genetics 16(5): e1008773. 2020. “

Chapter Three is a version of Alec M. Chiu, Erin K. Molloy, Zilong Tan, Ameet Talwalkar, Sriram Sankararaman. “Inferring population structure in biobank-scale genomic data.“ The American Journal of Human Genetics 109(4), 727-737. 2022.

Chapter Four is a version of a manuscript in preparation by Alec M. Chiu, Sriram Sankararaman*, Noah Zaitlen*. “A simple statistical testing framework for detecting differences in variance and covariance in gene expression networks.“

This work was supported in part by the NIH Training Grant in Genomic Analysis and Interpretation (T32HG002536) and the NSF grant NRT-HDR: Modeling and Understanding Human Behavior: Harnessing Data from Genes to Social Networks (DGE-1829071).

VITA

2012–2016	BS, Biochemistry, University of California, Los Angeles, CA, USA

SELECTED PUBLICATIONS

* Denotes equal contribution

Hilary A. Coller, Stacey Beggs, Samantha Andrews, Jeff Maloy, **Alec Chiu**, Sriram Sankararaman, Matteo Pellegrini, Nelson Freimer, Tracy Johnson, Jeanette Papp, Eleazar Eskin, Alexander Hoffmann. Bruins-in-Genomics: Evaluation of the impact of a UCLA undergraduate summer program in computational biology on participating students. *PloS One.* 2022.

Alec M. Chiu, Erin K. Molloy, Zilong Tan, Ameet Talwalkar, Sriram Sankararaman. Inferring population structure in biobank-scale genomic data. *The American Journal of Human Genetics.* 2022.

COVID-19 Host Genetics Initiative. Mapping the human genetic architecture of COVID-19. *Nature.* 2021.

Ali Pazokitoroudi, **Alec M. Chiu**, Kathryn S. Burch, Bogdan Pasaniuc, Sriram Sankararaman. Quantifying the contribution of dominance deviation effects to complex trait variation in biobank-scale data. *The American Journal of Human Genetics.* 2021.

Daniel J. Tan, Mithun Mitra, **Alec M. Chiu**, Hilary A. Coller. Intron retention is a robust marker of intertumoral heterogeneity in pancreatic ductal adenocarcinoma. *NPJ Genomic*

Medicine. 2020.

Aman Agrawal*, **Alec M. Chiu***, Minh Le, Eran Halperin, Sriram Sankararaman. Scalable probabilistic PCA for large-scale genetic variation data. *PLoS Genetics.* 2020.

Alec M. Chiu, Mithun Mitra, Lari Boymoushakian, Hilary A. Coller. Integrative analysis of the inter-tumoral heterogeneity of triple-negative breast cancer. *Scientific Reports.* 2018.

Elvira Khialeeva, Joan W. Chou, Denise E. Allen, **Alec M. Chiu**, Steven J. Bensinger, Ellen M. Carpenter. Reelin deficiency delays mammary tumor growth and metastatic progression. *Journal of Mammary Gland Biology and Neoplasia.* 2017.

CHAPTER 1

Introduction

1.1 Scope of Research

The availability and advancement of technology has led to the collection of unprecedented amounts of biological data [1, 2, 3] and enabled the possibility of exploring biological phenomena in several novel ways [4, 5, 6, 7, 8]. Amongst these include studies that specifically aim to elucidate the relationship between a biological phenomena and a phenotype [9, 10]. For instance, the genome wide associaton study (GWAS) [9, 11, 12] aims to link genetic variation to the variability of traits such as physical characteristics (*e.g.* height, body mass index) [13, 14] or diseases [15, 16]. Though a plethora of such studies have been performed [9, 15], there are still several challenges preventing many of these findings from being fully utilized [17, 18, 19].

One such challenge is that of structure [17, 20, 21, 22]. Discovering causal relationships between novel findings and biological phenomena requires one to distinguish between structure directly related to the biological phenomena and structure due to confounding factors [20, 21, 22]. Such confounding structure can be inherent such as ancestry and relatedness [21, 22, 23] or technical such as batch effects [20, 24]. Being able to infer and remove confounding structure becomes critical in preventing spurious results. [17, 20, 21, 22].

A variety of methods to detect structure exist [4, 25, 26]; the most commonplace method being principal component analysis (PCA). Simple adjustment such as the inclusion of small number of factors such as the top principal components (PCs) in regression models have been shown to be able to successfully eliminate spurious discoveries [27]. Despite the ability of existing methods to discover and account for the majority of confounding structure, there are growing concerns about the effect of cryptic and residual population structure [28, 29], which emphasizes the importance of utilizing the most appropriate statistical models and methods for a specific dataset [30].

The increasing scale of massive datasets has led to the ability to uncover and study such fine-scale structure [1, 3], but the size of the data has created computational bottlenecks in terms of runtime and memory required to run structure detection methods. One approach is to utilize parallel computing infrastructure from the cloud [31, 32, 33], but the cost of using such methods is often obscure and can become cost-prohibitive. As a result, there is a need for scalable methods that can be run in reasonable amounts of time on typical computing environments.

1.2 Contributions and Overview

In this dissertation, we propose scalable computational and statistical methods to infer structure from large-scale genomic data. A key assumption used by our methods is the fact that only a small number of factors are often required for downstream analysis. As a result, we utilize several forms of dimensionality reduction (*i.e.* PCA) to reduce and summarize massive datasets into sizes that are much more managable to analyze. We also utilize the properties of genotype matrices in taking on a finite number of values (*e.g.* human genotypes take on values of 0, 1, or 2). This unique property allows us to capitalize on speed-ups through techniques such as the Mailman algorithm [34].

In Chapter 2, we propose ProPCA, a scalable probabilstic PCA method that can compute genetic PCs efficiently, to enable researchers to perform a standard analysis on large-scale genetic datasets within reasonable amounts of time on typical computing requirements. We show that ProPCA maintains high accuracy when compared to existing methods while reducing runtime and consuming reasonable amounts of memory. We apply ProPCA to the UK Biobank [1], a dataset containing half a million individuals, to compute the top five principal components within thirty minutes. We also utilize the probabilistic model of ProPCA to derive a novel statistical test for recent putative selection. Using the population structure inferred within the White British individuals in the UK Biobank and selection test, we identify several novel signals of putative recent selection. Further extensions of ProPCA such as the adaptation of its statistical model to handle missing data or correlation between features (*i.e.* linkage disequilibrium) can potentially uncover less apparent structure by making the model more data-specific [35, 36].

Chapter 3 describes SCOPE, a novel method for inferring admixture proportions from biobank-scale data. SCOPE utilizes a previously proposed model [37] that consists of a dimensionality reduction step followed by a factorization step. SCOPE improves upon this model by integrating algorithmic speed-ups to both these steps through randomized eigendecomposition [38] and the Mailman algorithm [34]. We apply SCOPE to large-scale simulations and show that it is able to maintain competitive accuracy to state of the art methods while completing 3-144 times faster. We apply SCOPE in an unsupervised fashion to the UK Biobank, which at the time of writing and to our knowledge, is the first method of its kind to be able to do so. Our analysis revealed fine-scale structure present in several genomic datasets, but opens questions on the interpretability of deep structure in genomic datasets.

Chapter 4 describes a testing framework for discovering differences in variance and covariance structures. Several methods such as PCA can often miss sources of structure due the linear nature of PCA. As a result, several non-linear dimensionality reduction methods have been developed, but lack interpretability [39, 40]. We utilize two transformations that specifically allow one to exclusively test for variance or covariance differences by extending the eigengene testing framework found in weighted correlation network analysis (WGCNA) [41]. We apply our testing framework to RNA-seq data from individuals with common psychiatric diseases. Our analysis reveals several instances of variance and covariance changes despite there being no differences in mean expression; emphasizing the need to examine data beyond mean differences. In particular, we find several differences in variance and covariance between schizophrenia and bipolar disorder, two diseases which are highly correlated. Furthermore, we apply our method to cancer methylation data and stock market data to highlight our method's ability to extend to other data types. Our work opens up several ideas on how transformations, such as those employed in our testing framework, can be applied to other applications such as clustering and quantitative trait loci mapping.

CHAPTER 2

Scalable probabilistic PCA for large-scale genetic variation data

2.1 Background

Inference of population structure is a key step in population genetic analyses [42] with applications that include understanding genetic ancestry [43, 44, 45] and controlling for confounding in genome-wide association studies (GWAS) [46]. While several methods have been proposed to infer population structure (e.g., [47, 48, 49, 50, 51]), principal component analysis (PCA) is one of the most widely used [52, 47]. Unfortunately, the naive approach for estimating principal components (PCs) by computing a full singular value decomposition (SVD) scales quadratically with sample size (for datasets where the number of SNPs is larger than sample size), resulting in runtimes unsuitable for large data sets.

In light of these challenges, several solutions have been proposed for the efficient computation of PCs. One approach taken by many recent scalable implementations (FastPCA [53], FlashPCA2 [54], bigsnpr [55], TeraPCA [56], PLINK2 [57]) takes advantage of the fact that typical applications of PCA in genetics only require computing a small number of top PCs; *e.g.* GWAS typically use 5-20 PCs to correct for stratification [58]. These methods can be grouped according to their underlying algorithm: blanczos (FastPCA, PLINK2, TeraPCA) or the implicitly restarted Arnoldi algorithm (FlashPCA2, bigsnpr). An alternative approach for efficient computation of PCs takes advantage of the parallel computation infrastructure of the cloud [32]. However, the cost of cloud usage is roughly proportional to the number of CPU hours used by these algorithms, making them cost-prohibitive. Finally, these scalable implementations lack a full probabilistic model, making them challenging to extend to settings with missing genotypes or linkage disequilibrium (LD) between SNPs.

In this work, we describe ProPCA, a scalable method to compute the top PCs on genotype data. ProPCA is based on a previously proposed probabilistic model [59, 60], of which PCA is a special case. While PCA treats the PCs and the PC scores as fixed parameters, probabilistic PCA imposes a prior on the PC scores. This formulation leads to an iterative Expectation Maximization (EM) algorithm for computing the PCs. ProPCA leverages the structure of genotype data to further reduce the computation time in each iteration of the EM algorithm. The EM algorithm requires only a small number of iterations to obtain accurate estimates of the PCs resulting in a highly scalable algorithm.

ProPCA obtains a computational speed-up through the integration of the Mailman algorithm [34] into its EM algorithm. The Mailman algorithm allows for fast matrix-vector multiplication when there are a finite number of values (e.g. genotypes) in exchange for additional memory usage. As a result, ProPCA requires more memory than some of the other scalable PCA methods. However, the increased memory consumption is reasonable; often still within the memory available within typical computing environments.

In both simulated and real data, ProPCA is able to accurately infer the top PCs while scaling favorably with increasing sample size. We applied ProPCA to compute the top five PCs on genotype data from the UK Biobank, consisting of 488,363 individuals and 146,671 SNPs, in less than thirty minutes. To illustrate how the ability to compute PCs in large samples can lead to biological discovery, we leveraged the population structure inferred by ProPCA within the White British individuals in the UK Biobank [1] to scan for SNPs that are not well-modeled by the top PCs and, consequently, identify several novel genome-wide signals of recent positive selection. Our scan recovers sixteen loci that are highly differentiated across the top five PCs that are likely signals of recent selection. While these loci include previously reported targets of selection [53], the larger sample size that we analyze here allows us to identify eleven novel signals including a missense mutation in *RPGRIP1L* ($p = 2.09 \times 10^{-9}$) and another in *TLR4* ($p = 7.60 \times 10^{-12}$).

A number of algorithms that analyze genotype data, including methods for heritability estimation and association testing, can be modeled as iterative procedures where the core computational operation is similar to that solved by ProPCA. Thus, the algorithm that we employ in this work can potentially lead to highly scalable algorithms for a broad set of population genetic analyses.

2.2 Materials and Methods

2.2.1 Principal Components Analysis (PCA)

We observe genotypes from n individuals at m SNPs. The genotype vector for individual i is a length m vector denoted by $\boldsymbol{g}_i \in \{0,1,2\}^m$. The j^{th} entry of $\boldsymbol{g}_i$ denotes the number of minor allele carried by individual i at SNP j. Let $\boldsymbol{G}$ be the $m \times n$ genotype matrix where $\boldsymbol{G} = [\boldsymbol{g}_1 \dots \boldsymbol{g}_n]$. Let $\boldsymbol{Y}$ denote the matrix of standardized genotypes obtained by centering and rescaling each row of the genotype matrix $\boldsymbol{G}$ so that $\sum_j y_{i,j} = 0$ and $\sum_j y_{i,j}^2 = 1$ for all $i \in \{1, \dots, m\}$.

Principal components analysis (PCA) [52] attempts to find a low-dimensional linear transformation of the data that maximizes the projected variance or, equivalently, minimizes the reconstruction error. Given the $m \times n$ matrix $\boldsymbol{Y}$ of standardized genotypes and a target dimension k, PCA attempts to find a $m \times k$ matrix with orthonormal columns $\boldsymbol{W}$ and $n \times k$ matrix $\boldsymbol{Z}$ that minimizes the reconstruction error: $\|\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{Z}^{\mathrm{T}}\|_F$ where $\|A\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$ is the Frobenius norm of the matrix $\boldsymbol{A}$. To solve the PCA problem, we perform a singular-value decomposition (SVD) of the standardized genotype matrix $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$ and set $\widehat{\boldsymbol{W}} = \boldsymbol{U}_K$, where $\boldsymbol{U}_K$ is a $m \times k$ matrix containing the k columns of $\boldsymbol{U}$ corresponding to the k largest singular vectors of $\boldsymbol{Y}$.

2.2.2 Probabilistic PCA

PCA can be viewed as a limiting case of the probabilistic PCA model [51, 59, 60]. Probabilistic PCA models the observed data $\boldsymbol{y}_i \in \mathbb{R}^m, i \in \{1 \dots, n\}$ as a linear transformation of a k-dimensional latent random variable $\boldsymbol{x}_i$ ($k \leq m$) with additive Gaussian noise. Denoting the linear transformation by the $m \times k$ matrix $\boldsymbol{C}$, and the (m-dimensional) noise by $\boldsymbol{\epsilon}_i$ (with isotropic covariance matrix $\sigma^2 \boldsymbol{I}_m$), the generative model can be written as

$$\begin{aligned} \boldsymbol{y}_i | \boldsymbol{x}_i ; \boldsymbol{\epsilon}_i &= \boldsymbol{C}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i \\ \boldsymbol{x}_i &\overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I}_k) \\ \boldsymbol{\epsilon}_i &\overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_m) \qquad (2.1) \end{aligned}$$

The maximum likelihood estimate of the matrix $\boldsymbol{C}$ in this model has been shown to span the k-dimensional principal subspace of the data $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots \boldsymbol{y}_n]$ [61].

2.2.3 EM algorithm for PCA

Since probabilistic PCA is a probabilistic model endowed with latent variables, the EM algorithm presents a natural approach to compute the maximum likelihood estimates of the model parameters $(\boldsymbol{C}, \sigma^2)$ [59, 60]. The EM algorithm for learning the principal components can be derived as a special case of the EM algorithm for the probabilistic PCA model where the variance of the observation noise σ^2 tends to zero leading to these updates:

$$\text{E-step}: \qquad \boldsymbol{X} = (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{Y} \qquad (2.2)$$

$$\text{M-step}: \qquad \boldsymbol{C} = \boldsymbol{Y}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1} \qquad (2.3)$$

Here $\boldsymbol{X} = [\boldsymbol{x}_1 \ldots \boldsymbol{x}_n]$ is a $k \times n$ matrix and $\boldsymbol{Y} = [\boldsymbol{y}_1 \ldots \boldsymbol{y}_n]$ is a $m \times n$ matrix. Noting that all matrix inversions require inverting a $k \times k$ matrix, the computational complexity of the E-step is $\mathcal{O}(k^2m + k^3 + k^2m + mnk)$ while the computational complexity of the M-step is $\mathcal{O}(k^2n + k^3 + k^2n + mnk)$. For small k and large m, n, the per-iteration runtime complexity is $\mathcal{O}(mnk)$. Thus, the EM algorithm provides a computationally efficient estimator of the top k PCs when the number of PCs to be estimated is small.

2.2.4 Sub-linear time EM

The key bottleneck in the EM algorithm is the multiplication of the matrix $\boldsymbol{Y}$ with matrices $\boldsymbol{E} = (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T$ and $\boldsymbol{M} = \boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}$.

The vectors representing the sample mean and standard deviation of the genotypes at each SNP are denoted $\overline{\boldsymbol{g}}$ and $\boldsymbol{s}$. Assuming no entry in $\boldsymbol{s}$ is zero (we remove SNPs that have no variation across samples), the matrix of standardized genotypes $\boldsymbol{Y}$ can be written as:

$$\boldsymbol{Y} = diag(\boldsymbol{s})^{-1}\boldsymbol{G} - \boldsymbol{\rho}\mathbf{1}_n^{\mathrm{T}}$$

Here $diag(\boldsymbol{x})$ is an operator that constructs a diagonal matrix with the entries of $\boldsymbol{x}$ along its diagonals, $\mathbf{1}_n$ is a length n vector with each entry equal to one, and $\boldsymbol{\rho}$ is a length m vector with $\rho_j = \frac{\overline{g}_j}{s_j}, j \in \{1, \ldots, m\}$.

The EM updates can be written as:

$$\begin{aligned}
\boldsymbol{X} &= \boldsymbol{E}\boldsymbol{Y} = \boldsymbol{E}\ \mathsf{diag}\left(\boldsymbol{s}\right)^{-1}\boldsymbol{G} - \boldsymbol{E}\boldsymbol{\rho}\mathbf{1}_n^{\mathrm{T}} \\
&= \widetilde{\boldsymbol{E}}\boldsymbol{G} - \boldsymbol{E}\boldsymbol{\rho}\mathbf{1}_n^{\mathrm{T}} \qquad (2.4) \\
\boldsymbol{C} &= \boldsymbol{Y}\boldsymbol{M} = diag(\boldsymbol{s})^{-1}\boldsymbol{G}\boldsymbol{M} - \boldsymbol{\rho}\mathbf{1}_n^{\mathrm{T}}\boldsymbol{M} \qquad (2.5)
\end{aligned}$$

Here $\widetilde{\boldsymbol{E}}$ can be computed in time $\mathcal{O}(km)$ while $\boldsymbol{E}\boldsymbol{\rho}\mathbf{1}_n^{\mathrm{T}}$ and $\boldsymbol{\rho}\mathbf{1}_n^{\mathrm{T}}\boldsymbol{M}$ can be computed in time $\mathcal{O}(nk + mk)$.

The key bottleneck in the E-step is the multiplication of the genotype matrix $\boldsymbol{G}$ by each of the k rows of the matrix $\widetilde{\boldsymbol{E}}$ and in the M-step, multiplication of $\boldsymbol{G}$ by each of the k columns of the matrix $\boldsymbol{M}$ respectively. Leveraging the fact that each element of the genotype matrix $\boldsymbol{G}$ takes values in the set $\{0, 1, 2\}$, we can improve the complexity of these multiplication operations from $\mathcal{O}(nmk)$ to $\mathcal{O}(\frac{nmk}{\max(\log_3 n, \log_3 m)})$ by extending the Mailman Algorithm [34]. For additional implementation details, see Appendix A.5.

2.2.5 The Mailman algorithm

In the M-step, we need to compute $\boldsymbol{c} = \boldsymbol{A}\boldsymbol{b}$ for an arbitrary real-valued vector $\boldsymbol{b}$ and a $m \times n$ matrix $\boldsymbol{A}$ whose entries take values in $\{0, 1, 2\}$. We assume that $m = \lceil \log_3(n) \rceil$. Naive matrix-vector multiplication takes $\mathcal{O}(\lceil \log_3(n) \rceil n)$ time.

The Mailman algorithm decomposes $\boldsymbol{A}$ as $\boldsymbol{A} = \boldsymbol{U}_n\boldsymbol{P}$. Here $\boldsymbol{U}_n$ is the $m \times r$ matrix whose columns containing all $r = 3^m$ possible vectors over $\{0, 1, 2\}$ of length m. We set an entry $P_{i,j}$ to 1 if column j of $\boldsymbol{A}$ matches column i of U_n: $A^{(j)} = U_n^{(i)}$. The decomposition of any matrix $\boldsymbol{A}$ into $\boldsymbol{U}_n$ and $\boldsymbol{P}$ can be done in $\mathcal{O}(nm)$ time. Given this decomposition, the desired product $\boldsymbol{c}$ is computed in two steps, each of which has $\mathcal{O}(n)$ time complexity [34]:

$$\boldsymbol{d} = \boldsymbol{P}\boldsymbol{b}, \quad \boldsymbol{c} = \boldsymbol{U}_n\boldsymbol{d}$$

The Mailman algorithm provides computational savings in a setting where the cost of computing the decomposition of $\boldsymbol{A}$ are offset by the gains in repeated multiplication involving $\boldsymbol{A}$.

Similarly, in the E-step, we need to compute $\boldsymbol{f}^{\mathrm{T}}\boldsymbol{A}$ in $\mathcal{O}(\lceil \log_3(n) \rceil n)$ time by computing $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{f}$ and computing a decomposition of $\boldsymbol{A}^{\mathrm{T}}$. A drawback of this approach is the need to store both decompositions that would double the memory requirements of the algorithm. Instead, we propose a novel variant of the Mailman algorithm that can compute $\boldsymbol{f}^{\mathrm{T}}\boldsymbol{A}$ in $\mathcal{O}(\lceil \log_3(n) \rceil n)$ time using the same decomposition as $\boldsymbol{A}$ (Appendix A.6).

Additional details on efficient implementation of the EM and Mailman algorithms can be found in Appendix A.5.

2.2.6 Simulations

We simulated genotypes at m independent SNPs across n individuals in which a single ancestral population diverged into q sub-populations with drift proportional to the F_{st}, a measure of population differentiation. The allele frequency at SNP $f_{j,0}, j \in \{1, \ldots, m\}$ in the ancestral population was sampled from a uniform distribution such that $f_{j,0} \overset{iid}{\sim} Unif(0.05, 0.95)$. Allele frequencies in each of the l subpopulations were generated by simulating neutral drift from the ancestral allele frequency, $f_{j,l} \sim \mathcal{N}(f_{j,0}, f_{j,0}(1 - f_{j,0})F_{st}), l \in \{1, \ldots, q\}$ and were set to 0 or 1 if they fell outside the interval $[0, 1]$. The genotypes of an individual in population l at SNP j was sampled from a $Binomial(2, f_{j,l})$ distribution.

2.2.7 Benchmarking

To compare estimated PCs to reference PCs, we computed the mean of explained variance (MEV) – a measure of the overlap between the subspaces spanned by the two sets of PCs. Two different sets of K principal components each produce a K-dimensional column space. A metric for the performance of a PCA algorithm against some baseline is to see how much the column spaces overlap. This is done by projecting the eigenvectors of one subspace onto the other and finding the mean lengths of the projected eigenvectors. If we have a reference set of PCs $(v_1, v_2, ..., v_k)$ against which we wish to evaluate the performance of a set of estimated PCs $(u_1, u_2, ..., u_k)$, $MEV = \frac{1}{k}\sum_{i=1}^{k}\sqrt{\sum_{j=1}^{k}(\mathbf{v_i} \cdot \mathbf{u_j})^2} = \frac{1}{k}\sum_{i=1}^{k}\|\mathbf{U^T v_i}\|$ where $\mathbf{U}$ is a matrix whose column vectors are the PCs which we are testing.

In practice, when attempting to compute the top k PCs, ProPCA was found to converge faster by computing l PCs for $l > k$ PCs and retaining the top k PCs. The reason for this is that in the initial iterations of the EM algorithm, the estimates of the top PCs are noisy. We set $l = k$ in our experiments for an effective $2k$. While ProPCA could be run to convergence, we found that running it for k iterations already gave accurate results across the range of parameters considered. Our empirical results are consistent with our theoretical result that the EM algorithm converges exponentially fast in the spectral norm of the error matrix [38, 62] (Appendix A.7).

We compared ProPCA to the current state-of-the-art methods for computing PCs from genotype data: the SVD implementation in PLINK (PLINK_SVD [63]), FastPCA [53], FlashPCA2 [54], bigsnpr [55], PLINK2 [57], and TeraPCA [56]. PLINK_SVD refers to an exact computation of PCs using the full Singular Value Decomposition as implemented in the PLINK package (PLINK_SVD). FastPCA [53] is an implementation of the blanczos method, a randomized subspace iteration method [38] while FlashPCA2 [54] is an implementation of the implicitly restarted Arnoldi method [64]. PLINK2 [57] and TeraPCA [56] are reimplimentations of the FastPCA algorithm while bigsnpr [55] is a reimplimentation of the FlashPCA2 algorithm designed to utilize disk space as a file backend. We used default parameters for all methods unless otherwise stated. For benchmarking of bigsnpr, we included the creation of the file backend in timing as it is required to run any of the computations included in the backend. Furthermore, we excluded bigsnpr from some experiments due to the inability of its PCA function to natively handle missing data and when faced with monomorphic SNPs. All experiments were performed on a Intel(R) Xeon(R) CPU 2.10GHz server with 128 GB RAM, restricted to a single core, capped to a maximum runtime of 100 hours and a maximum memory of 64 GB.

2.2.8 Selection scan

The White British cohort was identified by the UK Biobank as participants who self-identified as 'British' within the broader-level group 'White' while having similar ancestral background [1]. For our selection scan, we further filtered the $409,634$ individuals in the White British subset to obtain an unrelated White British subset by removing individuals with one other related individual in the data set (individuals with kinship coefficients greater than 0.0625 (third-degree relatedness) to any other individual as determined by KING [65]). After removing these individuals, we obtained an unrelated White British subset containing $276,736$ individuals.

We inferred the top five PCs using ProPCA on all $276,736$ unrelated White British individuals and a filtered SNP set containing $146,671$ SNPs (UK Biobank SNP set). SNPs in the UK Biobank SNP set consist of SNPs on the UK Biobank Axiom array from which SNPs were removed if they have missing rates greater than 1.5%, minor allele frequencies (MAF) less than 1%, or if they were in regions of long-range linkage disequilibrium. The remaining SNPs were then pruned for pairwise r^2 less than 0.1 using windows of 1000 base pairs (bp) and a step-size of 80 bp.

We developed a selection statistic to search for SNPs whose variation is not well-explained by the ProPCA model, closely related to the selection statistic proposed in [53] (Appendix A.2). Under the probabilistic PCA model, the normalized genotype matrix is modeled by a low rank approximation and Gaussian noise, $\mathbf{Y} = \mathbf{C}\mathbf{X} + \boldsymbol{\epsilon}$. Given our low rank approximation of the genotype matrix, $\hat{\mathbf{Y}} = \mathbf{C}\mathbf{X}$, we have the residual : $\mathbf{Y} - \hat{\mathbf{Y}} = \boldsymbol{\epsilon}$. For a SNP j, the Gaussian noise, $\boldsymbol{\epsilon_j} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$. Projecting this residual onto a PC results in a univariate Gaussian with zero mean and constant variance across SNPs. This variance can be estimated as the sample variance $\hat{\sigma}^2$ of the resulting statistics across SNPs. In summary, we propose the statistic: $\frac{((\mathbf{y}_j - \hat{\mathbf{y}}_j)^T \mathbf{x}_k)^2}{\hat{\sigma}^2} \sim \chi^2_1$ for SNP j, given the k-th PC. The projection of the residual onto a PC allows the signal of selection to be interpreted in the context of deviations from ancestry captured by the specific PC.

Furthermore, a variant of this statistic, which we call the combined statistic, can be generated from the selection statistics computed on each individual PC using the observation that the resulting chi-squared statistics are independent of each other. This allows us to create an additional statistic by summing the individual PC statistics to create a combined statistic that follows a chi-squared distribution with additional degrees of freedom for each PC used.

Using the results from the PCA on the UK Biobank SNP set, we performed our selection scan on a different set of $516,140$ SNPs. We generated this set of SNPs by removing SNPs that were multi-allelic, had genotyping rates less than 99%, had minor allele frequencies less than 1%, and were not in Hardy-Weinberg equilibrium ($p < 10^{-6}$).

We performed an allele frequency test for each novel SNP using the Nomenclature of Territorial Units for Statistics level 3 (NUTS3) classification of regions for the UK. The NUTS3 classification defines non-overlapping borders for each region in the UK, allowing us to uniquely map each individual to a region in the UK using their birth location coordinates by checking which NUTS3 regions they fell into. For each of our novel loci, we then performed an two-tailed Z-test between each region's allele frequency against all other regions. We corrected for multiple testing using the Bonferroni correction.

2.3 Results

2.3.1 Accuracy

We first assessed the accuracy of ProPCA using the simulation framework described in the Methods. We generated datasets containing $50,000$ SNPs and $10,000$ individuals across q populations, where q was chosen to be 5 and 10. The populations were simulated with varying levels of population differentiation that are typical of present-day human populations (values of F_{st} ranging from 0.001 to 0.01) and were small enough so that we could compute the full SVD thereby allowing us to estimate the accuracy of the PCs computed by ProPCA. To measure accuracy, we computed the mean of explained variances (MEV), a measure of the overlap between the subspaces spanned by the PCs estimated by ProPCA compared to the PCs computed using a full SVD (Methods). ProPCA, All methods are able to estimate highly accurate PCs (values of MEV close to 1) across the range of parameters (Table 2.1).

2.3.2 Runtime

We assessed the scalability of ProPCA with increasing sample size (Methods). We simulated genotypes from six populations containing $100,000$ SNPs and sample sizes varying from $10,000$ to $1,000,000$ with $F_{st} = 0.01$.

We compared the wall-clock time for running ProPCA, the SVD implementation in PLINK (PLINK_SVD [63]), FastPCA [53], FlashPCA2 [54], bigsnpr [55], PLINK2 [57], TeraPCA [56]). The SVD implementation in PLINK could not run in reasonable time on datasets exceeding $70,000$ individuals (Fig. 2.1a). While all the other methods scale with sample size, ProPCA is faster than the methods compared against (Figure 2.1b). ProPCA computes PCs in about 30 minutes even on the largest data containing a million individuals and $100,000$ SNPs. We similarly explored how each method scale in terms of the number of variants. We repeated our experiment by varying the number of SNPs from $10,000$

Table 2.1: **ProPCA accurately estimates principal components relative to other methods**

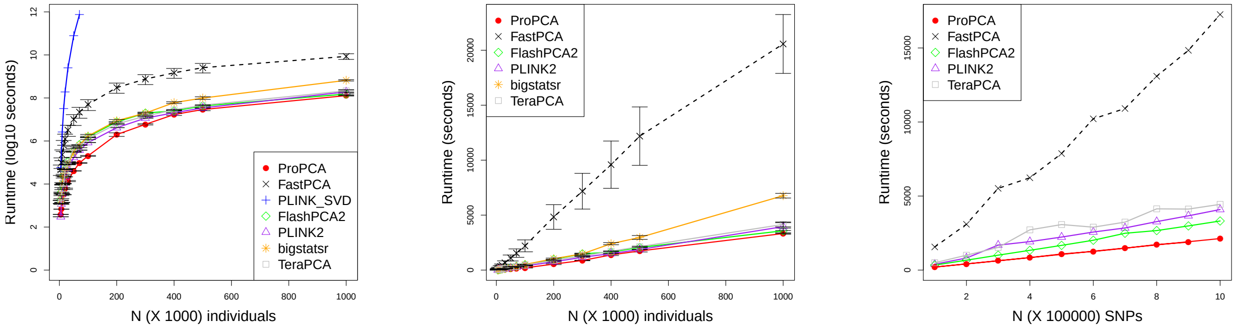
F_{st}	ProPCA		FlashPCA2		fastPCA		PLINK2		bigsnpr		TeraPCA	
	$K=5$	$K=10$	$K=5$	$K=10$	$K=5$	$K=10$	$K=5$	$K=10$	$K=5$	$K=10$	$K=5$	$K=10$
0.001	0.987	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.999
0.002	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.003	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.004	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.005	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.006	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.007	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.008	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.009	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
0.010	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

The principal components computed by ProPCA are compared to the PCs obtained from a full SVD on a genotype dataset containing $50,000$ SNPs and $10,000$ individuals. Accuracy was measured by the mean of explained variance (MEV) which measures the overlap between the set of PCs inferred from ProPCA and those from SVD across values of $F_{st} \in \{0.001, \ldots, 0.01\}$. We report MEV for $K = 5$ using 5 populations as well as for $K = 10$ PCs using 10 populations. Methods shown are run using their default parameters.

to $1,000,000$ while keeping the sample size constant at $100,000$ and found similar results (Figure 2.1c).

We further tested runtime as a function of the number of PCs on a simulated dataset containing 10,000 individuals, 50,000 SNPs, and 20 latent populations separated at $F_{ST} = 0.01$. We find that PLINK2 and ProPCA scale linearly when computing the upto the top 40 PCs (Figure A.1). FlashPCA2 is efficient at computing 2-20 PCs, but increases in runtime when computing a single PC or more than 20 PCs. We found that this trend was both reproducible across different datasets. We also tested bigsnpr/bigstatsr, which uses the same underlying algorithm as FlashPCA2 and found a similar trend, *i.e.*, its performance to be similar to FlashPCA2 in that it is efficient at computing 1-20 PCs, but we see a steady increase in runtime after 20 PCs.

We tested a final scenario in which each method computed 40 PCs on our two largest simulated datasets containing one million SNPs and 10,000 individuals dataset as well as the one million individuals and 10,000 SNPs dataset (Table 2.2). We find that ProPCA can



(a) (b) (c)

Figure 2.1: **ProPCA is computationally efficient.** Comparison of runtimes over simulated genotype data varied over individuals and SNPs. Figures 2.1a and 2.1b display the total runtime containing $100,000$ SNPs, six subpopulations, $F_{st} = 0.01$ and individuals varying from $10,000$ to $1,000,000$. We report the mean and standard deviation over ten trials. Figure 2.1b compares the runtimes of all algorithms excluding PLINK_SVD which could only run successfully up to a sample size of $70,000$. Figure 2.1c displays the total runtime containing $100,000$ individuals, six subpopulations, $F_{st} = 0.01$, and SNPs varying from $10,000$ to $1,000,000$. All methods were capped to a maximum of 100 hours and a maximum memory of 64 GB and run using default settings. We were unable to include bigstatsr in the SNP benchmark as it does not allow for monomorphic SNPs.

Table 2.2: **Runtimes of methods on largest simulated datasets for 40 principal components.**

Method	SNPs	Individuals
bigstatsr	-	103
FastPCA	-	-
FlashPCA2	93	114
PLINK2	74	72
ProPCA	35	28
TeraPCA	49	48

We computed 40 PCs from each method on each of our largest simulated datasets. Times are reported in hours. The 'SNPs' column contains the runtime on a 1 million SNP and 10,000 individuals dataset while the 'Individuals' contains the runtime on a 1 million individual and 10,000 SNP dataset. FastPCA could not be run to completion on either dataset due to a segmentation fault while bigstatsr could not run on the SNPs dataset due to the inclusion of monomorphic SNPs. All methods were run with default parameters except TeraPCA, which was run with '-rfetched 4000' for the SNPs dataset and '-rfetched 2000' for the Individuals dataset due to a segementation fault.

compute the 40 PCs most efficiently under two days for both datasets while other methods required 2-4 days.

Since ProPCA, FastPCA, and FlashPCA2 are all based on iterative algorithms, their runtimes depend on details of convergence criterion. We performed an additional experiment to compare the runtime of ProPCA, FastPCA (for which we could instrument the source code) for a single iteration and found ProPCA to be three to four times faster than FastPCA across the range of sample sizes (Figure A.2).

Measuring the accuracy of the PCs (MEV) as a function of runtime (on datasets with a range of F_{st} containing $50,000$ SNPs and $10,000$ individuals so that we could compare the estimated PCs to exact PCs), ProPCA attains a given MEV in about half the time as FastPCA and FlashPCA2 (Figure A.3).

2.3.3 Memory

We assessed the memory usage of ProPCA and other methods as a function of individuals and SNPs (Figurse A.4a, A.4b). Due to computations utilized by the Mailman algorithm, ProPCA uses more memory than other methods, but is still relatively efficient requiring about 40 GB on the largest dataset. Memory usage for ProPCA scales linearly with respect both individuals and SNPs.

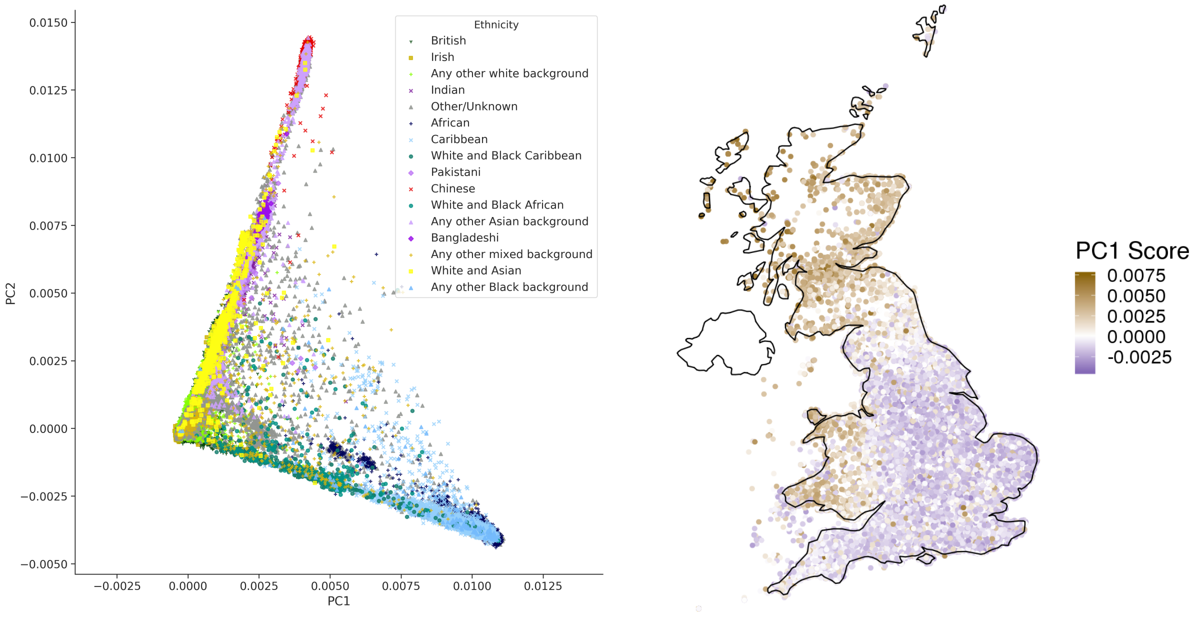
2.3.4 Application to real genotype data

We applied ProPCA to genotype data from Phase 1 of the 1000 Genomes project [66]. On a dataset of 1092 individuals and $442,350$ SNPs, ProPCA computes the top forty PCs that are qualitatively indistinguishable from running a full SVD (Figure A.5). Furthermore, we tested each method's ability to compute 5-40 PCs on this dataset. We took a small subset for 450k SNPs and 1,092 individuals for which we could compute the full SVD. We tested all methods at increments of 5 PCs to 40 PCs and ultimately found that all that all methods still performed well across the range tested (MEV $\geq$ 0.95) (Table A.1). We also applied ProPCA to genotype data from the UK Biobank [1] consisting of $488,363$ individuals and $146,671$ SNPs after QC. ProPCA can compute the top five PCs in about 30 minutes and the resulting PCs reflect population structure within the UK Biobank, consistent with previous studies [1] (Fig. 2.2a).

2.3.5 Application to scans for selection

Since the PCs in ProPCA are computed as maximum likelihood estimates under a probabilistic model, ProPCA provides a natural framework for applications such as hypothesis testing. By utilizing the statistical assumptions and set up provided by the ProPCA model, we developed a statistical test to search for SNPs that are not well-modeled by the ProPCA model as a means of discovering signals of natural selection (Methods and Appendix A.1). This statistic relies on the observation that a SNP evolving under positive selection is expected to exhibit differentiation in the frequencies of its alleles that is extreme compared to a typical SNP that is evolving neutrally [67].

Since deviations from the ProPCA model can occur due to reasons unrelated to selection, we filtered out SNPs with high rates of missingness, low minor allele frequency (MAF), and



(a)

(b)

Figure 2.2: **Principal components uncover population and geographic structure in the UK Biobank** We used ProPCA to compute PCs on the UK Biobank data. Figure 2.2a shows the first two principal components to reveal population structure. Figure 2.2b shows geographic structure by plotting the score of 276, 736 unrelated White British individuals on the first principal component on their birth location coordinates.

presence in regions of long-range LD [68](Methods). We ran ProPCA to infer the top five PCs on $276,736$ unrelated White British samples and the UK Biobank SNP set consisting of $146,671$ SNPs obtained by further removing SNPs in high LD (Figure A.6).

The Pearson correlation coefficient between birth location coordinates and the PC score for each individual reveals that the estimated PCs capture geographic structure within the UK (Fig. 2.2b, A.7, A.2). We used these PCs to perform a selection scan on a larger set of $516,140$ SNPs and we report SNPs that are genome-wide significant after accounting for the number of SNPs as well as PCs tested (p-value $< \frac{0.05}{6\times 516,140}$; we use 6 to account for the additional combined test statistic that we describe later). We ensured that the selection statistic for each PC was well-calibrated against a χ^2_1 distribution (Fig. A.8) and genomic inflation (λ_{GC}) values for each of the PCs showed no substantial inflation (Table A.3). While our statistic is closely related to a previously proposed statistic to detect selection on PCs (Appendix A.2), we found that our proposed statistic is better calibrated (Table A.3).

Our scan revealed a total of 59 SNPs that were genome-wide significant (Table A.4). Clustering these signals into 1 Mb windows centered around the most significant SNP for each PC, we obtained twelve non-overlapping loci that contain putative targets of selection (Figure 2.3, Table A.5). These twelve loci include five that were previously reported to be signals of selection in the UK with genome-wide significance: *LCT* (*rs7570971* with $p = 8.51 \times 10^{-16}$), *TLR1* (*rs5743614*, $p = 5.65 \times 10^{-25}$), *IRF4* (*rs62389423*, $p = 8.80 \times 10^{-42}$), *HLA* (*rs9267817*, $p = \times 6.17 \times 10^{-9}$), and *FUT2* (*rs492602*, $p = 7.02 \times 10^{-10}$) [53]. The larger sample size that we analyze here also reveals novel signals at additional loci. Four of the twelve signals were previously suggested to be signals of selection but were not genome-wide significant: *HERC2* (*rs12913832*, $p = 5.21 \times 10^{-10}$), *RPGRIP1L* (*rs61747071*, $p = 2.09 \times 10^{-9}$), *SKI* (*rs79907870*, $p = 2.58 \times 10^{-9}$), *rs77635680* ($p = 2.22 \times 10^{-10}$) [53] while the remaining three loci: *HERC6* (*rs112873858*, $p = 2.68 \times 10^{-11}$), *rs6670894* ($p = 4.98 \times 10^{-9}$), and *rs12380860* ($p = 8.62 \times 10^{-9}$) appear to be previously unreported.

To validate our findings, we utilized birth location coordinates for each individual and assigned them to geographical regions in the UK as defined in the Nomenclature of Territorial

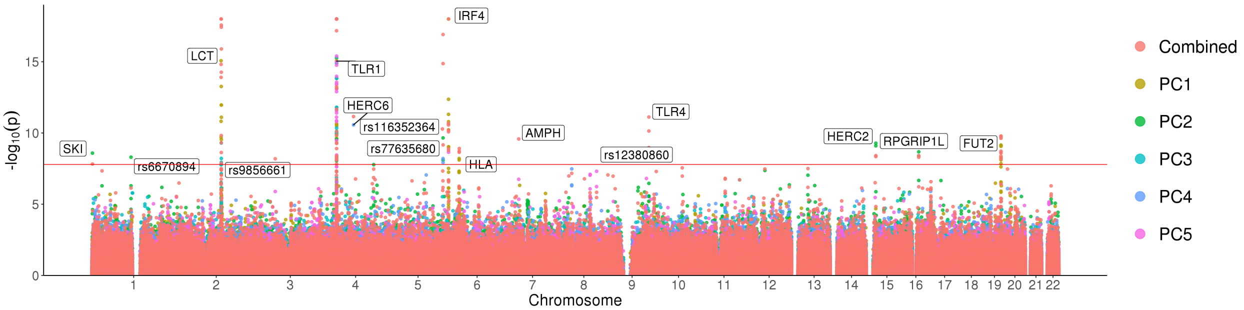


Figure 2.3: **Selection scan for the first five principal components in the white British individuals in the UK Biobank**: A Manhattan plot with the $-\log_{10} p$ values associated with the test of selection displayed for the first five principal components for the unrelated White British subset of the UK Biobank. The red line represents the Bonferroni adjusted significance level ($\alpha = 0.05$). Significant loci are labeled. Signals above $-\log_{10}(p) = 18$ were capped at this value for better visualization.

Units for Statistics level 3 (NUTS3) classification. We performed a test of association between the allele frequency of the top SNP in each of our novel loci with geographical regions and confirmed that SNPs identified in our selection scan show differences in allele frequencies across specific geographical regions (Table A.6).

One of the novel genome-wide significant loci is *RPGRIP1L*. *RPGRIP1L* is a highly conserved gene that encodes a protein that localizes at primary cilia and is important in development [69]. Mutations in this gene have been implicated with neurological disorders such as Joubert syndrome and Meckel syndrome [70], conditions that sometimes also result in additional symptoms such as eye diseases and kidney disorders [71]. The SNP with the most significant p-value in our scan in *RPGRIP1L*, *rs61747071*, is a missense loss-of-function mutation A229T that has been shown to lead to photoreceptor loss in ciliopathies [72].

We created an additional variant of our selection statistic which tests for SNPs that are not well-modeled by a linear combination of the first five PCs by summing the per-PC χ_1^2 statistics resulting in a new chi-squared statistic with five degrees of freedom. Combining signals across PCs has been previously shown to boost power in association testing [73]. We verified that the resulting combined statistic is also calibrated (Figure A.8, Table A.3). Under this combined statistic, we recover majority of the loci found on each individual PC, but we also discover four additional novel loci: *AMPH* (*rs118079376*, $p = 2.64 \times 10^{-10}$), *TLR4* (*rs4986790*, $p = 7.60 \times 10^{-12}$), *rs9856661* ($p = 6.46 \times 10^{-9}$), and *rs116352364* ($p = 5.24 \times 10^{-11}$) (Table A.7).

TLR4 is a member of the toll-like receptor family. The *TLR* gene family is known to play a fundamental role in pathogen recognition and activation of innate immunity, but *TLR4* in particular is involved with proinflammatory cytokines and has a pro-carcinogenic function [74]. The SNP with the most significant p-value at our $TLR4$ locus is *rs4986790*, a missense D299G mutation and D259G mutation on two different transcripts for the *TLR4* gene. The D299G mutation is of particular interest as this mutation is strongly correlated with increased infection by *Plasmodium falciparum*, a parasite that causes malaria [75, 76].

To better understand the signals of selection that the proposed statistic is sensitive to, we compared the time-scale for our selection hits to those from a recent study that is designed to detect recent positive selection [77] (Figure A.9). Using estimates of allelic ages for variants in the 1000 Genomes Project [78], we find that the variants detected by the proposed statistic tend to be older on average than those found to have a singleton-density score > 4 from Field *et al.* 2016 (average age of $19,007$ generations for our statistic vs $11,944$ generations for the SDS statistic using the combined mutation and recombination clock). We caution that the interpretation of these results is complicate by the considerable uncertainty in the allelic age estimates. Further, the timing of an episode of selection might post-date the age of the mutation – for example, when selection acts on standing variation. Finally, there is substantial variation in the mean age estimates of the hits. While the average age is around $19,000$ generations, 17 of the 42 putatively selected variants have ages less than $5,000$ generations. This suggests that the proposed statistic could be sensitive to both recent and older selection where the resulting allele frequencies are not well-modeled by the PCs.

To further illustrate how the ability to compute PCs in large samples is necessary for biological discovery, we analyzed how many selection signals we discover as a function of sample size by randomly subsampling the number of individuals from the White British population and repeating our analyses (Figure A.10). We ultimately find that sample sizes larger than $150,000$ individuals are required to retain over 80% of the total signals of selection we discover.

2.4 Discussion

We have presented, ProPCA, a scalable method for PCA on genotype data that relies on performing inference in a probabilistic model. Inference in this model consists of an iterative procedure that uses a fast matrix-vector multiplication algorithm. We have demonstrated its accuracy and efficiency across diverse settings. Further, we have demonstrated that ProPCA can accurately estimate population structure within the UK Biobank dataset and how this structure can be leveraged to identify targets of recent putative selection.

The algorithm that we employ here to accelerate the EM updates is of independent interest. Beyond PCA, several algorithms that operate on genotype data perform repeated matrix-vector multiplication on the matrix of genotypes. For example, association tests and permutation tests, can be formulated as computing a matrix-vector product where the matrix is the genotype matrix while the vector consists of phenotype measurements. Indeed, the algorithm has been used to accelerate heritability estimation [79]. The idea that SVD computations can leverage fast matrix-vector multiplication operations to obtain computational efficiency is well known in the numerical linear algebra literature [38]. Indeed, the algorithms [38, 54] implemented in other PCA methods can also utilize these ideas to gain additional computational efficiency. Alternate approaches to improve matrix-vector multiplication in the genetics setting include approaches that rely on sparsity of the genotype matrix. It is important to note that the speedup obtained from the Mailman algorithm does not rely explicitly on sparsity and could be applied even to dense matrices. It would be of interest to contrast the use of sparse multiplication versus the Mailman algorithm and to investigate the potential to combine these two approaches to be able to leverage sparsity as well as the discrete nature of the genotype matrix.

It is likely that different algorithms and implementations to compute PCs (and more generally, infer population structure) might be appropriate based on the specific application.

The choice of the specific algorithm and implementation involves a number of trade-offs. While ProPCA is computationally efficient, its use of the Mailman algorithm results in a bigger memory footprint relative to other methods. The probabilistic formulation underlying ProPCA allows the algorithm to be generalized in several directions. One direction is the application of PCA in the presence of missing data that often arises when analyzing multiple datasets. We have explored an extension of the ProPCA model to this setting (Appendix A.4, Figure A.11). While this approach is promising, a limitation of the use of the Mailman algorithm within ProPCA is the requirement of discrete genotypes, which prevents ProPCA from being directly applied to dosages (e.g. imputed genotypes). Another potential future direction is in modeling linkage disequilibrium and in incorporating rare variants which have the potential to reveal structure that is not apparent from the analysis of common SNPs [35, 36]. Current applications of PCA remove correlated SNPs and singletons though this has been shown to discard information [53]. One possible way to incorporate LD would leverage the connection between haplotype copying models [80] and the multivariate normal model of PCA [81], or by a whitening transformation [45]. Further, the observation model can also be modified to account for the discrete nature of genotypes [44, 82]. A number of non-linear dimensionality reduction methods have been recently proposed [39, 40]. A comparison of these methods to ProPCA (in terms of statistical structure that the methods aim to detect, ability to handle missing data, and computational scalability) would be of great interest. Finally, leveraging fine-scale population structure inferred from large-scale data to study recent positive selection in human history is an important direction for future work. While the probabilistic model underlying ProPCA leads to a natural model for testing for selection, other hypotheses about models of selection could lead to other tests of selection. The challenge is to design realistic statistical models of population structure while enabling inference at scale.

2.5 Software Availability

ProPCA is available at https://github.com/sriramlab/ProPCA.

CHAPTER 3

Inferring population structure in biobank-scale genomic data

3.1 Background

Inference of population structure is a central problem in human genetics with applications ranging from fine-grained understanding of human history [4] to correcting for population stratification in genome-wide association studies (GWAS) [58]. Approaches to population structure inference [49, 83, 25, 84, 85, 37] typically formalize the problem as one of estimating admixture proportions of each individual and ancestral population allele frequencies given genetic variation data.

The growth of repositories of genetic variation data over large numbers of individuals has opened up the possibility of inferring population structure at increasingly finer resolution [1, 3]. For instance, the UK Biobank [1] contains genotype data from approximately half a million British individuals across millions of SNPs. This development has necessitated methods that can be applied to large-scale datasets with reasonable runtime and memory requirements. Existing methods, however, do not scale to these datasets. Thus, we have developed SCOPE (SCalable pOPulation structure inferencE) – a scalable method capable of inferring population structure on biobank-scale data.

SCOPE utilizes a previously proposed likelihood-free framework [37] that involves estimation of the individual allele frequency (IAF) matrix through a statistical technique known as latent subspace estimation (LSE) [86] followed by a decomposition of the estimated IAF matrix into ancestral allele frequencies and admixture proportions. SCOPE uses two ideas to substantially improve the scalability of this approach. First, SCOPE uses randomized eigendecomposition [38] to efficiently estimate the latent subspace. Specifically, SCOPE avoids the need to form matrices that are expensive to compute on or require substantial memory instead working directly with the input genotype matrix. Second, SCOPE leverages the insight that the resulting method involves repeated multiplications of the genotype matrix and uses the Mailman algorithm for fast multiplication of the genotype matrix [34].

We benchmarked the accuracy and efficiency of SCOPE on simulated and real datasets. In simulations, SCOPE obtains accuracy comparable to existing methods while being up to 1,800 times faster. Relative to the previous state-of-the-art scalable method (TeraStructure [85]), SCOPE is 3 to 144 times faster. SCOPE can estimate population structure in about a day for a simulated dataset consisting of one million individuals and SNPs for six latent populations whereas TeraStructure, is extrapolated to require approximately 20 days on this same dataset. We additionally used SCOPE to infer continental ancestry proportions (four ancestry groups) on the UK Biobank dataset (488,363 individuals and 569,346 SNPs) in about a day. We find that the inferred continental ancestry proportions are highly concordant with self-reported race and ethnicity (SIRE).

SCOPE additionally can be applied in a supervised setting. Given allele frequencies from reference populations [87, 2, 66], SCOPE can estimate admixture proportions corresponding to the reference populations, to enable greater interpretability.

3.2 Methods

3.2.1 The Structure/Admixture Model

The structure/admixture model links the $m \times n$ genotype matrix $\boldsymbol{X}$ (where rows refer to single nucleotide polymorphisms (SNPs) and columns refer to individual diploid genotypes, $x_{ij} \in \{0, 1, 2\}, i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$) to the $m \times n$ individual allele frequency (IAF) matrix $\boldsymbol{F}$, $m \times k$ ancestral population allele frequencies $\boldsymbol{P}$, and the $k \times n$ individual admixture proportions $\boldsymbol{Q}$ (also termed the global ancestry of an individual). Here m denotes the number of SNPs, n denotes the number of individuals, and k denotes the number of latent populations. The IAF matrix, ancestral allele frequencies, and admixture proportions are mathematically related as $\boldsymbol{F} = \boldsymbol{P}\boldsymbol{Q}$. Furthermore, there are constraints on $\boldsymbol{P}$ and $\boldsymbol{Q}$. Each element of $\boldsymbol{P}$ is constrained to lie between 0 and 1 ($0 \leq p_{il} \leq 1, i \in \{1, \dots, m\}, l \in \{1, \dots, k\}$). Each element of $\boldsymbol{Q}$ is non-negative ($q_{lj} \geq 0, l \in \{1, \dots, k\}, j \in \{1, \dots, n\}$) and the admixture proportion of each individual must sum to one ($\sum_l q_{lj} = 1$). Finally, each entry of the genotype matrix is an independent draw from the corresponding entry of the IAF matrix $\boldsymbol{F}$ as: $x_{ij}|f_{ij} \sim \text{Binomial}(2, f_{ij})$. The goal of population structure inference under the structure/admixture model is to estimate $\boldsymbol{P}$ and $\boldsymbol{Q}$ given $\boldsymbol{X}$.

3.2.2 SCOPE

For scalable inference, SCOPE uses as its starting point a likelihood-free estimator of population structure previously proposed in ALStructure [37]. This estimator has two major steps: latent subspace estimation (LSE) and alternating least squares (ALS). LSE attempts to estimate the subspace spanned by the rows of $\boldsymbol{Q}$ [86] by computing a low-rank approximation to the matrix $\boldsymbol{G} = \frac{1}{m}\boldsymbol{X}^T\boldsymbol{X} - \boldsymbol{D}$ where each entry d_j of the $n \times n$ diagonal matrix $\boldsymbol{D}$ is obtained as $d_j = \frac{1}{m}\sum_{i=1}^{m} 2x_{ij} - x_{ij}^2$. The latent subspace of $\boldsymbol{Q}$ is estimated as the span of the top k eigenvectors of $\boldsymbol{G}$: $\boldsymbol{v}_1, \dots, \boldsymbol{v}_k$. After obtaining the top k eigenvectors $\boldsymbol{V} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_k]$, ALStructure projects the data $\boldsymbol{X}$ onto $\boldsymbol{V}$ to obtain an estimate of $\boldsymbol{F}$: $\hat{\boldsymbol{F}} = \frac{1}{2}\boldsymbol{X}\boldsymbol{V}\boldsymbol{V}^T$. Truncated alternating least squares (ALS) is used to factorize the estimate, $\hat{\boldsymbol{F}}$, into estimates of $\boldsymbol{P}$ and $\boldsymbol{Q}$: $\hat{\boldsymbol{F}} = \hat{\boldsymbol{P}}\hat{\boldsymbol{Q}}$. $\hat{\boldsymbol{Q}}$ are the estimates of the individual admixture proportions.

A naive approach to compute the top k eigenvectors of $\boldsymbol{G}$ would involve first forming the matrix $\boldsymbol{G}$ and then computing its top k eigenvectors which would require $\mathcal{O}(n^2m + n^2k)$ (if a full SVD is performed, this step would require $\mathcal{O}(\min(n, m)nm)$). To perform scalable LSE, SCOPE uses techniques from randomized linear algebra [38], specifically the implicitly restarted Arnoldi method [54], to obtain the top k eigenvectors. This step involves repeatedly multiplying estimates of the eigenvectors $\boldsymbol{v}_l : l \in \{1, \dots, k\}$ with the genotype matrix: $(\frac{1}{m}\boldsymbol{X}^T\boldsymbol{X} - \boldsymbol{D})\boldsymbol{v}_l = \frac{1}{m}((\boldsymbol{X}\boldsymbol{v}_l)^T\boldsymbol{X})^T - \boldsymbol{D}\boldsymbol{v}_l$ and can be performed without explicitly forming the matrix $\boldsymbol{G}$. Instead, this approach requires repeatedly computing $\boldsymbol{w}_l \equiv \boldsymbol{X}\boldsymbol{v}_l$, $\boldsymbol{w}_l^T\boldsymbol{X}$, and $\boldsymbol{D}\boldsymbol{v}_l$ which can be computed in $\mathcal{O}(nmk)$ time. We use the C++ Spectra library (https://spectralib.org/) to implement these computations in SCOPE.

To efficiently compute $\hat{\boldsymbol{P}}$ and $\hat{\boldsymbol{Q}}$ using truncated ALS, the matrix $\hat{\boldsymbol{P}}$ is initialized randomly with all values between 0 and 1 ($0 \leq \hat{p}_{il} \leq 1$). We iteratively solve for estimates of $\boldsymbol{P}$ and $\boldsymbol{Q}$, projecting the estimates onto the constraint space until convergence:

$$\hat{\boldsymbol{Q}} = \frac{1}{2}(\hat{\boldsymbol{P}}^{\boldsymbol{T}}\hat{\boldsymbol{P}})^{-1}\hat{\boldsymbol{P}}^T\boldsymbol{X}\boldsymbol{V}\boldsymbol{V}^T$$
$$\hat{\boldsymbol{P}} = \frac{1}{2}\boldsymbol{X}\boldsymbol{V}\boldsymbol{V}^{\boldsymbol{T}}\hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^T)^{-1}$$

All values in $\hat{\boldsymbol{P}}$ are truncated to be between 0 and 1 while $\hat{\boldsymbol{Q}}$ is projected onto the appropriate simplex. Each step of the ALS algorithm has runtime $\mathcal{O}(nmk)$. We note here that we never store $\hat{\mathbf{F}}$, but instead compute it implicitly per iteration. This allows us to reduce the memory footprint of SCOPE as $\hat{\mathbf{F}}$ is a continuous, real-valued matrix with the same dimensions as the genotype matrix. It is not feasible for most computers to be able to store this in memory. For instance, to store our larger UK Biobank dataset (488,363 individuals and 569,346 SNPs), one is estimated to require around 2,072 GB of memory.

Each of the computations in SCOPE require multiplying a genotype matrix with entries consisting of only 0, 1, and 2 for diploid genotype. These operations can be efficiently performed using the Mailman algorithm [34] that provides computational savings when there are repeated multiplications involving a matrix with a finite alphabet. We utilize the Mailman algorithm in computations involving the genotype matrix in both LSE and ALS so that the final time complexity of SCOPE is $\mathcal{O}(\frac{nmk}{\max(\log_3 n, \log_3 m)})$.

3.2.3 Supervised Population Structure Inference

SCOPE can utilize allele frequencies from reference populations to infer corresponding admixture proportions. In this scenario, we assume $\hat{\boldsymbol{P}}$, the population allele frequencies, are known. As a result, one only needs to compute $\hat{\boldsymbol{Q}}$ using the supplied $\hat{\boldsymbol{P}}$. This allows the admixture proportions corresponding to the reference populations to be inferred in a single step of ALS once the LSE step is completed.

3.2.4 Permutation Matching of Inferred Results

The output of population structure inference methods can result in output that is permuted even between different runs of the same method. It is critical to correctly match latent populations between methods and runs in order to properly assess results. To perform permutation matching, we employed a strategy similar to that of [88]. This permutation matching problem is better known as the Assignment Problem, which can be solved efficiently using linear programming. We first construct a score matrix using the distance metric created in [88]. The optimal permutation match can then be found by optimizing the total score from assignments through linear programming. We utilize the *lpSolve* (https://CRAN.R-project.org/package=lpSolve) package in R to solve the linear program.

3.2.5 PSD Model Simulations

We perform simulations under the Structure or Pritchard-Stephens-Donnelly (PSD) model [89]. In the PSD model, priors are placed on $\boldsymbol{P}$ and $\boldsymbol{Q}$:

$$\begin{aligned} p_{il} &\overset{iid}{\sim} \text{Beta}\left(\frac{1-F_{ST}}{F_{ST}}p_A, \frac{1-F_{ST}}{F_{ST}}(1-p_A)\right), i \in \{1,\ldots,m\}, l \in \{1,\ldots,k\} \\ \boldsymbol{q}_{:,j} &\overset{iid}{\sim} \text{Dirichlet}(\alpha \mathbf{1}_K), j \in \{1,\ldots,n\} \end{aligned}$$

The allele frequencies p_{il} are drawn from the Balding-Nichols model [90], which is a Beta distribution parametrized by the fixation index (F_{ST}) and an initial allele frequency (p_A). For our simulations, we calculated F_{ST} and p_A from our real datasets. Admixture proportions $\boldsymbol{q}_{:,j}$ are drawn at random from a Dirichlet distribution. We take the product of the two matrices to form the IAF matrix, $\boldsymbol{F} = \boldsymbol{PQ}$, and draw each genotype from a Binomial distribution parametrized by entries of $\boldsymbol{F}$: $x_{ij} \sim \text{Binomial}(2, f_{ij})$.

Spatial Model Simulations

We also perform simulations under a spatial model similar to that in [91]. In the spatial model, allele frequencies p_{il} are drawn as in the PSD model, but the admixture proportions, $\boldsymbol{q}$, are drawn from a 1D geography.

$$\begin{aligned}\boldsymbol{z} &\equiv (1, ..., k)\\ y_j &\overset{iid}{\sim} \text{Uniform}(0, k+1)\\ q_{lj} &= \frac{f_{z_l}(y_j)}{\sum_{l=1}^{k} f_{z_l}(y_j)}\end{aligned}$$

Populations are placed at integer values on a line. We get the resulting population position vector, $\boldsymbol{z} \equiv (1, ..., k)$. Each individual has a position, y_j drawn from a uniform distribution between 0 and $k+1$. Proportions for each population are generated by using a normal distribution, where f_{z_l} denotes the normal density function using z_l ($l \in \{1, \ldots, k\}$) as the mean and σ^2 as variance. The resulting vector of proportions is then normalized to satisfy the constraints on $\boldsymbol{Q}$. We used $\sigma^2 = 4$ for our simulations.

3.2.6 Assessment of Results

We assess our results using two metrics: average Jensen-Shannon divergence (JSD) and average root-mean-square error (RMSE). We calculate the metrics between the true global ancestry proportions, $\boldsymbol{Q}$ and the estimates, $\hat{\boldsymbol{Q}}$, after $\hat{\boldsymbol{Q}}$ has been permutation matched to the true proportions.

$$\begin{aligned}\text{RMSE}(\boldsymbol{Q}, \hat{\boldsymbol{Q}}) &= \frac{1}{\sqrt{nk}} ||\boldsymbol{Q} - \hat{\boldsymbol{Q}}||_F\\ \text{JSD}(\boldsymbol{Q}, \hat{\boldsymbol{Q}}) &= \frac{1}{2}\left[\text{KL}(\boldsymbol{Q}, \frac{1}{2}[\boldsymbol{Q} + \hat{\boldsymbol{Q}}]) + \text{KL}(\hat{\boldsymbol{Q}}, \frac{1}{2}[\boldsymbol{Q} + \hat{\boldsymbol{Q}}])\right]\end{aligned}$$

$||\cdot||_F$ represents the Frobenius norm. KL is the Kullback-Leibler divergence, which is defined as:

$$\text{KL}(\boldsymbol{Q}, \hat{\boldsymbol{Q}}) = \frac{1}{n} \sum_{j=1}^{n} \sum_{l=1}^{k} q_{lj} \log \left(\frac{q_{lj}}{\hat{q}_{lj}} \right)$$

In the JSD calculations, we replace values of 0 in $\boldsymbol{Q}$ or $\hat{\boldsymbol{Q}}$ with 1×10^{-9} to avoid numerical issues.

3.2.7 Datasets

We use the 1000 Genomes Project (TGP) [2, 87, 66], Human Origins (HO) [92], Human Genome Diversity Project (HGDP) [93, 94], and the UK Biobank (UKB) [1] in this study. The HGDP dataset is the complete Stanford HGDP SNP genotyping data filtered to only include individuals in the H952 set [95], greater than 95% genotyping rate, and greater than 1% minor allele frequency (MAF), resulting in 940 individuals and 642,951 SNPs. The TGP dataset is the 2012-01-31 Omni Platform genotypes filtered to only include unrelated individuals, greater than 95% genotyping rate, and greater than 1% MAF, resulting in 1,718 individuals and 1,854,622 SNPs. The HO dataset was filtered for human-only samples, greater than 99% genotyping rate, and greater than 5% MAF, resulting in 1,931 indiviuals and 385,089 SNPs. For the UK Biobank, we filtered the UK Biobank Axiom Array genotypes for greater than 1% MAF, long range linkage disequilibrium (LD), and pairwise LD pruning in 50 kilobase windows, 80 variant step size, and an r^2 threshold of 0.1, resulting in 488,363 individuals and 568,346 SNPs. This is similar to the UK Biobank manuscript's first round of quality control for principal component analysis [1] with the differences of using all individuals and no genotype filter. We also use the UK Biobank's final set of PCA SNPs [1], which consists of 147,604 SNPs, to explore higher number of latent populations. We calculate metrics such as F_{ST} from the provided population and superpopulation labels provided by each dataset. To perform our supervised analyses, we use the common SNPs between the datasets involved. All genotype processing was performed using PLINK [57]. Links to the publicly available datasets as well as scripts to apply our preprocessing are available in code repository for SCOPE.

3.2.8 Visualization of Results

We visualize our inferred admixture proportions as stacked bar plots. Estimates from all methods were permutation matched to enable easy comparison. For our PSD simulations, we performed hierarchical clustering with complete linkage on a Euclidean distance matrix calculated from the true admixture proportion matrix ($\boldsymbol{Q}$) to obtain the order of samples. For our spatial simulations, we sorted by decreasing membership of the first population. For our real datasets, we perform the same hierarchical clustering strategy used for our PSD simulations, but use the estimates from ADMIXTURE ($\hat{\boldsymbol{Q}}$) in place of the true admixture proportions. For the HGDP, TGP, and UK Biobank, we first took the average proportions for each SIRE group and performed hierarchical clustering on the averages to determine the order of the SIRE groups. We then performed hierarchical clustering within each SIRE group to determine the order of individuals within groups. For large datasets, we utilized *genieclust* [96], a scalable method for hierarchical clustering.

3.2.9 Benchmarking

We compared SCOPE to ADMIXTURE v1.3.0 [25], fastSTRUCTURE [84], TeraStructure [85], and ALStructure v0.1.0 [37], and sNMF v1.2 [97].

ADMIXTURE computes maximum-likelihood estimates while TeraStructure and fastSTRUCTURE compute approximate posterior estimates in a Bayesian model using variational inference. ALStructure, the framework which SCOPE builds upon, utilizes a two-stage strategy of first performing dimensionality reduction (latent subspace estimation) followed by matrix factorization (alternating least squares).

Each method was run with 8 threads with the exception of fastSTRUCTURE and ALStructure, which do not have multi-threaded implementations. Default parameters were used. TeraStructure has an additional ’rfreq’ parameter, which was set to 10% of the number of SNPs as recommended by its authors. For SCOPE, we used convergence criteria of either 1,000 iterations of the ALS algorithm or a change between iterations less than 1×10^{-5}, which we calculate as the RMSE between the estimated admixture matrices between two iterations. All experiments were performed on a server with two AMD EPYC 7501 32-Core Processors and 1 terabyte of RAM.

3.3 Results

3.3.1 Accuracy

We assessed the accuracy of SCOPE using simulations under the Pritchard-Stephens-Donnelly (PSD) model [49] to study accuracy under a standard population genetics model and a basic model of spatial structure [91] to study the robustness of SCOPE and other methods in the presence of model violations. We simulated several independent datasets using parameters calculated from two real datasets: the 1000 Genomes Project (TGP) [2] and the Human Genome Diversity Project (HGDP) [98] (benchmarking sections of Methods). It is important to note that each simulation dataset was created independently of the others and are not subsets of the largest dataset. Thus, performance should only be compared between methods run on the same dataset.

Under the PSD model, which matches the assumptions of the methods tested, ADMIXTURE is the most accurate followed by SCOPE and ALStructure (Figures 3.1, B.1, B.2, B.3, B.4). Among the scalable methods, TeraStructure and SCOPE, SCOPE tends to be more accurate in terms of both Jensen-Shannon divergence (JSD) (Table 3.1) and root-mean-square error (RMSE) (Table 3.2). We also assessed accuracy under a spatial model, which violates the assumptions of the PSD model by inducing a spatial relationship between the admixture proportions (Figures 3.2, B.5, B.6, B.7). Under this scenario, SCOPE, ALStructure, and sNMF are typically the most accurate (Tables 3.1, 3.2).

We also observe similar trends when calculating Kullback-Leibler (KL) divergence (Tables B.1, B.2), but opt to use Jensen-Shannon divergence as a primary accuracy measurement

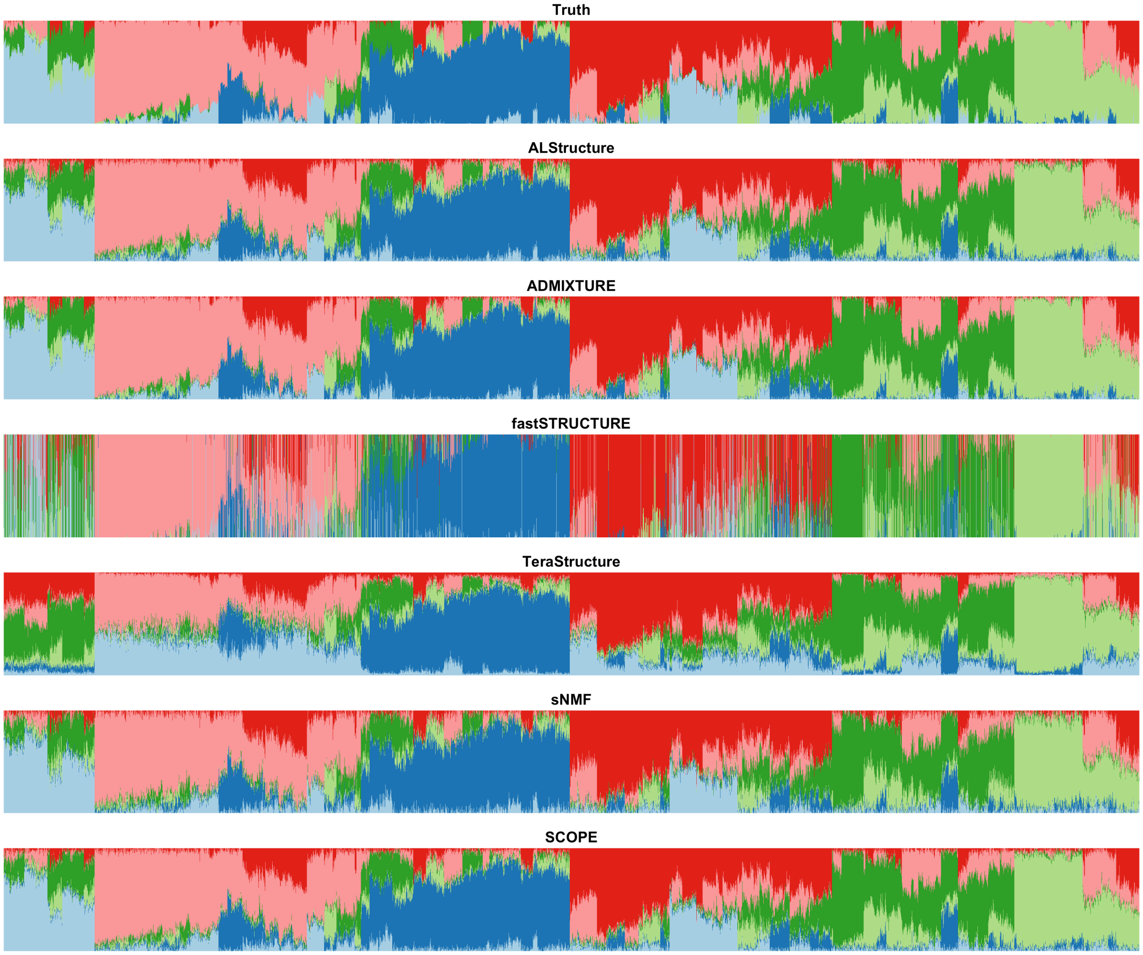


Figure 3.1: **Population structure inference for simulations under PSD model generated using 1000 Genomes Phase 3 data.** PSD model parameters were drawn from TGP data to generate a simulation dataset with 10,000 samples and 10,000 SNPs. The true admixture proportions and resulting inferred admixture proportions from each method are shown. Colors and order of samples are matched between each method to the truth.

Table 3.1: **Jensen-Shannon divergence measurements for methods on simulated data.** Jensen-Shannon divergence (JSD) was computed against the ground truth admixture proportions for each simulation. Values are displayed as percentages rounded to one decimal place. Estimated proportions of 0 were set to 1×10^{-9} (see Methods). A '-' denotes that the method was not run due to projected time or memory usage. Bold values denote the best value for each dataset.

Dataset Type	Base Dataset	k	n	m	ADMIXTURE	fastStructure	TeraStructure	ALStructure	sNMF	SCOPE
PSD	HGDP	6	10,000	10,000	**2.4**	6.3	13.7	3.6	**2.4**	3.6
PSD	TGP	6	10,000	10,000	**0.8**	11.3	8.8	1.9	2.4	1.9
PSD	TGP	6	10,000	1,000,000	**0.03**	8.1	0.2	-	-	0.2
PSD	TGP	6	100,000	1,000,000	-	-	0.3	-	-	**0.2**
PSD	TGP	6	1,000,000	1,000,000	-	-	-	-	-	**0.2**
Spatial	HGDP	6	10,000	10,000	6.5	33.9	5.7	**2.1**	2.3	2.6
Spatial	TGP	6	10,000	10,000	6.8	31.1	3.4	**2.4**	4.0	3.3
Spatial	TGP	10	10,000	100,000	12.4	34.7	6.3	8.1	5.7	**5.6**
Spatial	TGP	10	10,000	1,000,000	-	-	10.0	-	-	**8.2**

Table 3.2: **Root-mean-square error measurements for methods on simulated data.** Root-mean-square error (RMSE) was computed against the ground truth admixture proportions for each simulation. RMSE is displayed in percentage and rounded to the first decimal place. A '-' denotes that the method was not run due to projected time or memory usage. Bold values denote the best value for each dataset.

Dataset Type	Base Dataset	k	n	m	ADMIXTURE	fastStructure	TeraStructure	ALStructure	sNMF	SCOPE
PSD	HGDP	6	10,000	10,000	**4.0**	10.3	16.6	5.6	4.1	5.6
PSD	TGP	6	10,000	10,000	**1.8**	15.9	13.7	3.2	4.1	3.2
PSD	TGP	6	10,000	1,000,000	**0.2**	12.4	0.9	-	-	0.3
PSD	TGP	6	100,000	1,000,000	-	-	1.0	-	-	**0.4**
PSD	TGP	6	1,000,000	1,000,000	-	-	-	-	-	**0.5**
Spatial	HGDP	6	10,000	10,000	11.9	31.1	10.2	**5.7**	**5.7**	6.5
Spatial	TGP	6	10,000	10,000	12.5	29.1	**6.8**	7.5	9.4	7.3
Spatial	TGP	10	10,000	100,000	10.8	22.8	8.8	8.5	**6.7**	**6.7**
Spatial	TGP	10	10,000	1,000,000	-	-	**6.6**	-	-	7.2

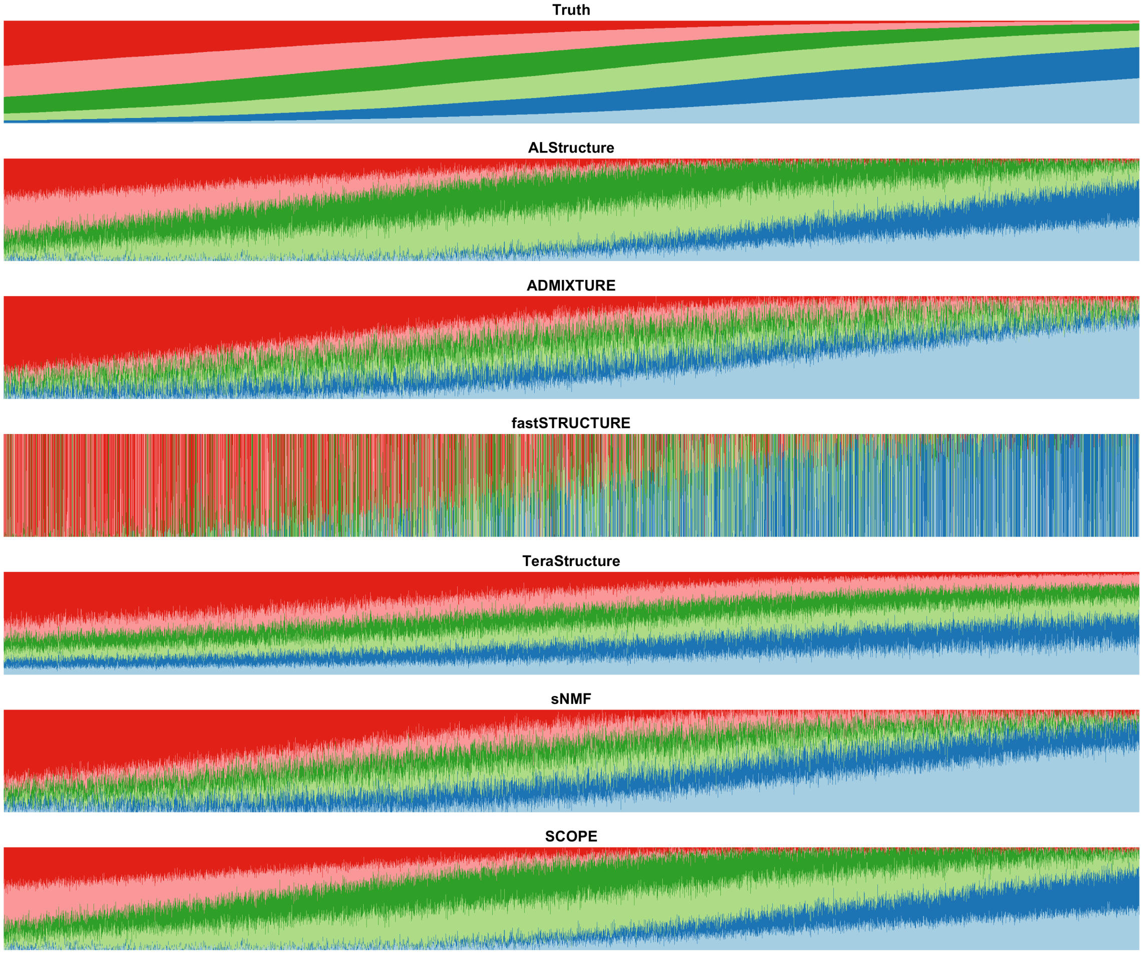


Figure 3.2: **Population structure inference for simulations under a spatial model generated using 1000 Genomes Phase 3 data.** Model parameters were drawn from TGP data to generate a simulation dataset with 10,000 samples and 10,000 SNPs under a spatial model (see Methods). The true admixture proportions and resulting inferred admixture proportions from each method are shown. Colors and order of samples are matched between each method to the truth.

due to the asymmetric nature of KL divergence, which changes depending on the order of inputs. We also assessed whether SCOPE can consistently arrive at similar solutions across runs regardless of the stochastic approximations used in SCOPE's algorithm. We ran five replicates of SCOPE from 2-40 inferred populations on a HGDP PSD simulation (Figure B.8a), TGP PSD simulation (Figure B.8b), HGDP dataset (Figure B.8c), and HO dataset (Figure B.8d). We observe in our simulated datasets that SCOPE in consistent across both JSD and RMSE between solutions up to the simulated number of populations. Both accuracy measures decrease when inferred more populations than simulated. For the HGDP and HO datasets, we observer that SCOPE is mostly consistent even up to 40 inferred populations. On ocassion, we see slight inconsistency, but this is largely due to one replicate differing from the other (Figure B.9).

3.3.2 Runtime and memory

Using simulated and real datasets, we compared the runtime of SCOPE to ADMIXTURE, fastStructure, TeraStructure, sNMF, and ALStructure (Table 3.3). Not all of the compared methods could be run on all datasets within practical constraints of time and memory.

On the largest PSD datasets that each method could be run on, SCOPE is over 150 times faster than ADMIXTURE (10,000 individuals by 1 million SNPs), over 500 times faster than fastStructure (10,000 individuals by 1 million SNPs), about 100 times faster than ALStructure (10,000 individuals by 100,000 SNPs), over 110 times faster than TeraStructure (100,000 individuals by 1 million SNPs), and as fast as sNMF (10,000 individuals by 10,000 SNPs). SCOPE is also capable of running on a dataset containing one million SNPs and individuals in just over 24 hours (≈ 1 day) whereas TeraStructure is extrapolated to require about 500 hours (≈ 20 days) based on times reported in its manuscript [85] as well as our experiments (see benchmarking sections of Methods).

The runtime of all methods increases under the spatial model. In this scenario, SCOPE is over 1800 times faster than ADMIXTURE (10,000 individuals by 100,000 SNPs), about 210 times faster than fastStructure (10,000 individuals by 100,000 SNPs), over 155 times

Table 3.3: **Runtimes and fold-speedups of methods on simulations and real datasets.** ADMIXTURE, TeraStructure, sNMF, and SCOPE were run using 8 threads. ALStructure and fastStructure were run on a single thread due to their lack of multithreading implementations. Default parameters were used. TeraStructure's '-rfreq' parameter was set to 10% of the number of SNPs. Times are rounded to the nearest minute and displayed in hours:minutes. The fold-speedup (runtime of method in seconds divided by runtime of SCOPE in seconds) achieved by SCOPE is denoted beneath each time in parentheses and rounded to the nearest integer. Bold values denote the best value for each dataset. Runtimes for SCOPE under one minute are denoted as "< 1 min." A '-' denotes that the method was not run due to projected time or memory usage.

Dataset Type	Base Dataset	k	n	m	ADMIXTURE	fastStructure	TeraStructure	ALStructure	sNMF	SCOPE
PSD	HGDP	6	10,000	10,000	0:14 (48)	3:44 (746)	0:11 (36)	0:30 (101)	< **1 min** (1)	< **1 min**
PSD	TGP	6	10,000	10,000	0:17 (206)	1:22 (987)	0:12 (144)	0:23 (271)	< **1 min** (1)	< **1 min**
PSD	TGP	6	10,000	1,000,000	35:12 (156)	114:51 (509)	20:31 (91)	-	-	**0:14**
PSD	TGP	6	100,000	1,000,000	-	-	237:02 (113)	-	-	**2:06**
PSD	TGP	6	1,000,000	1,000,000	-	-	-	-	-	**24:37**
Spatial	HGDP	6	10,000	10,000	5:52 (440)	4:06 (308)	0:03 (3)	1:39 (124)	< **1 min** (1)	< **1 min**
Spatial	TGP	6	10,000	10,000	3:11 (239)	3:19 (249)	0:07 (9)	1:55 (144)	∼ **1 min** (1)	< **1 min**
Spatial	TGP	10	10,000	100,000	284:47 (1808)	33:03 (210)	4:29 (28)	24:51 (158)	0:33 (4)	**0:09**
Spatial	TGP	10	10,000	1,000,000	-	-	15:22 (9)	-	-	**1:47**
Real	HGDP	10	940	642,951	4:24 (31)	4:39 (33)	0:40 (5)	0:55 (7)	0:16 (2)	**0:08**
Real	HO	14	1,931	385,089	13:28 (122)	24:49 (224)	1:37 (15)	2:11 (20)	0:30 (4)	**0:07**
Real	TGP	8	1,718	1,854,622	31:33 (33)	8:53 (9)	4:20 (5)	11:16 (12)	-	**0:57**
Real	UKB	4	488,363	569,346	-	-	-	-	-	**25:57**
Real	UKB	20	488,363	147,604	-	-	-	-	-	**23:42**
Real	UKB	40	488,363	147,604	-	-	-	-	-	**51:25**

faster than ALStructure (10,000 individuals by 100,000 SNPs), about 9 times faster than TeraStructure (10,000 individuals by 1 million SNPs), and 4 times faster than sNMF (10,000 individuals by 100,000 SNPs) on the largest dataset each method could be run on. Over all of the datasets, SCOPE is up to 1800 times faster than existing methods and three to 144 times faster than TeraStructure. Furthermore, SCOPE scales linearly with the number of latent populations inferred (Figure B.10). Additional threads can also be used by SCOPE to speed-up runtime up until a fundamental I/O bound is reached (Figure B.11).

SCOPE has a reasonable memory footprint: for large datasets for which only TeraStructure and SCOPE were feasible, SCOPE uses slightly less memory than TeraStructure with the memory usage of SCOPE scaling linearly in the size of genotype matrix (*i.e.* the number of individuals times the number of SNPs) (Table B.3). SCOPE requires less than 250 GB for the UK Biobank dataset (488,363 individuals and 569,346 SNPs) and 750 GB for the dataset consisting of one million individuals and SNPs. When using smaller SNP sets such as the UK Biobank's PCA set (147,604 SNPs), SCOPE uses about 60 GB of memory (488,363 individuals and 147,604 SNPs).

3.3.3 Accuracy of supervised analysis

Out of the methods tested, only SCOPE and ADMIXTURE are able to use supplied allele frequencies to perform population structure inference in a supervised fashion (Tables 3.4, B.4). In the PSD model simulations, we observe a small improvement to both RMSE and JSD relative to unsupervised population structure inference (Figures B.12, B.13, B.14, B.15, B.16). Under the spatial model simulations, the use of supervision obtains much greater accuracy compared to unsupervised inference (Figures 3.3, B.17, B.18, B.19).

3.3.4 Application to real genotype data

We applied SCOPE to several real, genomic datasets: TGP (1,718 indviduals and 1,184,622 SNPs) with 8 latent populations ($k = 8$) (Figure B.20), HGDP (940 indviduals and 642,951 SNPs) with 10 populations ($k = 10$) (Figure B.21), Human Origins (HO) (1,931 indviduals

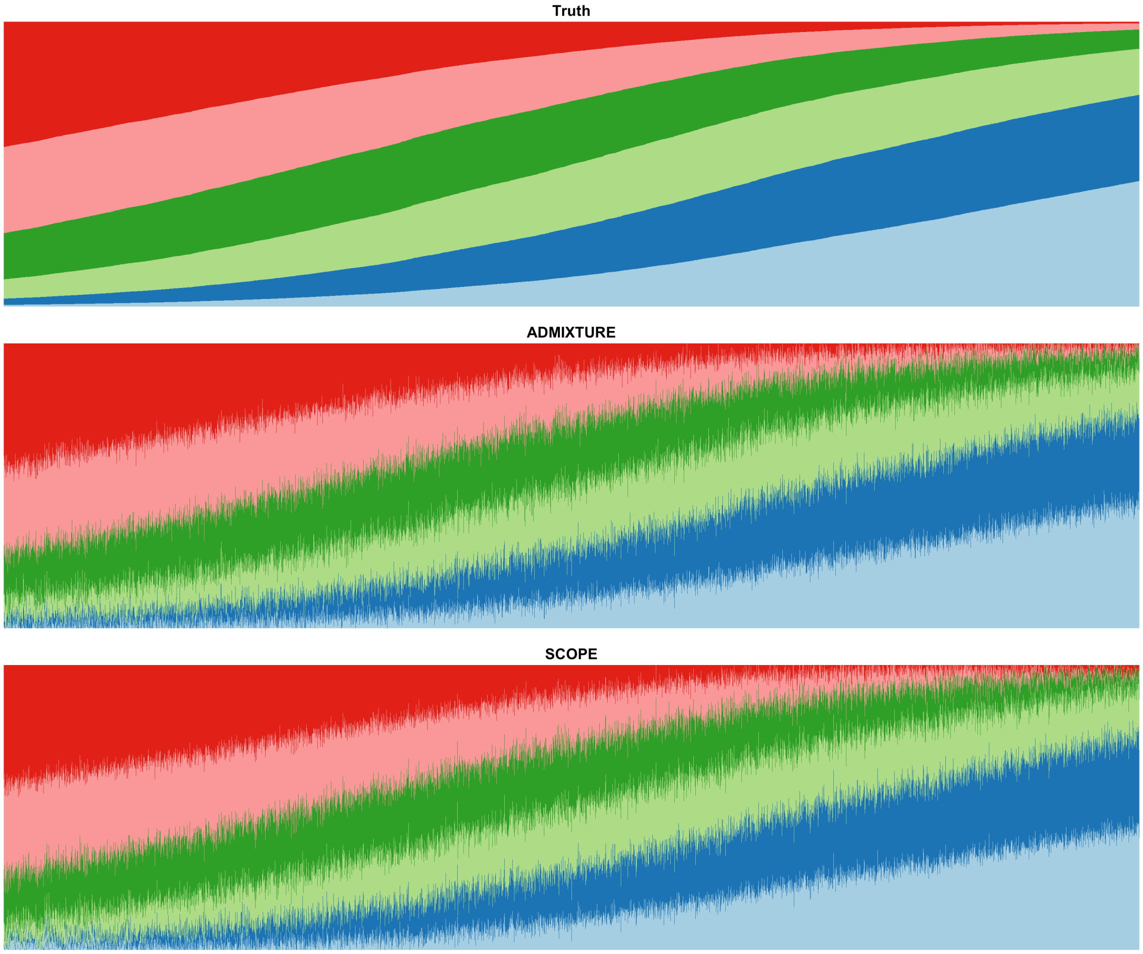


Figure 3.3: **Supervised population structure inference for simulations under a spatial model generated using 1000 Genomes Phase 3 data.** Model parameters were drawn from TGP data to generate a simulation dataset with 10,000 samples and 10,000 SNPs under a spatial model. Both methods were provided the true population allele frequencies as input. Colors and order of samples are matched between each method to the truth.

Table 3.4: **Accuracy of supervised population structure inference using supplied allele frequencies on simulations.** True allele frequencies were supplied to SCOPE to use in supervised population structure inference. Root-mean-square error (RMSE) and Jensen-Shannon Divergence (JSD) were computed against the true admixture proportions. Estimated proportions of 0 were set to 1×10^{-9} for JSD calculations (see Methods). Values are displayed in percentages and rounded to the first decimal place. Bold values denote the best value for each dataset.

					Supervised		Unsupervised	
Dataset Type	Base Dataset	k	n	m	RMSE	JSD	RMSE	JSD
PSD	HGDP	6	10,000	10,000	**2.9**	**1.5**	5.6	3.6
PSD	TGP	6	10,000	10,000	**2.0**	**0.9**	3.2	1.9
PSD	TGP	6	10,000	1,000,000	**0.2**	**0.1**	0.3	0.2
PSD	TGP	6	100,000	1,000,000	**0.2**	**0.1**	0.4	0.2
PSD	TGP	6	1,000,000	1,000,000	**0.2**	**0.1**	0.5	0.2
Spatial	HGDP	6	10,000	10,000	**2.4**	**0.6**	6.5	2.6
Spatial	TGP	6	10,000	10,000	**1.7**	**0.3**	7.3	3.3
Spatial	TGP	10	10,000	100,000	**0.6**	**0.3**	6.7	5.6
Spatial	TGP	10	10,000	1,000,000	**0.3**	**0.1**	8.2	7.2

and 385,089 SNPs) [92] with 14 populations ($k = 14$) (Figure B.22), the UK Biobank (488,363 indviduals and 569,346 SNPs) with 4 populations ($k = 4$) (Figure 3.4), and the UK Biobank (488,363 indviduals and 147,604 SNPs) with 20 populations ($k = 20$) (Figure B.24) and 40 populations ($k = 40$) (Figure B.25) (see Methods for quality control). We chose the number of latent populations to be consistent with previous studies on these datasets [37, 85]. For the UK Biobank analysis, we chose four latent populations to infer continental ancestry groups for the larger SNP set and 20 and 40 latent populations to explore SCOPE's ability to infer larger number of latent populations on real data. In terms of runtime and memory, we continued to observe trends consistent with our simulations where SCOPE is orders of magnitude faster than other methods while consuming reasonable amounts of memory (Tables 3.3, B.3). We note that the runtime for inference on the larger UK Biobank dataset is about the same as the runtime for our 1 million individual and SNP simulation despite the UK Biobank being approximately a quarter of its size, consistent with the increase in runtimes with model deviations as seen in the context of spatial simulations.

Since there is no ground truth to assess accuracy on these datasets, we used concordance between SIRE and inferred admixture proportions as a metric. We trained multinomial logistic regression models to predict continental ancestry for the TGP (5 populations) and HGDP (7 populations) using the inferred admixture proportions from each method (Table B.5). We find that all methods perform similarly on both datasets. For the UK Biobank, SCOPE is able to obtain 88.27% accuracy when using labels provided by UK Biobank (22 labels) and 95.75% accuracy when ambiguous/heterogeneous labels (*e.g.* Other, Mixed) are removed and population labels are collapsed to continental groupings (8 labels). We did not perform this analysis for the HO dataset due to several population labels only containing one sample.

We additionally assessed SCOPE's ability to infer finer population structure using the British individuals in the UK Biobank. We trained ordinary least squares models to predict the self-reported birth location GPS coordinate using the inferred proportions from the different runs of SCOPE under different numbers of latent populations (4, 20, and 40 latent populations) (Table B.6). Increasing the number of latent populations generally improves the prediction accuracy when measured through coefficient of determination (R^2). With four latent populations, the R^2 is 0.007 and 0.008 for latitude and longitude prediction, respectively. This increases to 0.2-0.3 and approximately 0.15 when increasing the number of latent populations to 20 and 40. We also examined the prediction accuracy in terms of residual distance (difference between predicted and reported location). The 95% quantile for the residual distances decreases from ≈334 kilometers to ≈290 kilometers when increasing the number of inferred populations from 4 to 20 or 40.

We also utilized the supervised mode of SCOPE using known population allele frequencies from TGP superpopulations to infer continental ancestry for all individuals in the UK Biobank. We find that the supervised mode of SCOPE largely agreed with the unsupervised inference (Figures 3.4, B.23).

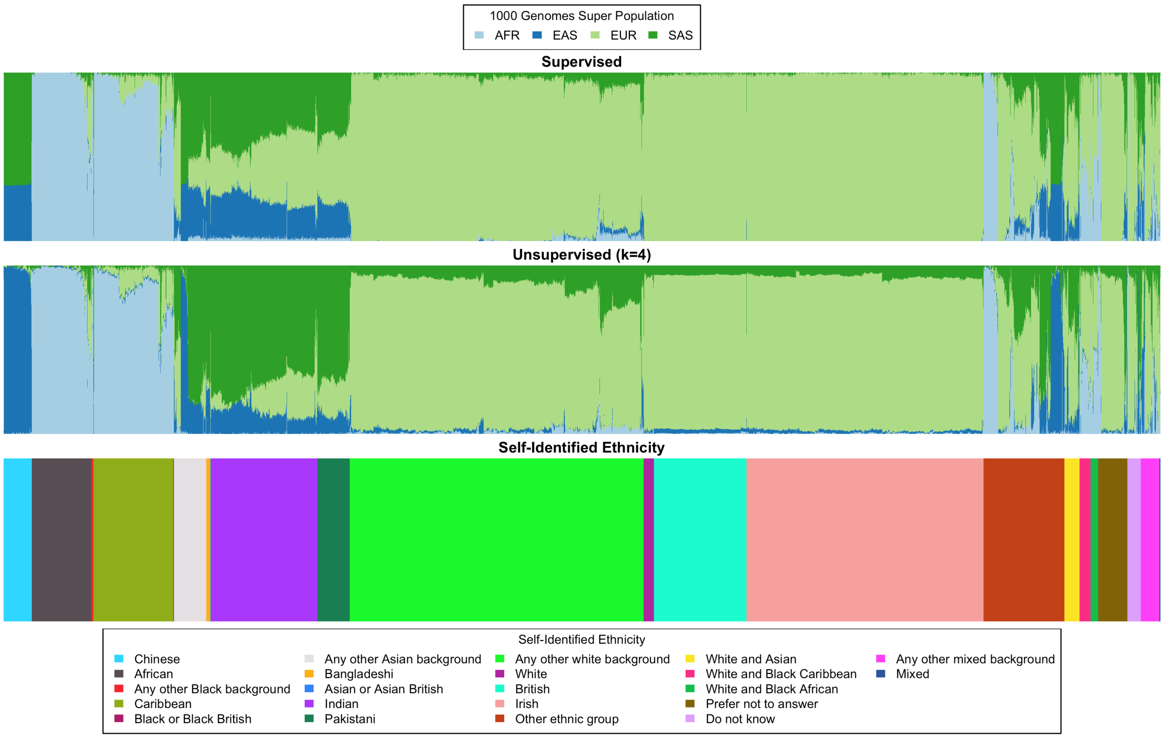


Figure 3.4: **Continental ancestry inference on the UK Biobank.** We ran population structure inference using SCOPE on the UK Biobank (488,363 individuals and 569,346 SNPs) both supervised using 1000 Genomes Phase 3 allele frequencies (top) and unsupervised with 4 latent populations (middle). For reference, we plot the self-identified race/ethnicity (bottom). For visualization purposes, we reduced the number of self-identified British individuals to a random subset of $5,000$ individuals. Colors and order of samples are matched between each row of the figure. The full figure without individuals removed can be found in Figure B.23.

3.4 Discussion

We have presented SCOPE, a scalable method for inferring population structure from biobank-scale genomic data. We show that SCOPE remains accurate while being scalable in terms of runtime and memory requirements. SCOPE is also able to perform supervised analyses that leverage allele frequency estimates from previous studies to improve interpretability, runtime, and accuracy.

SCOPE enables new analyses by improving the scalability of admixture proportion inference. The inclusion of more individuals and/or genomic sites allows more rare latent population structure to be discovered in addition to improving estimation of the true latent population frequencies. These are often the cases where scaling to biobank-level data becomes a necessity. Furthermore, many admixture tools are often used as an exploratory analysis being run with different numbers of latent populations (*i.e.* k). Being able to perform several runs quickly becomes important for initial analysis.

The use of SCOPE is not without limitations for real data analysis and interpretation. For instance, while larger non-trivial numbers of latent populations (k) such as 20 (Figure B.24) and 40 (Figure B.25) from the UK Biobank explored in this study increase our ability to dissect finer scale population structure, they remain very difficult to interpret. Furthermore, when exploring these settings, care must be taken to curate a well-defined SNP set. For example, we see a decrease in prediction accuracy when moving from 20 to 40 latent populations in the UK Biobank. This may be attributed to the fact that the UK Biobank's PCA SNP set was curated to differentiate continental population structure rather than intracontinental structure. We also observed that SCOPE is consistent when inferring a large number of latent populations as exemplified by our replicate studies on the HGDP (Figure B.8c) and HO (Figure B.8d) datasets, suggesting there is more fine-scale population structure being detected and opens the question of what these latent populations may correspond to. While the ability to use supervised analysis as we did for the UK Biobank can greatly improve interpretability, supervision with SCOPE largely depends on the accuracy of the reference dataset and frequencies used. Finally, there is still the open question of choosing the appropriate number of latent populations (k). While SCOPE allows one to run several different values for k, we do not provide any criteria to choose a specific value of k. We defer deeper analysis of these questions for future studies.

The methodology used in SCOPE can also be extended in several ways. Several methods that perform structure inference on other genomic datasets [99, 100] utilize semi-supervised approaches where there are both known and unknown populations. A possible approach for semi-supervision using SCOPE is to perform a multi-stage inference procedure where supervised inference is first applied and unsupervised inference is applied on the residual or unexplained structure. Most current methods, including SCOPE, ignore additional information within the data such as correlation patterns (*i.e.* linkage disequilibrium or LD). Some methods such as fineSTRUCTURE [101] can perform linkage-disequilibrium aware population structure inference but are challenging to scale. Methods that can model LD while retaining scalability is a key step in advancing population structure inference.

Though not directly related to the admixture model, there are several approaches to finding broader forms of structure that are not explicitly in the form of admixture proportions. For instance, possible usage of non-linear dimensionality reduction techniques such as UMAP[102] could provide promising ways to extend beyond current methods, which solely utilize linear methods such as PCA. Other approaches to detecting fine-scale structure include using identity-by-descent (IBD) [103] or tree-based methods [104]. Finding ways to scalably bridge these different approaches with the admixture model is still an open question. Finally, extensions of the techniques used in SCOPE can be used to infer relevant structure in other domains such as metagenomics and single-cell transcriptomics.

3.5 Data and Code Availability

SCOPE can be found at https://github.com/sriramlab/SCOPE. Scripts for simulations, visualization, assessment, downloading of pubicly available data, and real data filtering, and additional code used in this study can be found at the repository as well. UK Biobank data is the only dataset used in this study that is not publicly available, but can be obtained by application (https://www.ukbiobank.ac.uk/).

CHAPTER 4

A simple statistical testing framework for detecting differences in variance and covariance in gene expression networks

4.1 Background

A major goal in molecular biology is the characterization of gene expression and how it is altered in response to perturbations. Many of the studies involving gene expression alterations tend to focus on changes in mean expression, but there is increasing evidence that changes in variability [105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115] and gene co-expression [116, 117, 118, 119, 120, 121] play important roles in biological processes.

Previous studies suggest that expression variability plays an key role in incomplete penetrance [105, 108]. It has also been implicated in several developmental phenotypes such as embyro development [106] and cell fate [105]. Several diseases also exhibit changes in expression variance such as schizophrenia [109, 110, 111] and leukemia [112]. Furthermore, expression variability is suggested to have roles in evolution [113, 114] and adaptation to perturbations [122] such as drug response [115]. Meanwhile, much more studies have focused on gene co-expression changes between conditions, particularly in the context of disease such as cancer [118, 119] and brain disorders [120, 121]. These studies often attribute characteristics of the phenotype of interest to disruptions in the functions of gene networks and interactions between the genes inside them.

There are several available tools for performing differential analysis on mean gene expression [123, 124], gene variability [125], and gene co-expression [117, 126]. However, these tools focus on detecting such differences between individual genes or pairs of genes between conditions.

As opposed to testing individual genes or gene pairs, we present a simple statistical testing framework to detect perturbations in variance or co-expression/covariance within sets of genes, as a whole, across conditions. Several important gene sets can be derived such as gene networks [41]. Our approach utilizes the fact that the variance of a gene's expression and covariance between two genes' expression can be estimated by simply squaring each feature or taking the product of the two features, respectively, after normalizing and demeaning each gene. Principal component analysis (PCA) can then be used on the resulting matrices to summarize the network's gene (co)variance. The resulting projections or principal component scores can then be used as in hypothesis testing to statistically test for differences in a network between two conditions in terms of variance or covariance.

This work extends the concept of eigengenes [127] from the context of gene expression. Eigengene analysis primarily focuses on the module eigengene, which is defined as defined as the first principal component of the gene expression matrix for a module/network. For simplicity, we will use the term 'eigengene' to specifically refer to the module eigengene. Several characteristics of the network can be quantified using eigengene analysis [128]. One specific application of eigengene analysis is to perform differential expression analysis by performing statistical tests (*e.g.* t-test) between groups using the eigengene values [129]. Our proposed testing framework utilizes a similar strategy of performing PCA, but instead decomposes transformed versions of the gene expression matrix to specifically identify changes in variance and covariance. As a result, our testing framework is analogous to differential expression analysis using eigengenes, but instead, tests for differences in variance or covariance.

We apply our testing framework to simulations under a variety of combinations of different mean, variance, and/or covariance changes in gene expression to assess the power and calibration of our testing framework. Under these simulations, we show that this statistical framework is well-powered for differences in variance or covariance and exclusively detects variance or covariance differences despite the presence of related effects (*e.g.* mean effects). In addition to simulations, we generate positive and negative controls from previously developed differential variance [125] for individual genes and differential covariance methods [126, 117] for pairs of genes to verify that our framework extends to real data.

We apply our testing framework to uncover mean, variance, and covariance differences in several previously discovered gene networks derived from weighted gene co-expression analysis (WGCNA) [41] for psychiatric diseases. In our analyses, we find instances of several combinations of mean, variance, and covariance differences including scenarios with no mean differences, but difference in variance and covariance, further highlighting the importance of looking beyond mean differences. We additionally apply our method to breast cancer methylation data from The Cancer Genome Atlas (TCGA) [5] to uncover variance differences related to vital status. Lastly, we apply our testing framework to stock market data to reveal variance and covariance differences between presidencies. The application to methylation data and stock market data exemplifies our testing framework's ability to generalize to several data types, including non-omic data.

4.2 Methods

4.2.1 Statistical testing framework

Given two groups, A and B, and an $n \times m$ (individuals by genes) normalized gene expression matrix for a network, G, one can generate two matrices that correspond to individual estimates of variance or covariance. We wish to detect whether there are differences in G between conditions, A and B, in terms of variance or covariance. We propose the following statistical testing procedure. Let g_i be the gene expression for a gene/column in G. Let $\bar{g}_{i,A}$ and $\bar{g}_{i,B}$ be the mean gene expression for gene i in groups A and B, respectively. We can create a demeaned gene expression vector $\tilde{g}_i$ by demeaning by the appropriate group mean. We repeat this procedure for every gene $i \in \{1, ..., m\}$ to create $\tilde{G}$, a centered gene expression matrix that has been demeaned by group.

The $n \times m$ squared matrix used to test for variance differences be generated by simply squaring every term in $\tilde{G}$. The product matrix can be created by taking pairwise products for every gene in $\tilde{G}$, resulting in a $n \times \frac{m(m-1)}{2}$ matrix of pairwise products. PCA is then performed on the newly generated matrix. In the case of multiple groups, each group is demeaned prior to apply the transformations.

The principal components scores or projections of the first principal component (eigengene) can then be used as a outcome variables for a linear model while the input variable consists of the group membership. A simple Wald test can then be performed on the coefficient of this linear model to determine whether there are differences in the network. To test for variance differences, one constructs the linear model using the projections of the PCA performed on the squared matrix while covariance differences can be detected by constructing the linear model using the projections from the PCA performed on the product matrix.

PCA is performed using the implicitly-restarted Lanczos method implemented in the *irlba* package in R [130]. This algorithm allows us to scalably obtain the top principal components on large matrices such as the product matrix. We specifically use the *irlba* function to perform singular value decomposition to perform the PCA.

4.2.2 Simulations

We performed 1,000 simulations with different seeds for each simulation. We additionally used the R package *mvtnorm* [131] to simulate from a multivariate normal distribution.

4.2.2.1 Simulating mean effects

We simulated mean effects by generating two groups from multivariate normal distributions with the identity matrix for the covariance matrix. To simulate a difference in means, we simulated one group with the zero vector for the mean while the other group had a constant vector with μ for all entries, where μ is the magnitude of the mean difference.

4.2.2.2 Simulating variance effects

To simulate variance differences, we generated two groups with zero vectors for the mean vectors and a diagonal matrix for the covariance matrix. For one group, we drew m random numbers from a random uniform distribution between 0 and 5, where m is the number of features/genes. For the second group, we added a constant shift σ^2 to the diagonal of the first group. Samples were drawn from each group, accordingly.

4.2.2.3 Simulating covariance effects

We simulated covariance differences by using a Cholesky decomposition followed by a Kronecker product. We first generate two bivariate correlation matrices, $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2$ that have 1 on the diagonals. One group has the off-diagonal or correlation values set to -0.99 while the other group has a constant shift added to the correlation. Using bivariate correlation matrices allows us to simulate pairs of correlated genes while maintaining independence from other pairs.

First, the Cholesky decomposition is performed on the two correlation matrices, $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2$, to obtain $\mathbf{L}_1$ and $\mathbf{L}_2$. We then take the Kronecker product between the identity matrix and $\mathbf{L}_1$ and $\mathbf{L}_2$ to obtain, $\mathbf{K}_1$ and $\mathbf{K}_1$, respectively. We then draw from a standard normal distribution and multiply the resulting values with $\mathbf{K}_1$ and $\mathbf{K}_1$ to obtain the observed values.

4.2.3 Analysis of psychiatric disorder bulk RNA-seq data

Normalized data and functional annotations of the bulk RNA-seq data can be found from the original study's manuscript [132]. We applied our transformation procedures to generate the squared matrix and pairwise product matrix for each module across all samples. We then performed pairwise testing using a linear module containing the group, age, sex, and intercept for each model using the standard *lm* function in R. We used the *summary* function to obtain the Wald test results and used the *p.adjust* function across all test to obtain the false discovery rate (FDR).

4.2.4 Analysis of breast cancer methylation data

We used TCGAbiolinks [133] to download all methylation beta value matrices from the Illumina Human Methylation 450 array in the TCGA-BRCA project. We additionally downloaded all clinical and biospeciment supplements available for these patients. We filtered the data to only contain primary tumor samples. We then removed individuals not in Asian, Black or African American, or White ancestries due to low counts. We also filtered samples to only include females and no missing values for initial weight of the tumor sample. We removed any methylation probes with more than 5% missingness and transformed the beta values to M-values by applying the logit transform (base-2). We then regressed out initial weight, OCT embedding, ancestry, age, and ethnicity from the M-values.

For each set of genes we analyzed, we found all probes on the methylation array that lied within the gene set. For the P53 pathway, we took the union of the hallmark P53 pathway and the KEGG P53 signaling pathways. For *Bao et al. 2019* genes, we simply took the list of 13 genes from their study (*C2orf40, CCND2, EZR, HIF3A, ITPRIPL1, KCNH8, KRT19, NDRG2, PCDHGA12, PCDHGA3, SIAH2, STAC2, TPD53*). We did the same for the genes in *Kristiansen et al. 2013* (*RASSF1A, CDH1, CCND2, ESR1, APC*). We applied our testing framework to each probe set while regressing out nitial weight, OCT embedding, ancestry, age, and ethnicity once again.

4.2.5 Analysis of stock market data

We obtained the daily adjusted close data from 01/03/2020 to 07/19/2022 using the *pdfetch* package in R for the following 30 stocks: BRK-B, TSLA, AMZN, AAPL, MSFT, ATVI, PEP, NTDOY, PFE, DIS, LULU, K, KMB, JNJ, NKE, ADDYY, GOOG, FB, IBM, HAS, GS, NSRGY, INTC, AMD, GPS, GLW, TSN, NFLX, MAT, KO. The data was then converted to percentage gains over the previous day. We then applied our testing framework to this data with no additional covariates except an intercept.

4.2.6 Datasets

Datasets can be found in the Gene Expression Omnibus or Array Expression. The datasets from *Gandal et al.* can be found at GSE28521, GSE28475, GSE35978, GSE53987, GSE17612, GSE12649, GSE21138, GSE54567, GSE54568, GSE54571, GSE54572, GSE29555, GSE11223, and E-MTAB-184. Normalized data and processing scripts as done in this study and *Gandal et al.* can be found in their manuscript. The TCGA breast cancer methylation data can be found publicly through the Genomics Data Commons. Stock market data can be pulled from publicly available databases such as Yahoo Finance.

4.2.7 Code Availability

The R package and examples for installing and applying our testing framework can be found at https://github.com/alecmchiu/EGExtend.

4.3 Results

4.3.1 Detecting differences in variance or covariance in gene networks between conditions

Given groups of individuals and a normalized gene expression matrix for a gene set (*e.g.* gene network), we generate two additional matrices that correspond to individual estimates of variance or covariance. We first demean each gene by the appropriate group mean. This explicit removal of mean effects allows us to exclusively detect changes in variance or covariance. Once demeaned, we generate the element-wise squared version of the demeaned matrix, which we refer to as the squared matrix. We also generate a matrix containing the pairwise product of every column combination of the demeaned matrix, which we refer to as the product matrix.

To perform a statistical test, we perform PCA on the squared or product matrix and use the resulting principal component scores/projections as an outcome variable to a linear

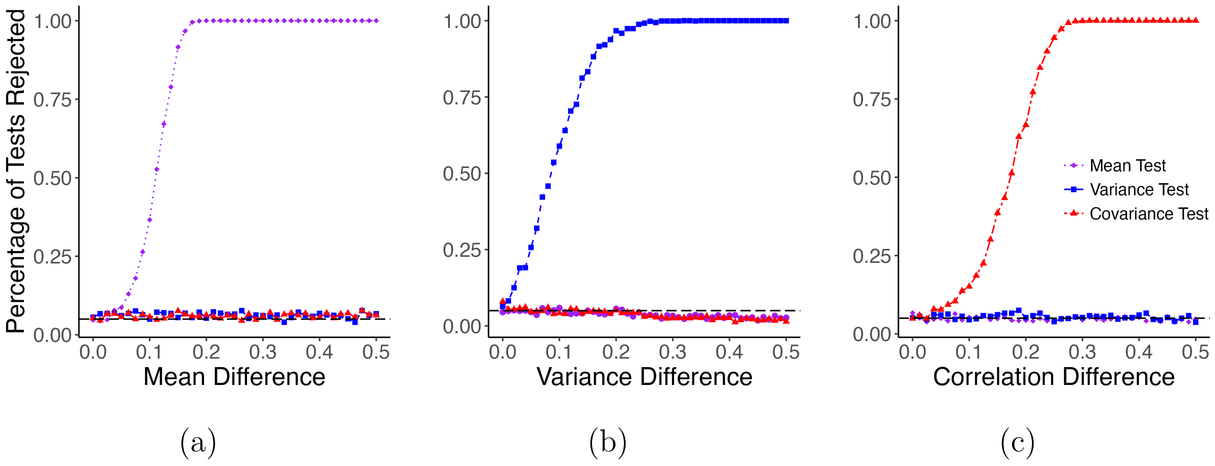


Figure 4.1: **All tests are calibrated under each effect of interest.** We performed simulations containing either mean differences (Figure 4.1a), variance differences (Figure 4.1b), or covariance differences (Figure 4.1c). $1,000$ simulations were performed on each point using a simulation dataset of 500 genes and 100 individuals. The black dashed line represents $p = 0.05$.

model with the input variable being the group membership. We then perform a Wald test on the coefficient of the group membership variable. To test for differences in variance, one performs the procedure on the squared matrix while performing the procedure on the product matrix will detect differences in covariance.

4.3.2 Simulations and controls reveal power to detect differences across several settings

To verify that our testing procedure calibrates properly, we perform a series of simulations across different ranges of parameters. To ensure that our simulations of mean effects were proper, we additionally performed a mean test, which applies our procedure using PCA on the original gene expression matrix rather than the squared or product matrix. The mean test is equivalent to the established test of performing a t-test on the eigengene of the standard gene expression matrix [129].

We first performed simulations across differing magnitudes of mean, variance, and covariance differences (Figure 4.1) using 100 individuals, 500 features, 250 non-zero interactions, and 1000 replicates for each setting. For each type of simulation, we generated data for two groups and added systematic shifts to ensure mean (Figure 4.1a), variance (Figure 4.1b), or covariance (Figure 4.1c) differences. For our mean simulations, we simulated data from two groups using a multivariate normal with the identity matrix for the covariance matrix. For the mean vector, we set one group to have the zero vector for the mean while the other group had a mean vector shifted between 0-0.75. For the variance simulations, we similarly generated data for two groups using a multivariate normal. However, we used a mean vector of zero for both and a diagonal matrix with the diagonal randomly generated from from a uniform distribution between 0 and 5. One group would use this diagonal matrix as a covariance matrix while the other would use the diagonal matrix with a shift between 0-0.5 as its covariance matrix. Lastly, our covariance simulations we used a Cholesky decomposition followed by a Kronecker product from two bivariate correlation matrices with each univariate normal representing a group. One group was set to have correlation value -0.99 while the other group had correlation shifted by a value between 0-0.5.

We ultimately find that each test is well calibrated for the specific phenomena targeted. Furthermore, the simulations reveal that each test is able to isolate either mean, variance, or covariance differences regardless of the presence of other effects (Figure 4.1).

We additionally show how the number of samples (Figure C.1), number of features (Figure C.2), number of features/genes with differences (Figure C.3), and percentage of samples in each group (Figure C.4) affects the power of our tests when other parameters are fixed. As we increase the number of samples, number of features, and number of features/gene with differences, we find that power increases as expected. Meanwhile, the power of the tests increases, as expected, as sample sizes between groups approaches the balanced setting.

While our testing framework is calibrated under simulations, we also assessed the framework against real data for which individual feature level differences had been found. For variance differences, we compared the results of MDSeq [125], a method comparing dispersion parameters between groups for individuals genes, for discovering differences in variance between sun-exposed and non-exposed skin tissue from Gene-Tissue expression project (GTEx) [134] as reported in their original study. In addition, we also obtained results from a basic F-test comparing the variance of two groups implemented in base R on the GTEx data. For covariance differences, we compared against DiffCor [126], a method that employs a Fisher Z-transformation on the Pearson correlation coefficient and performs a Z-test on the difference between groups, on two types of cancer [135] as performed in the original study for DiffCor. We additionally ran the CILP testing framework [117] on this data as well. CILP applies a Wald test between groups on the pairwise of products of features.

To assess our framework's ability to extend to real data, we generated artificial gene sets composed of either all significant (positive) genes or gene pairs or all non-significant (negative) obtained from running the individual feature methods (Table C.1). We randomly generated $1,000$ artificial gene sets for each method under both settings. The sizes of each gene set was drawn randomly between 50-44,850 features. Amongst the negative control gene sets, our testing framework never yielded a positive result at either the 0.05 or 0.1 significance levels. For the positive controls, our testing framework yielded a minimum of 91% and 98% significant tests at the 0.05 and 0.1 significance levels, respectively.

We also note that our method can be compared to two analogous works: network preservation analysis found in the WGCNA framework [136] and differential gene correlation analysis (DGCA) [137], a method for discovering differential gene correlation pairs. Network preservation analysis aims to seek if a network is preserved well within the data by performing a permutation test by shuffling genes between networks. However, this maintains the structure between groups and tests the robustness of the gene set rather than the relationship between the groups. DGCA works similarly to DiffCor in employing a Fisher Z-transformation followed by a permutation test, but also proposes a differential network test for correlation/covariance analogous to our framework by comparing the median Z-statistic in a gene set between two groups. We applied the DGCA test for comparing two groups and found it did not calibrate in our simulations (Figure C.5).

4.3.3 Differences in covariance and variance in psychiatric disorders with shared patterns of gene expression

We applied our testing framework to dataset found in *Gandal et al. 2018* [132]. This dataset consists of bulk RNA-seq data from 625 cerebral cortical samples divided between 266 controls, 15 alcoholism (ETOH), 41 autism (ASD), 83 major depressive disorder (MDD), 81 bipolar (BD), and 139 schizophrenia (SCZ) individuals. This study applied WGCNA [41] to identify 13 gene network modules. These 13 networks were broadly classified into different neuron types. Four modules (turquoise, salmon, purple, green) are neuron-specific modules. Two modules (greenyellow, blue) was found to be microglia-specific. Three modules were also found to cell-type specific for other brain-related cell-types. The yellow module is astrocyte-specific, the tan module is endothelial-specific, and the brown module is oligodendrocyte-specific.

Some modules were also found to be enriched in other specific functions. The turquoise module was found to be enriched in genes harbouring rare, non-synonymous mutations found in ASD. The purple module is enriched in mitrochondrial genes associated with neuron firing rates. The blue module was enriched in G protein–coupled receptors, cytokine-cytokine interactions, and hormone activity pathways. The red module is enriched in sequence-specific DNA binding. The brown module is enriched in mylination. The magenta module is enriched for nucleoplasm parts and chromatin binding. The black module is enriched in RNA binding and mRNA splicing. The pink module is enriched in DNA catabolism.

Similar to the original study, we use a random intercept model followed by a Wald test on coefficients for the group to account for the multiple group membership for some individuals. We additionally test all combinations of pairs as opposed to just the control as done in the original paper. After multiple testing correction, we find 26 significant differences in variance and 30 significant differences in covariance at the $FDR < 0.05$ significance level (Figure 4.2, 4.3, C.6).

Analogous to the original paper, we find that certain modules are disease-specific in terms of mean gene expression. However, we find that they are also disrupted in terms of variance

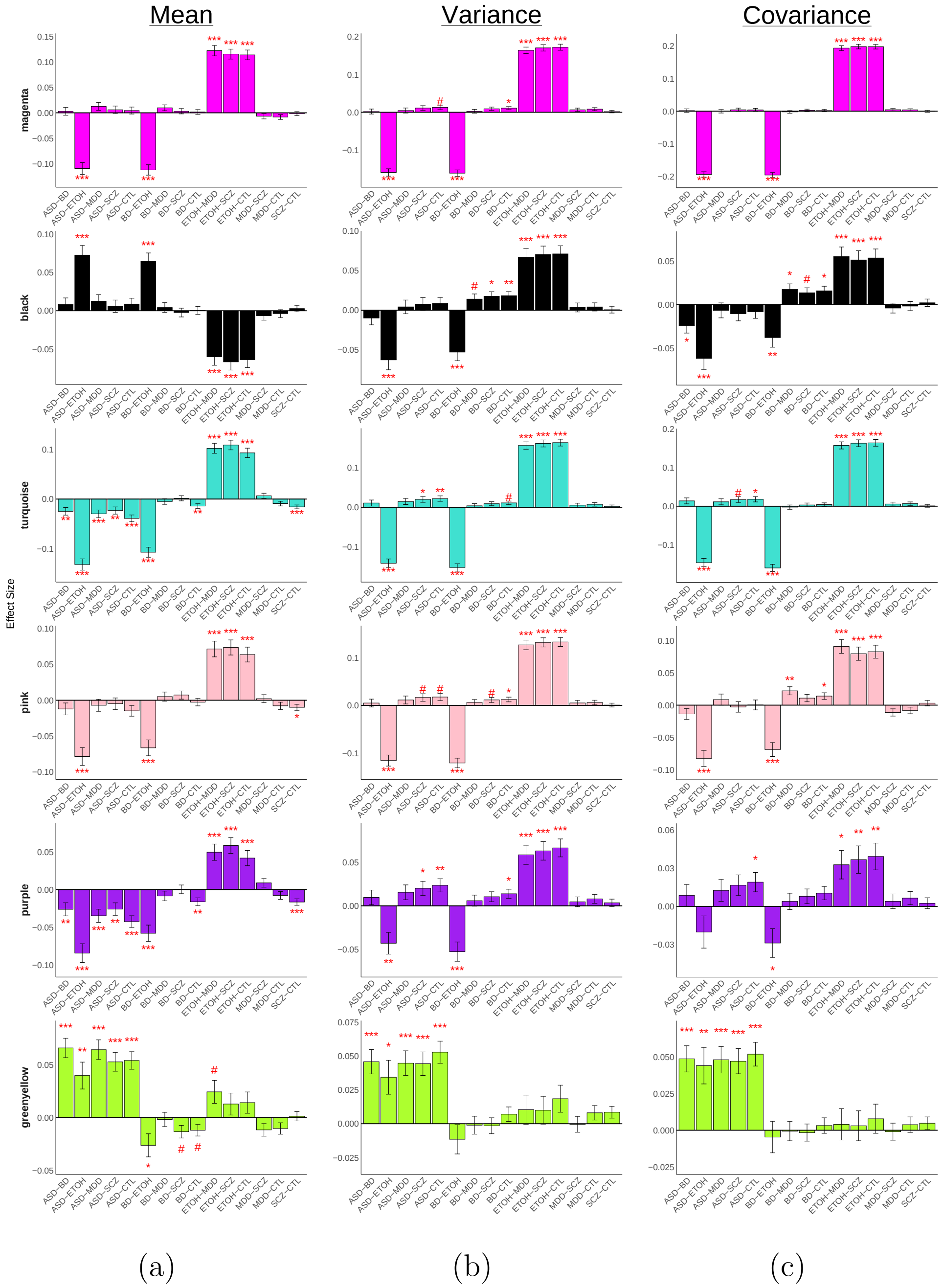
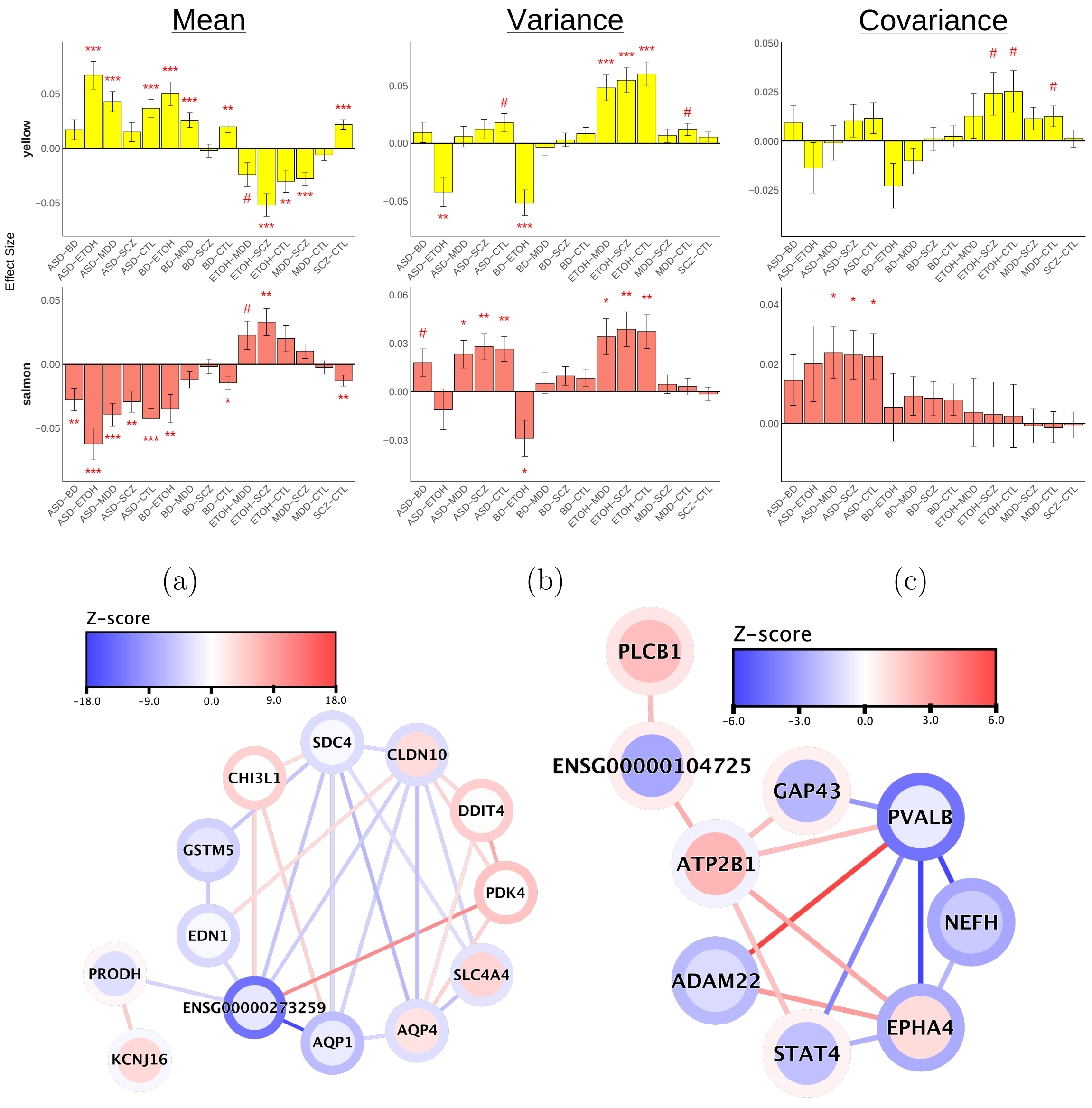


Figure 4.2: **Disease-specific modules disrupted on multiple levels.** We found that several modules are disease specific and disrupted on multiple levels such as mean differences (Figure 4.2a), variance differences (Figure 4.2b), or covariance differences (Figure 4.2c). The second group in each label denotes the reference group in the comparison. A pound sign, single asterisk, double asterisks, and triple asterisks denote significance at 0.1, 0.05, 0.01, and 0.001 after multiple testing correction, respectively.

and covariance as well (Figure 4.2). Five modules (magenta, black, turquoise, pink, and purple) display increased changes in terms of mean, variance, and covariance for alcoholism while the greenyellow module shows increased changes in terms of all three phenomena specifically for ASD.

We find that variance and covariance differences are able to discriminate diseases that share disruption in mean effects (Figure 4.3). The salmon and yellow modules both show disruption for alcoholism and ASD in terms of mean differences. The salmon module also displays changes in variance for both ASD and alcoholism, but only ASD has increases in covariance. The yellow module shows increases in variance for only alcoholism.

In addition to overall trends based on disease, we find differences in diseases that have high transcriptome overlap (Figure 4.4). SCZ and BD have a reported transcriptome correlation near 0.75, SCZ and ASD have a correlation near 0.5, and ASD and BD have a correlation near 0.4 [132]. Between BD and SCZ, only one difference in mean expression is found for the greenyellow module ($FDR = 0.0648$). Investigating for variance differences, additionally yields an increase in variance of the green ($FDR = 0.0521$), pink ($FDR = 0.0923$), red ($FDR = 0.0881$), and black ($FDR = 0.0108$) modules for BD relative to SCZ, despite showing no difference in mean expression for any of these modules. The green ($FDR = 0.0883$) and black ($FDR = 0.0622$) modules also show an enrichment in BD in terms of covariance. We find that ASD has several disease-specific modules (Figure 4.2, 4.3). However, we additionally find variance and covariance differences that are overlooked when only looking at mean differences. ASD shows decreases in variance in the blue module ($FDR = 0.0350$) and covariance in the blue ($FDR = 0.0004$), black ($FDR = 0.0234$), and red ($FDR = 0.0406$) module relative to BD, but no difference in terms of any of these modules in terms of mean expression. Similarly, ASD has a decrease in covariance in the blue module ($FDR = 0.00201$) compared to SCZ, but no differences in mean or variance for the blue module.



(a)

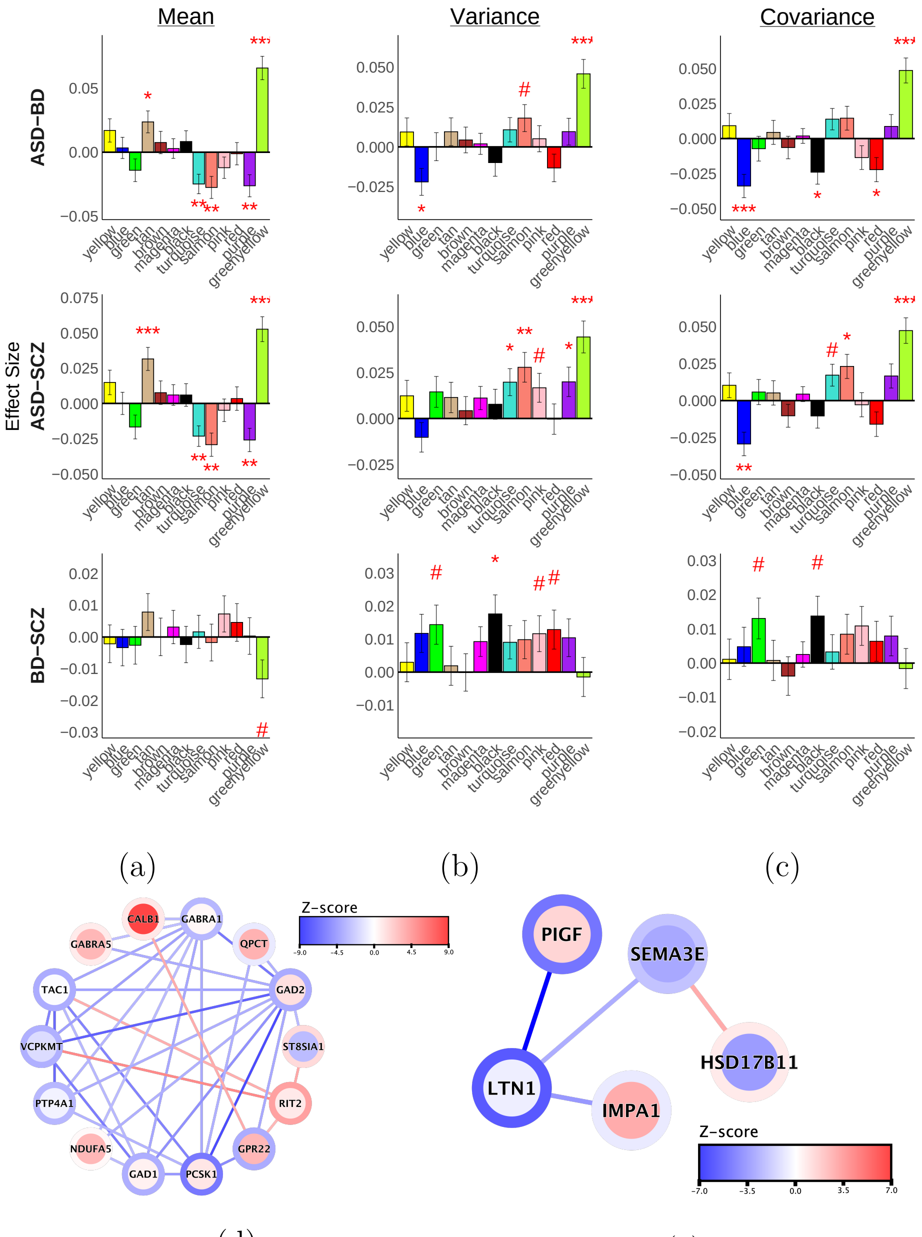
(b)

(c)

(d)

(e)

Figure 4.3: **Variance and covariance differences discriminate diseases.** We found that two modules are that disrupted for multiple diseases on the mean level (Figure 4.3a) become disease-specific when examining variance (Figure 4.3b) or covariance differences (Figure 4.3c). The second group in each label denotes the reference group in the comparison. A pound sign, single asterisk, double asterisks, and triple asterisks denote significance at 0.1, 0.05, 0.01, and 0.001 after multiple testing correction, respectively. Cytoscape network diagrams for key characteristics of the salmon module (Figure 4.3d) and yellow module (Figure 4.3e) are also shown. Network diagrams show ASD relative to BD. Nodes shown must past a Z-score threshold of 1.96 (Figure 4.3e) or 3 (Figure 4.3d) in either mean or variance differences and a covariance difference. Node color represents mean value, border color represents variance value, and edge color represents covariance value.



(a) (b) (c)

(d) (e)

Figure 4.4: **Disruption in diseases with highly correlated transcriptomes show difference in variance or covariance.** We found that diseases that show high transcriptome correlation exhibit difference in several modules including ones where there are no disruptions on the mean level (Figure 4.4a) but do exhibit differences in variance (Figure 4.4b) or covariance (Figure 4.4c). The second group in each label denotes the reference group in the comparison. A pound sign, single asterisk, double asterisks, and triple asterisks denote significance at 0.1, 0.05, 0.01, and 0.001 after multiple testing correction, respectively. Cytoscape network diagrams for key features of SCZ relative to BD are shown for the green (Figure 4.4d) and black (Figure 4.4e) modules. Nodes shown must past a Z-score threshold of 3 in either mean or variance differences and a covariance difference. Node color represents mean value, border color represents variance value, and edge color represents covariance value.

4.3.4 Applications to other data types

Similar to the original eigengene mean testing framework, our transformations do not assume to be operating on expression data. WGCNA and eigengene analysis have both been applied to several other data types such as methylation [138, 139, 140, 141]. Similarly, our testing framework extends to other data types as well. We exemplify this by applying our testing framework to breast cancer methylation data from The Cancer Genome Atlas (TCGA) and stock market data.

4.3.4.1 Differences in variances for methylation in vital status-related genes

We analyzed 737 breast cancer samples from TCGA for mean, variance, and covariance differences across three sets of methylation probes between live and dead patients. The first set of probes involved genes within the P53 gene pathway, a critical pathway in breast cancer [142, 143], (290 probes). The second set of probes were probes found in a 13-gene signature found to be indicative of low survival from *Bao et al. 2019* [144] (586 probes). The third set of probes were involved in a 5-gene signature from *Kristiansen et al. 2013* [145] found to be indicative of various clinical factors (203 probes). In all three probe sets, variance was found to be significantly increased ($p < 0.05$) (Figure C.7) in patients who did not survive while there were no significant mean or covariance effects.

4.3.4.2 Application to non-omic data

We analyzed daily stock market returns for 30 stocks from 01/03/2020 to 07/19/2022 (640 days). We obtained closed values and converted them to percentage gains or losses over the previous day's close. We labeled each date under the presidency the fell under, Trump (01/03/2020 to 01/20/2021) or Biden (01/21/2021 to 07/19/2022) and ran our testing framework on the stock market data using the presidency as the class. We uncovered significant changes in variance ($p = 2.1 \times 10^{-7}$)) and covariance ($p = 5.3 \times 10^{-5}$) under Trump relative to Biden, but no significant change in mean returns ($p = 0.38$) (Figure C.8).

4.4 Discussion

We propose a simple testing framework that extends mean differential expression testing using eigengene analysis to test for differences in variance or covariance in sets of genes. The testing framework consists of transforming normalized data, running PCA, and performing linear regression. We show through simulations that our testing procedure is calibrated and extends to controls generated from differences detected in real data from previous established feature-level methods. We further apply our testing framework to several psychiatric diseases to discover novel differences in variance or covariance in gene networks relevant to differentiating various psychiatric diseases.

Our testing framework allows researchers to prioritize certain functions and modules for diseases. For instance, modules with differences on multiple levels (*e.g.* mean, variance, covariance) can be prioritized since they are disrupted on multiple levels (Figure 4.2). We also find that differences in variance and covariance can disentangle diseases with shared mean effects as is the case between ASD and alcoholism (Figure 4.3). Lastly, our exploration of psychiatric disorders reveals differences in variance or covariance for modules that show no differences in mean. This illustrates the important point that relevant modules that are disrupted can be overlooked by only looking at differences in mean effects.

Our testing procedure relies on general statistical principles and has no assumptions restricting it to gene expression. Therefore, this test can be applied to several other biological data types such as methylation or even stock market data as we have shown. Differences in variance or covariance for many diseases in such data types has largely been unexplored and we hope our work helps motivate more analyses in these data types.

Since we utilize the eigengene framework, our use of linear regression is not the only extension of eigengene analysis that can be used. Several studies use other tests such as multiple group comparison tests (*e.g.* ANOVA, Tukey HSD test) [146]. The median test from DGCA [137] failed to calibrate in our simulations likely due to the fact that our simulations are mainly composed of feature pairs with no covariance difference, resulting in poor performance. That being said, the DGCA test performed well at smaller effect sizes, highlighting the importance of using the correct test for one's scenario. In fact, several proposed testing frameworks such as the DGCA median test can be combined with our eigengene framework to form a new testing procedure. Several other concepts relevant to eigengene analysis such as connectivity [147] can also potentially be used in conjunction with our transformations. Functions for such downstream eigengene analyses can be found in the WGCNA R package [41] and work seamlessly with our framework when provided with the relevant transformed matrix.

The summarizing of a matrix into a single vector also allows for a new approach to quantitative trait loci (QTL) mapping. Several works have explored expression QTLs (eQTLS) as well as variance [148, 149] and covariance/co-expression QTLs [117, 150]. However, the approaches have largely been dedicated to single genes or pairs of genes rather than a network or collection of genes. Previous approaches to perform summarization include averaging a matrix into a single vector [151, 152, 153, 154, 155], but the use of the eigengene offers new ways to examine each type of QTL in a new way.

A limitation of our testing framework is that it relies on having discrete groups in order to remove any changes in means between the group. This is what ultimately allows our testing framework to exclusively test for either variance or covariance differences. An open question for this testing framework is how we can extend it to be applicable to continuous phenotypes such as age. While one can derive discrete groups through processes such as binning, specificity and ability to detect smaller changes are lost.

A constant topic of discussion in the dimensionality reduction community is often what the true size of latent dimension should be [156, 157, 158]. In classical eigengene analysis, PCA is used to reduce to a single dimension explaining the most variance. This is a conservative approach as important trends that explain smaller amounts of variability often appear in deeper principal components. However, there is much work exploring how multiple principal components can be used or combined for statistical testing [159, 160, 161]. How many principal components and how they should be combined in the context our transformations remains an open question.

Lastly, how we can leverage covariance or variance differences in an unsupervised fashion remains an open question. Our testing framework exists as a supervised analysis as it requires group labels to detect variance and covariance differences. However, the same transformations we utilize (*i.e.* element-wise squaring) could be utilized as forms of feature augmentation for unsupervised analyses such as clustering. Such ideas are not unheard of as there are forms of PCA such as kernel PCA [162], which utilize similar ideas of transforming the data, but have limitations in terms of interpretability. Similarly, techniques such as t-SNE [163] and UMAP [164] can also utilize non-linear effects such as changes in variance or covariance but lack interpretability.

CHAPTER 5

Closing Remarks

5.1 Conclusions

We have presented three scalable methods for detecting structue within large-scale genomic data. We present ProPCA, a scalable probabilistic principal component analysis method, that utilizes the discrete structure of genotypes to accelerate matrix-vector multplication through the Mailman algorithm [34]. We demonstrate that ProPCA is accurate and efficient across both simulations and real datasets. Furthermore, we estimate population structure within the UK Biobank and leverage the probabilistic model of ProPCA along with the inferred structure to perform a statistical test to identify genomic sites under recent putative selection.

We also present SCOPE, a scalable method for inferring population structure in the form of admixture proportions. SCOPE utilizes the previously proposed model from ALStructure [37] and accelerates its inference using similar strategies as ProPCA such as the Mailman algorithm. We further accelerate ALStructure by using randomized eigendecomposition, a strategy often used for fast decompositions such as PCA. We show that SCOPE is orders of magnitude faster than existing methods while maintaining accuracy and memory requirements through comparisons on both simulated and real genomic datasets. We additionally allow SCOPE to perform supervised analyses using allele frequency estimates from previous studies to improve interpretability, runtime, and accuracy.

Lastly, we present a simple statistical testing framework to investigate structure changes in variance or covariance between two groups in a set of genes by extending mean differential expression testing from eigengene analysis. Our framework consists of applying transformations on normalized data, performing PCA through randomized eigendecomposition, and conducting a statistical test. We show through simulations that our testing procedure is calibrated and apply it on psychiatric disorder RNA-seq data to uncover several differences in variance and covariance.

5.2 Future Work

The work presented here provides several directions for new analyses and extensions to develop more novel methods. The Mailman algorithm [34] employed in both ProPCA and SCOPE can be incorporated into several genomic methods to speed-up inference. For example, association testing can be formulated as a matrix-vector multiplication problem. The algorithm can also be integrated into other PCA methods similar to the combination of the Arnoldi method and the Mailman algorithm used in SCOPE. The probabilistic model of ProPCA also provides for several extensions such as a natural method of handling missing data or incorporating other information such as linkage disequilibrium.

SCOPE can also be extended in several ways. Several methods that perform structure inference on other genomic datasets [100, 99] utilize semi-supervised approaches that allow it to handle both known and unknown populations. One approach for enabling SCOPE to perform semi-supervised analysis is using a multi-stage inference procedure where supervised inference is first applied and unsupervised inference is applied on the residual or unexplained structure. Another open problem is the incorporation of linkage disequilibrium into the method. Most methods, including SCOPE, ignore this information. While there are methods such as fineSTRUCTURE [101] that can utilize this information, scaling the inference of these methods remains an unsolved challenge.

Finally, our testing framework is general enough to be applied to several other data types. While we demonstrate its use on methylation and stock market data, we hope it is utilized in other contexts such as microbiome data or imaging data. A broader question brought up by our exploration of structure differences from variance and covariance is how such differences can be incorporated into unsupervised analyses such as clustering. There are existing methods such as kernel PCA [162], t-SNE [39], and UMAP [40] that can non-linear effects, but these methods often lack interpretability and the ability to scale.

Finally, we hope that these methods provide new biological insight in large-scale datasets yet to come. We primarily focus on our analyses on the UK Biobank, as it is the largest repository of genomic data available to us, but several other biobanks are yet to come such as the Million Veterans Project [3] and BioBank Japan [165].

APPENDIX A

Supplementary Material - Scalable probabilistic PCA for large-scale genetic variation data

Section A.1 contains additional information on the White British analysis. Section A.2 compares our selection statistic to other existing selection statistics. Section A.3 explores the time-scales of our selection hits. Section A.4 explores the application of ProPCA to missing data. Section A.5 details the implementation of ProPCA. Section A.6 explains our variant of the Mailman algorithm for left multiplication. Section A.7 details the convergence of ProPCA in the noiseless setting. Section A.8 explores the contribution of the Mailman algorithm to scalability.

A.1 White British Selection Scan and Analysis

Among the significant loci that we did not highlight in the main text, there are several genic loci have biological significance.

Transcriptome-wide association studies (TWAS) suggest that gene expression at *HERC6* is associated with gout ($p = 3.8 \times 10^{-123}$) [166]. Epidemiological studies in the UK also have shown that Wales, the geographic region associated with differences in *HERC6* allele frequencies, is among the regions of the UK with the highest prevalence and incidence in the UK [167]. The specific variant that is putatively under selection at this locus, rs112873858, does not appear to be significantly associated with gout in the UK Biobank however (logistic regression $p = 0.2395$).

HERC2 (hect domain and RLD2), contains a single SNP in *HERC2* that is a primary determinant of light eye color in modern Europeans [168] and has been previously shown to be under selection [169]. A number of other SNPs in the *HERC2* locus have also been shown to be associated with iris color [170]. In the UK Biobank, we find that the SNP with the most significant p-value in *HERC2*, rs1129038, is associated with childhood sunburn occasions ($p = 6 \times 10^{-134}$) as well as skin and hair pigmentation ($p = 9.4 \times 10^{-103}$) (Tables A.8, A.9).

SKI is a proto-oncogene located at a region close to the p73 tumor suppressor gene [171]. It is implicated in the TGF-β signaling pathway [172] and has been shown to play a role in a variety of cancers [171, 173]. However, our specific locus does not appear to be significantly associated with any cancer in the UK Biobank.

Our combined selection statistic also resulted in an additional genic loci we did not highlight in the main text. The *AMPH* locus is located in the gene that codes for the amphiphysin protein, which is associated with the cytoplasmic surface of synaptic vesicles [174]. The gene is also implicated in stiff person syndrome and breast cancer [174]; however we were unable to find any significant associations with traits in the UK Biobank.

A.2 Comparison of Selection Statistics

Several approaches have been previously proposed [53, 175, 176] to discovering signals of putative selection based on PCA. These approaches look for variants with large differences in alleles frequencies between populations or individuals differentiated along an axis (principal component). Typically, the PCs correspond to population structure so that these methods correspond to tests for SNPs that are not well described by the PCs. The proposed statistics attempt to detect SNPs as outliers relative to the structure captured by either a single PC or the space spanned by k PCs. The differences across all these statistics arise from the statistical assumptions of the underlying model of population structure.

[175] examines several statistics to rank SNPs based on the PC loadings and uses an outlier approach to determine putative targets of selection. [53] formulates a hypothesis

testing framework to show that, under a model of drift, their proposed statistic for the k-th PC has a chi-squared distribution with one degree of freedom. [176] employs a chi-squared Malahanobis distance distance as a means of outlier detection after regressing each SNP by the k principal components.

Our proposed statistic is similar to the statistic proposed in [176] in its use of an outlier detection approach, *i.e.*, looking for SNPs that are not well-described by the first K PCs. To aid interpretability, we further project the residual variance along each of the k PCs, in turn, to identify the specific axes of variation along which the SNP tends to be an outlier.

A.3 Time-scale of selection hits

To better understand the time-scale of the episode of selection that our proposed statistic is sensitive to, we examined the estimated allelic ages of the mutations at the hits detected by our statistic. We obtained estimates of allelic ages using the Human Genome Dating Atlas of Variant Age [78]. We restricted our analysis to ages estimated from variants genotyped in the 1000 Genomes Project [66]. Further, the underlying method for estimating variant ages assumes that the alternate allele is the derived allele. When this assumption is violated, the resulting estimates may not be valid. Thus, we restricted our analysis to variants at which the alternate allele is the derived allele to obtain a total of 42 variants (out of our initial list of 63 hits that are significant across each of the five PCs as well as the combined statistic). The mean ages of these alleles was estimated to be around $11,555$ generations using the mutation clock, $18,946$ generations using the recombination clock, and $19,007$ generations using the combined clock. However, there is substantial variation in the allelic ages estimates. For eample, 17 of the variants have ages less than $5,000$ generations using the combined clock.

We compared our allelic ages estimates to those of the hits from a recent work designed to detect recent episodes of positive selection [77]. We restricted our analysis to the list of variants from the UK10K with SDS scores > 4. This resulted in 1620 variants out of a total of $4,451,435$ variants with SDS scores available (top 4×10^{-4} of the SDS scores). We then used the allelic ages for each of these variants available from the Human Genome Dating Atlas again restricting our analysis to those variants where the alternate allele matches the derived allele yielding a list of 920 variants. The mean ages for these variants are approximately $7,620$ generations (mutation clock), $12,471$ generations (recombination clock), and $11,944$ generations (combined clock). We note that there is considerable variation in the ages across variants. Figure A.9 shows that the variants identified by the SDS statistic tend to younger on average than those from our statistic (mean age of $12,471$ generations for SDS vs $18,700$ for our statistic). This difference is nominally statistically significant using a Mann-Whitney-Wilcoxon test ($p = 0.002$,0.001, and 8×10^{-4} for each of the mutation, recombination, and combined clocks). We caution however that the hits in each of the lists are unlikely to be statistically independent (for example, there are multiple variants that are present in the LCT locus). Further, there is considerable uncertainty associated with the age of these variants and a more careful analysis would need to account for this uncertainty.

A.4 Application of ProPCA to missing data

A.4.1 PCA with Missing Data

The use of a probabilistic model allows for handling missing entries in the genotype matrix. We assume that the genotype data is missing at random (MAR) [177], *i.e.*, the missingness depends only on the other observed values. We partition the observed data $\boldsymbol{G}$ into observed and unobserved entries. In the missing data setting, the observation model becomes:

$$\boldsymbol{g}_i|\boldsymbol{x}_i, \boldsymbol{\epsilon}_i \quad = \quad \boldsymbol{\mu} + \boldsymbol{C}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i \tag{A.1}$$

Here $\boldsymbol{\mu}$ is a length m vector denoting the mean genotype vector. Unlike the fully observed setting where the maximum likelihood estimate of $\boldsymbol{\mu}$ is equal to the sample mean $\overline{\boldsymbol{g}}$, in the missing data setting, we need to estimate $\boldsymbol{\mu}$ within the EM algorithm.

$$
\begin{aligned}
O &= \{\ (i,j) \ \mid \ \ g_{ij} \ is\ observed\ \}, \\
O_j &= \{\ i \ \mid \ \ (i,j) \in O\ \}, \\
O_i &= \{\ j \ \mid \ \ (i,j) \in O\ \}, \\
\boldsymbol{x}_j &= j^{th}\ column\ of\ \mathbf{X}, \\
\boldsymbol{c}_i &= i^{th}\ row\ of\ \mathbf{C}\ written\ as\ a\ column, \\
\mu_i &= the\ mean\ of\ g_{ij}\ where\ j \in O_i
\end{aligned}
$$

A.4.2 EM for PCA with Missing Data

$$\text{E Step:} \qquad \boldsymbol{x}_j = (\sum_{i \in O_j} \boldsymbol{c}_i \boldsymbol{c}_i^T)^{-1} \sum_{i \in O_j} \boldsymbol{c}_i (y_{ij} - \mu_i) \tag{A.2}$$

$$\text{M Step:} \qquad \boldsymbol{c}_i = (\sum_{j \in O_i} \boldsymbol{x}_j \boldsymbol{x}_j^T)^{-1} \sum_{j \in O_i} (y_{ij} - \mu_i) \boldsymbol{x}_j \tag{A.3}$$

$$\mu_i = \frac{1}{|O_i|} \sum_{j \in O_i} (g_{ij} - \boldsymbol{c}_i^T \boldsymbol{x}_j) \tag{A.4}$$

Using the same ideas of the Mailman algorithm, the EM algorithm for missing data has a running time of $\mathcal{O}(\frac{nmk}{max(\log_3 n, \log_3 m)} + n_{missing} k^2)$ per iteration. Since the percentage of missing data is quite low, we can use the probabilistic model to efficiently handle missing data.

We evaluated the effectiveness of this extended model using simulated genotypes with missing data (Figure A.11). We compared the accuracy of the PCs estimated using the extended model to the PCs estimated by running the EM algorithm on genotype data that was imputed through a random draw from a binomial distribution parameterized by the allele frequencies.

We simulated ten sets of complete genotypes with 50,000 SNPs and 10,000 individuals from 5 and 10 populations, each at differing F_{ST} levels from 0.001 to 0.01 at intervals of 0.001. We simulated missing data by randomly removing 5% and 20% of the genotypes. To estimate the variance of our method, we averaged over 10 datasets.

For each method tested, we computed the MEV between the PCs inferred from the missing data and the PCs computed by applying SVD to the original genotype data with no missing values. Figure A.11 shows that the PCs inferred from the ProPCA implementation that explicitly handles missing data are more accurate than the PCs computed by running ProPCA on imputed genotypes (Figures A.11a, A.11b). Furthermore, we see that ProPCA can infer PCs comparable to running mean imputation followed by a full SVD (Figure A.11c).

A.5 Implementation details

Application of the Mailman algorithm to the EM algorithm For a genotype matrix $\boldsymbol{G}$ where $m > \lceil \log_3(n) \rceil$, we partition $\boldsymbol{G} = \left(\boldsymbol{G}_1^{\mathrm{T}} \dots \boldsymbol{G}_B^{\mathrm{T}}\right)^{\mathrm{T}}$ into $B = \lceil \frac{m}{log_3(n)} \rceil$ sub-matrices each of size $\lceil log_3(n) \times n \rceil$ and decompose each $\boldsymbol{G}_b = \boldsymbol{U}_n \boldsymbol{P}_b$.

The M-step (Equation 2.5) requires computing $\boldsymbol{G\alpha}$ for k distinct vectors $\boldsymbol{\alpha}$. We compute $\boldsymbol{G\alpha} = \begin{pmatrix} G_1\boldsymbol{\alpha} \\ G_2\boldsymbol{\alpha} \\ \vdots \\ G_B\boldsymbol{\alpha} \end{pmatrix}$. Since each of the products $\boldsymbol{G}_b\boldsymbol{\alpha}, b \in \{1, \dots, B\}$ can be computed in $\mathcal{O}(n)$ operations (given $\boldsymbol{U}_n$, and $\boldsymbol{P}_b$), the entire matrix-vector product $\boldsymbol{G\alpha}$ can be computed in $\mathcal{O}(\frac{nm}{\log_3(n)})$ time.

The E-step (Equation 2.4) requires computing $\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{G}$ for k distinct vectors $\boldsymbol{\beta}$. We compute this product as $\sum_{b=1}^{B} \boldsymbol{\beta}_b^{\mathrm{T}} \boldsymbol{G}_b$ in $\mathcal{O}(\frac{nm}{\log_3(n)})$ time where each term in the sum is computed using our novel variant of the Mailman algorithm.

Likelihood Computation To check for convergence, we need to compute the likelihood of the parameters in each iteration of the EM algorithm which is equivalent to the computing the squared Frobenius norm of the error matrix, *i.e.*, $||\mathbf{Y} - \mathbf{CX}||_F^2$.

$$
\begin{aligned}
||\mathbf{Y} - \mathbf{CX}||_F^2 &= tr[\ (\mathbf{Y} - \mathbf{CX})(\mathbf{Y} - \mathbf{CX})^T] \\
&= -2tr(\mathbf{Y^T CX}) + tr(\mathbf{X^T C^T CX}) + const
\end{aligned}
$$

Let $\boldsymbol{Z} = \boldsymbol{C}^{\mathrm{T}}\boldsymbol{Y}$. $\boldsymbol{Z}$ and $\boldsymbol{X}$ are $k \times n$ matrices so that the first term in the sum above ($tr(\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{X})$) can be computed in $\mathcal{O}(nk)$ time. $\boldsymbol{Z}$ can be computed in $\mathcal{O}(\frac{nmk}{max(\log_3(n),\log_3(m)})$ using the Mailman algorithm. Thus, the likelihood can be computed in $\mathcal{O}(\frac{nmk}{max(\log_3(n),\log_3(m)} + nk)$.

We note that the columns of the Maximum Likelihood Estimate (MLE) of $\mathbf{C}$ do not correspond to the principal components of $\boldsymbol{Y}$ but instead span the principal subspace of the top k eigenvectors of $\mathbf{Y}$. We can orthogonalize the matrix $\mathbf{C}$ to obtain the principal components in time $\mathcal{O}(mk^2)$, using *e.g.*, the Q-R decomposition.

Efficient implementation of the Mailman algorithm There are several considerations in an efficient practical implementation of the Mailman algorithm. While the multiplication with the $\boldsymbol{U}$ matrix is obtained by a recursion, we convert this into an iterative algorithm. Another important factor arises from the fact that the Mailman algorithm needs access to elements in the input vector that are not necessarily located in consecutive memory addresses. This can lead to frequent cache misses that can substantially reduce the efficiency of the implementation. To get around this limitation, we implemented a batched version of the Mailman algorithm. This version uses the idea that typically we need to multiply more than one vector at a time, *e.g.*, we often need to compute $k = 5$ PCs. Our implementation operates on the batch of input vectors at a time using the resulting locality among the input vectors. We use a default batch size of 10 although other batch sizes could also be used.

Memory considerations In the mailman algorithm, the matrix $\boldsymbol{U}_n$ is only used implicitly and need not be stored. The $\boldsymbol{P}$ matrix has the property that each column has exactly one entry that is one while all the other entries are zero. $\boldsymbol{P}$ can be stored as a length n vector $\boldsymbol{p}$ indicating the locations of the one entry in each column of $\boldsymbol{P}$. Since each element of the

$\boldsymbol{p}$ vector is an integer, such that $p_i \in [1, n], i \in \{1, \ldots, n\}$, we can store $\boldsymbol{p}$ in $\lceil log_2(n) \rceil$ bits. This can be efficiently represented by storing 2 or more elements of p in a single four byte integer. The storing and retrieval of an element can be performed by bit operations which increase the computational complexity moderately while reducing the memory requirements considerably.

A.6 Novel variant of the Mailman algorithm for left multiplication

The EM algorithm requires alternate left and right multiplication of genotype matrix $\boldsymbol{G}$ in the E- and M-steps respectively. One approach to using the Mailman algorithm for each step consists of partitioning $\boldsymbol{G}$ along the columns and the rows respectively followed by computing decompositions of each of the resulting sub-matrices. This approach, however, doubles the memory requirement of the resulting algorithm. Instead, we propose a variant of the Mailman algorithm for left multiplication of a matrix with a vector that uses the same decomposition as for right multiplication.

Recall that for right multiplication, we would like to compute $\boldsymbol{c} = \boldsymbol{A}\boldsymbol{b}$ for an arbitrary real-valued vector $\boldsymbol{b}$ and a $m \times n$ matrix $\boldsymbol{A}$ whose entries take values in $\{0, 1, 2\}$. We assume that $m = \lceil \log_3(n) \rceil$. The Mailman algorithm decomposes $\boldsymbol{A}$ as $\boldsymbol{A} = \boldsymbol{U}_m\boldsymbol{P}$. Here $\boldsymbol{U}_m$ is the $m \times m_0$ matrix whose columns containing all $m_0 = 3^m$ possible vectors over $\{0, 1, 2\}$ of length m. $\boldsymbol{P}$ is a $m_0 \times n$ matrix. We set an entry $P_{i,j}$ to 1 if column j of $\boldsymbol{A}$ matches column i of U_m: $A^{(j)} = U_m^{(i)}$. All other entries of $\boldsymbol{P}$ are set to zero. The decomposition of any matrix $\boldsymbol{A}$ into $\boldsymbol{U}_m$ and $\boldsymbol{P}$ can be done in $\mathcal{O}(nm)$ time. Given this decomposition, the desired product $\boldsymbol{c}$ is computed in two steps, each of which has $\mathcal{O}(n)$ time complexity [34]:

$$\boldsymbol{d} = \boldsymbol{P}\boldsymbol{b}, \quad \boldsymbol{c} = \boldsymbol{U}_m\boldsymbol{d}$$

We now describe an algorithm to compute $\boldsymbol{f}^{\mathrm{T}} = \boldsymbol{e}^{\mathrm{T}}\boldsymbol{A}$ using the same decomposition

$\boldsymbol{A} = \boldsymbol{U}_m\boldsymbol{P}$. As in the setting of right multiplication, this algorithm proceeds in two steps:

$$\boldsymbol{g}^{\mathrm{T}} = \boldsymbol{e}^{\mathrm{T}}\boldsymbol{U}_m, \quad \boldsymbol{f}^{\mathrm{T}} = \boldsymbol{g}^{\mathrm{T}}\boldsymbol{P}$$

For the first step, we have:

$$\begin{aligned} \boldsymbol{g}^{\mathrm{T}} = \boldsymbol{e}^{\mathrm{T}}\boldsymbol{U}_m &= \begin{pmatrix} e_1 & \boldsymbol{e}^{\mathrm{T}}_{2:m} \end{pmatrix} \begin{pmatrix} \mathbf{0}^{\mathrm{T}}_{3^{m-1}} & \mathbf{1}^{\mathrm{T}}_{3^{m-1}} & \mathbf{2}^{\mathrm{T}}_{3^{m-1}} \\ \boldsymbol{U}_{m-1} & \boldsymbol{U}_{m-1} & \boldsymbol{U}_{m-1} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{e}^{\mathrm{T}}_{2:m}\boldsymbol{U}_{m-1} & e_1\mathbf{1}_{3^{m-1}} + \boldsymbol{e}^{\mathrm{T}}_{2:m}\boldsymbol{U}_{m-1} & e_1\mathbf{2}^{\mathrm{T}}_{3^{m-1}} + \boldsymbol{e}^{\mathrm{T}}_{2:m}\boldsymbol{U}_{m-1} \end{pmatrix} \end{aligned} \quad \text{(A.5)}$$

Here e_1 is the first element of $\boldsymbol{e}$ and $\boldsymbol{e}^{\mathrm{T}}_{2:m}$ is a vector of length $m-1$ consisting of elements 2 to m of vector $\boldsymbol{e}$.

This gives us a recursive algorithm to compute $\boldsymbol{g}$ with base case :

$$\begin{aligned} e_m\boldsymbol{U}_1 &= e_m\begin{pmatrix} 0 & 1 & 2 \end{pmatrix} \\ &= \begin{pmatrix} 0 & e_m & 2e_m \end{pmatrix} \end{aligned} \quad \text{(A.6)}$$

The time complexity of this algorithm is given by $T(m) \leq 3^m + T(m-1) \leq 3^{m+1} = 3 \times 3^{\lceil \log_3(n) \rceil} = \mathcal{O}(n)$.

For the second step, note that each column of $\boldsymbol{P}$ has exactly one non-zero entry (with value equal to one). Thus, each entry of $\boldsymbol{f}$ can be computed in constant time so that $\boldsymbol{f}$ can be computed in $\mathcal{O}(3^m) = \mathcal{O}(n)$ time.

Thus, the total time complexity of computing $\boldsymbol{f}$ is $\mathcal{O}(n)$ instead of $\mathcal{O}(n\log_3(n))$ using naive matrix-vector multiplication.

For a general matrix $\boldsymbol{A}$ where $m > \lceil \log_3(n) \rceil$, we partition $\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_B \end{pmatrix}$ into $B = \lceil \frac{m}{log_3(n)} \rceil$ sub-matrices each of size $\lceil log_3(n) \times n \rceil$ and decompose each $\boldsymbol{A}_b = \boldsymbol{U}_n\boldsymbol{P}_b$. To now compute $\boldsymbol{f}^{\mathrm{T}} = \boldsymbol{e}^{\mathrm{T}}\boldsymbol{A}$, we compute $\sum_{b=1}^{B} \boldsymbol{e}_b^{\mathrm{T}}\boldsymbol{A}_b$. Each product can be computed in $\mathcal{O}(n)$ time so that $\boldsymbol{f}$ can be computed in $\mathcal{O}(\frac{nm}{\log_3(n)})$.

A.7 Convergence of ProPCA in the noiseless setting

There are several techniques to analyze the convergence properties of ProPCA. Under the assumption that the linear Gaussian model is true, convergence results of the EM algorithm can be invoked [178]. An alternate view of convergence in the setting where $\sigma^2 \to 0$ arises from viewing the EM updates as mathematically equivalent to alternating least squares [179]. In this view, we can show that the spectral norm of the reconstruction error, *i.e.*, the error between the data matrix $\boldsymbol{Y}$ and its rank-k approximation $\boldsymbol{CX}$, decreases to the optimal value at a rate that is exponential in the number of iterations. Our arguments follow from a combination of previous theoretical results.

The range of the matrix $\boldsymbol{C}^{(t)}$ obtained at the end of iteration t of the EM algorithm is the same as the range of the matrix $\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}^t}\boldsymbol{C}_0$ (Theorem 5 of Szlam et al. 2017). Setting $\boldsymbol{C}_0 = \boldsymbol{Y}\boldsymbol{\Omega}$ where $\boldsymbol{\Omega}$ where $\boldsymbol{\Omega}$ is a $n \times l$ matrix ($l = 2k$) with entries drawn independently from a standard normal distribution. Let $\boldsymbol{Q}^{(t)}$ denote the orthonormal basis for the range of $\boldsymbol{C}^{(t)}$. Then $\mathbb{E}\left[\|\boldsymbol{Y} - \boldsymbol{Q}^{(t)}\boldsymbol{Q}^{(t)^T}\boldsymbol{Y}\right]\| \leq (1+\alpha)^{\frac{1}{2t+1}}\sigma_{k+1}$ (Corollary 10.10 of Halko *et al.*, 2009). Here σ_{k+1} is the $(k+1)^{st}$ largest singular value of $\boldsymbol{Y}$ and α is a constant that depends on the m, n and k.

A.8 Exploring the contribution of the Mailman algorithm to scalability

To explore the contribution of the Mailman algorithm to the scalability, we explored variants of the EM algorithm underlying ProPCA that differ in the implementation of the core genotype matrix-vector multiplication. In addition to the Mailman algorithm for genotype matrix-vector multiplication (EM-Mailman), we considered an implementation where the genotypes are stored as a matrix of doubles using the Eigen matrix library [180] (EM1) as well as another implementation where the genotypes are stored in a compact representation in which each genotype is represented using two bits (EM2). The representation in EM2 is expected to be memory-efficient relative to EM1. However, since EM1 represents genotypes directly as a matrix object in Eigen, we expect EM1 to be computationally more efficient. Figure A.12 supports this intuition. EM1 could only be applied to sample sizes of up to $70,000$ before reaching our memory limit. While EM2 can run sample sizes up to $1,000,000$, it is more than two orders of magnitude slower than EM-Mailman. While EM1 is substantially faster, EM-Mailman is about three times faster. We expect that, even if memory were not a constraint, the Mailman algorithm would remain faster than the basic EM algorithm. We note that the Mailman algorithm is only 3-4 times faster than the basic EM algorithm instead of the log factor predicted by theory. We suspect that a reason for this gap is that the Mailman algorithm, as implemented, has not been optimized for specific computing architectures unlike standard matrix algorithms.

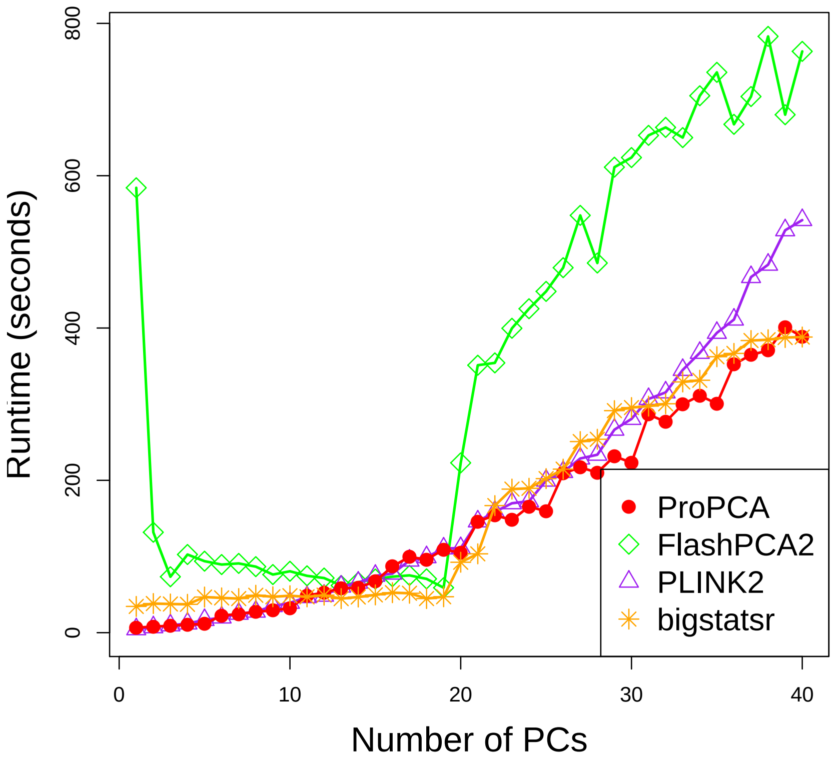


Figure A.1: **ProPCA is efficient at computing large numbers of PCs.** Comparison of methods when calculating differing numbers of principal components. We computed principal components ranging from 1-40 on a dataset containing 20 populations separated at $F_{st} = 0.01$, $10,000$ individuals, and $50,000$ SNPs. All methods were run with default settings.

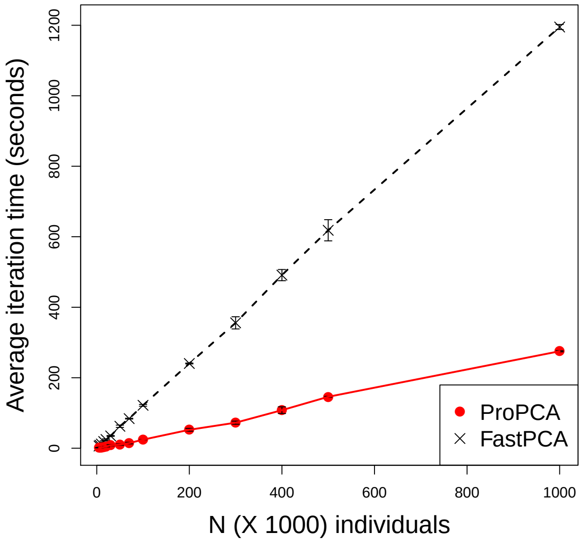


Figure A.2: **ProPCA has faster per-iteration runtimes versus FastPCA**: Comparison of average per-iteration runtimes over simulated genotype data containing $100,000$ SNPs, six subpopulations, $F_{st} = 0.10$ and individuals varying from $10,000$ to $1,000,00$. We were unable to leverage the source code for FlashPCA2 for this comparison.

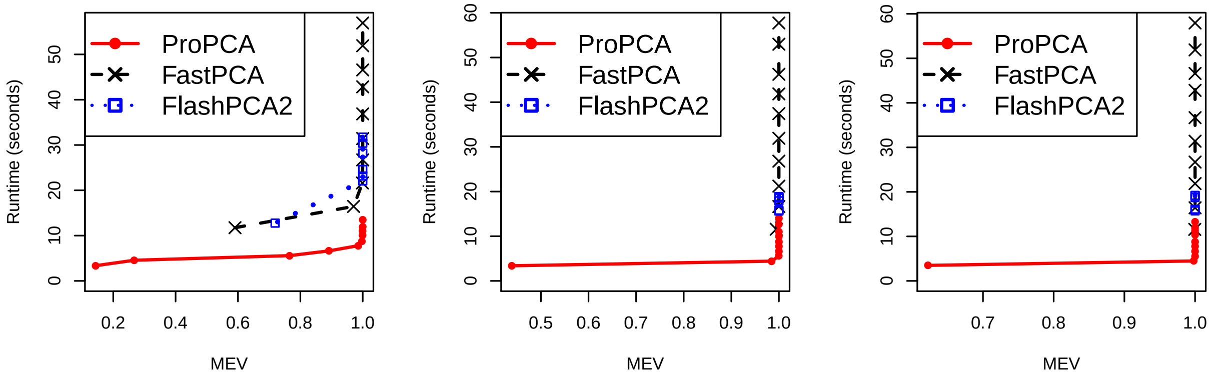
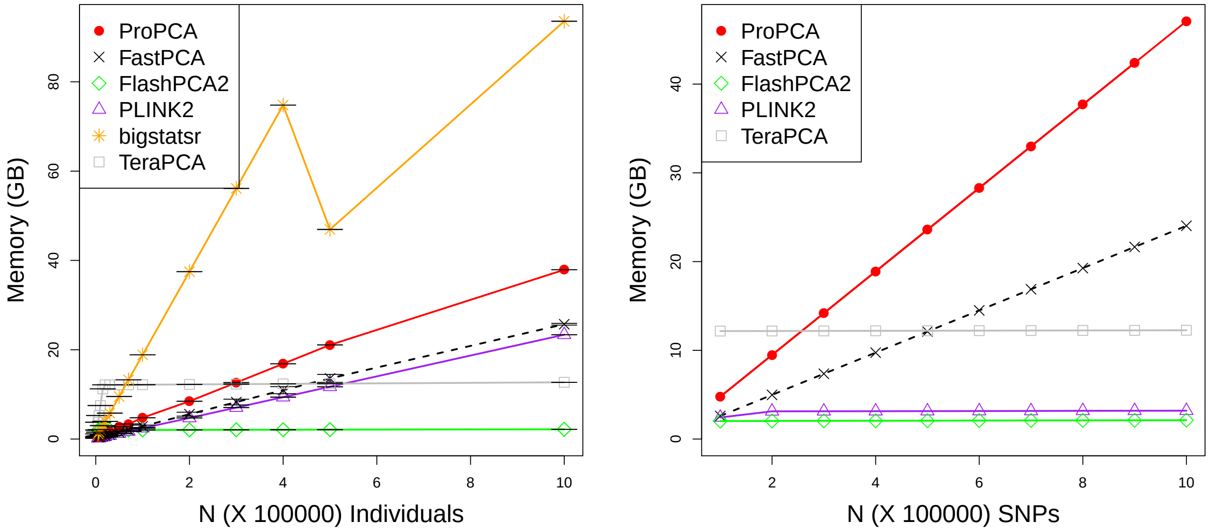


Figure A.3: **ProPCA is computationally efficient relative to other methods.** We compute the total time taken to estimate the top five principal components as a function of a measure of accuracy (MEV) for ProPCA compared to FastPCA and FlashPCA2. We performed these comparisons on simulated genotype data containing $50,000$ SNPs, $10,000$ individuals, six subpopulations, and $F_{st} \in \{0.001, 0.005, 0.01\}$.



(a) (b)

Figure A.4: **ProPCA memory usage scales linearly.** We display the memory usage in gigabytes of each method when computing the top 5 principal components. Figure A.4a show the average memory usage from each method over 10 runs on a dataset containing six populations separated at $F_{ST} = 0.01$, $100,000$ SNPs, and individuals varying from 100,000 to 1,000,000. Figure A.4b shows a similar result, but with $100,000$ individuals and SNPs varying from 100,000 to 1,000,000 over a single run. All methods were run using default settings. We were unable to run bigstatsr for the SNPs experiment due to a bug that causes the method to crash in the presence of monomorphic SNPs.

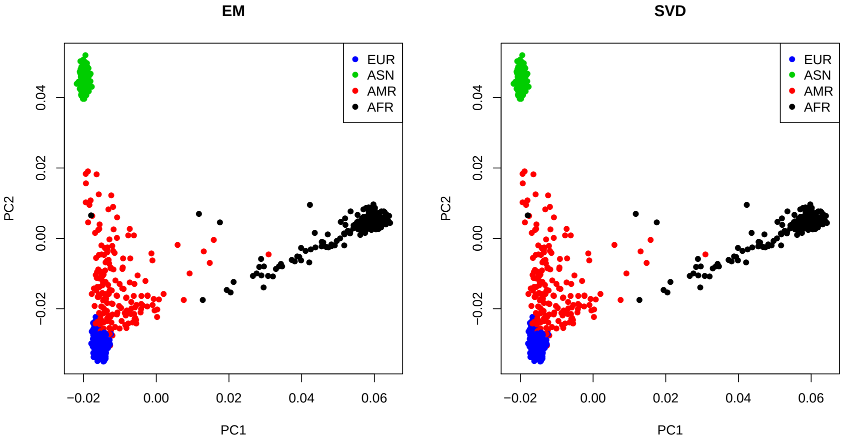


Figure A.5: **ProPCA estimates principal components that are qualitatively indistinguishable from a full SVD on 1000 Genomes Phase 1 data.** We applied our method to genotype data from Phase 1 of the 1000 Genomes project. On a dataset of $1,092$ individuals and $442,350$ SNPs, ProPCA computes the top five PCs in about 17 seconds on a single core. The top two PCs computed by ProPCA and by running SVD on this data set are qualitatively indistinguishable. EM refers to ProPCA.

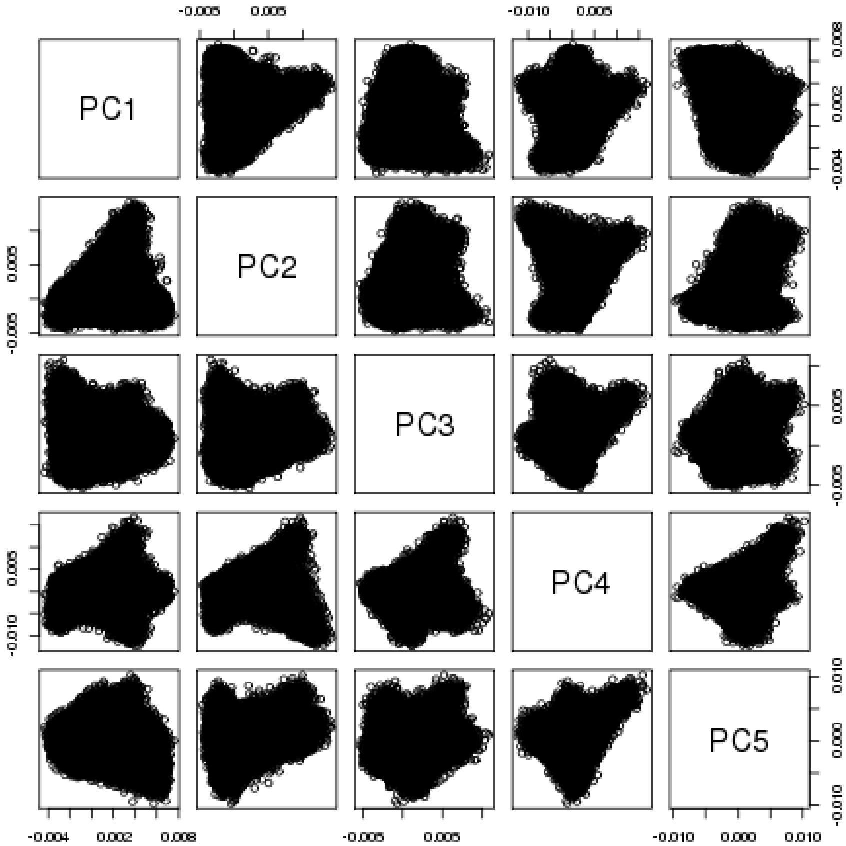


Figure A.6: **Scatterplot pairs between the projections of the first five principal components of the unrelated White British**: Plotting pairs of the first five principal components reveals structure amongst the unrelated White British. This structure diminishes as we increase the number principal components used.

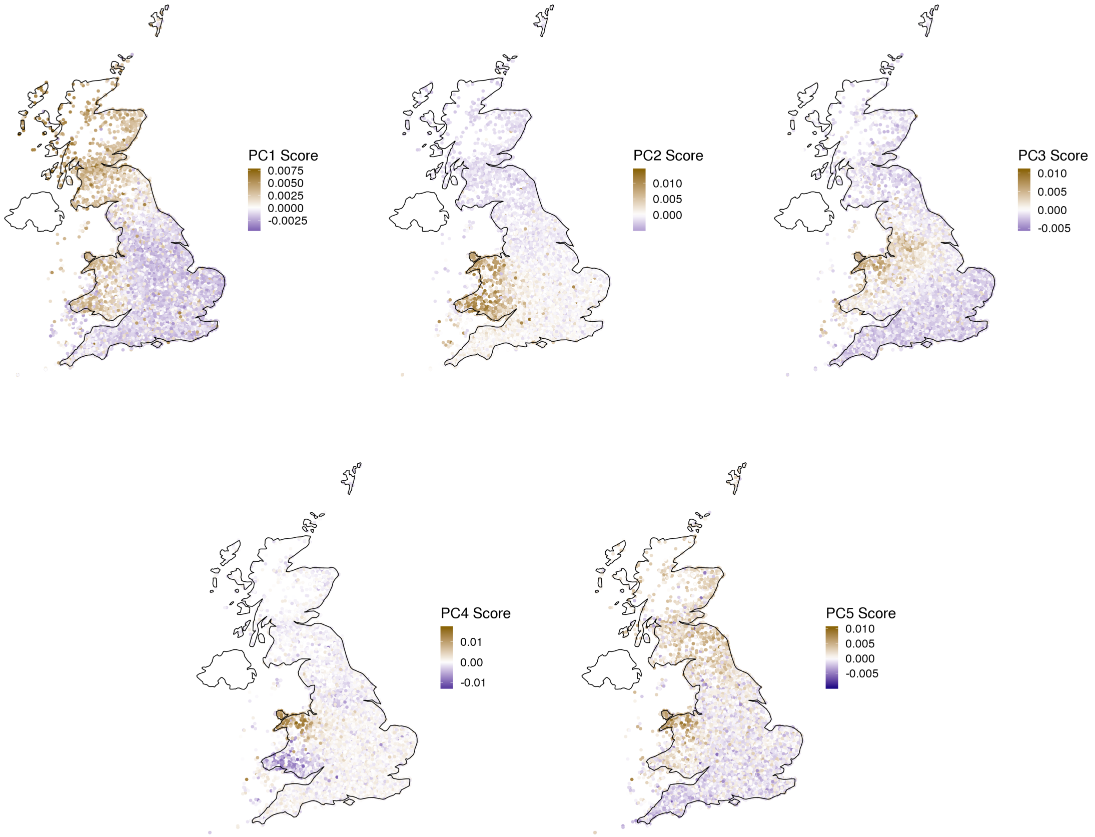


Figure A.7: **Principal component scores of the unrelated White British overlaid on a map of the UK**: Using birth location data available in the UK Biobank, we placed a scatter plot colored by principal component score to reveal geographic variation captured by the principal components.

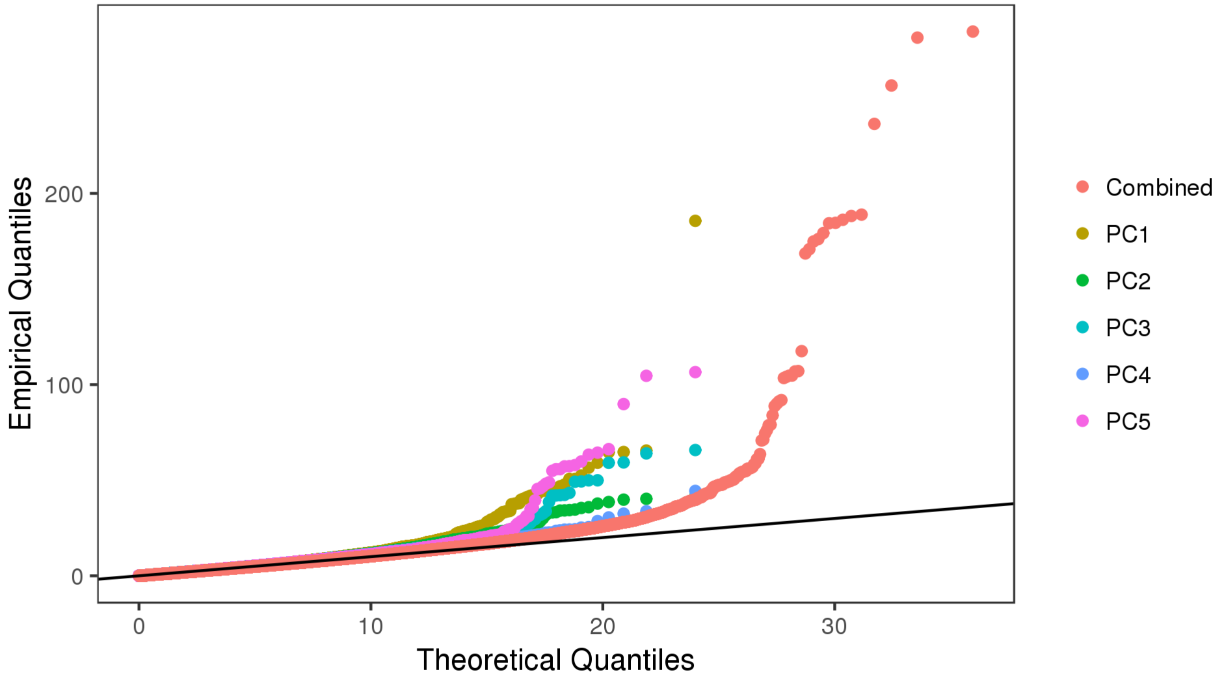


Figure A.8: **The selection statistic is calibrated in the unrelated White British**: We plot the theoretical quantiles of the χ_1^2 distribution against each of the empirical quantiles observed from the first five principal components. All five principal components follow the theoretical distribution well until the upper tail. We additionally show the calibration of the combined statistic against the theoretical quantiles of the χ_5^2 distribution.

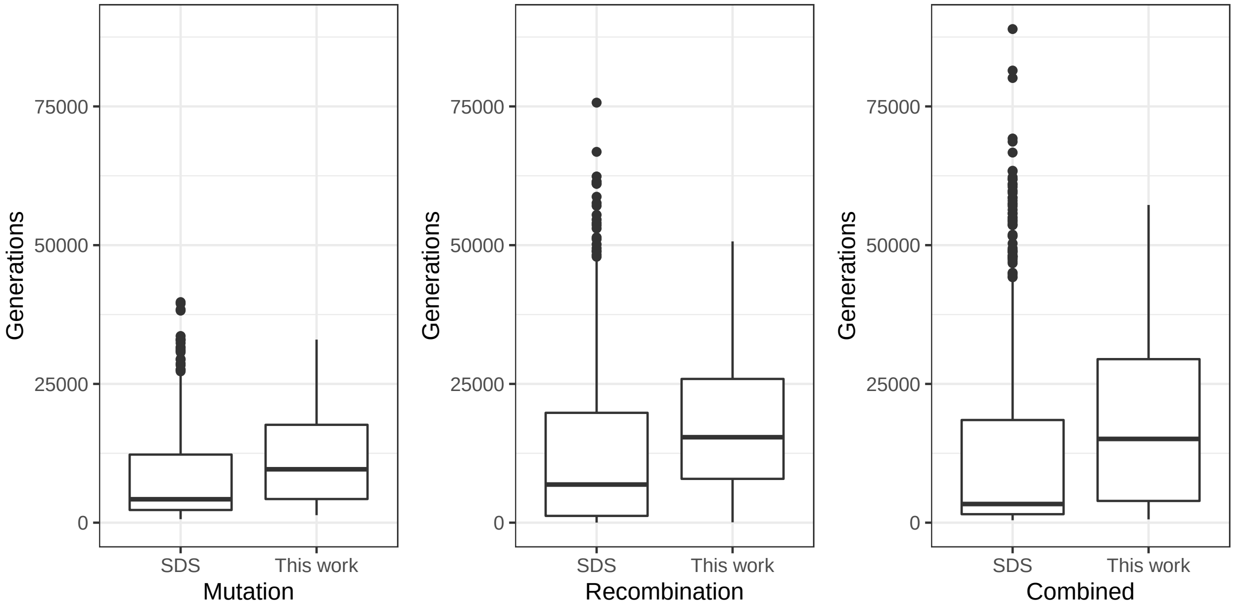


Figure A.9: **Boxplot of estimated allelic ages of putative signals of selection**: Using allelic age estimates from the Human Genome Dating Atlas of Variant Age, we compared the estimated allelic ages of the significant signals of selection in Field et al. 2016 (SDS score > 4) and signals found by our own selection statistic. The x-axis denotes different clock models used to estimate allelic ages while allelic age estimates are denoted in generations on the y-axis. The joint clock model estimates allelic age using information from both the recombination and mutational clock models.

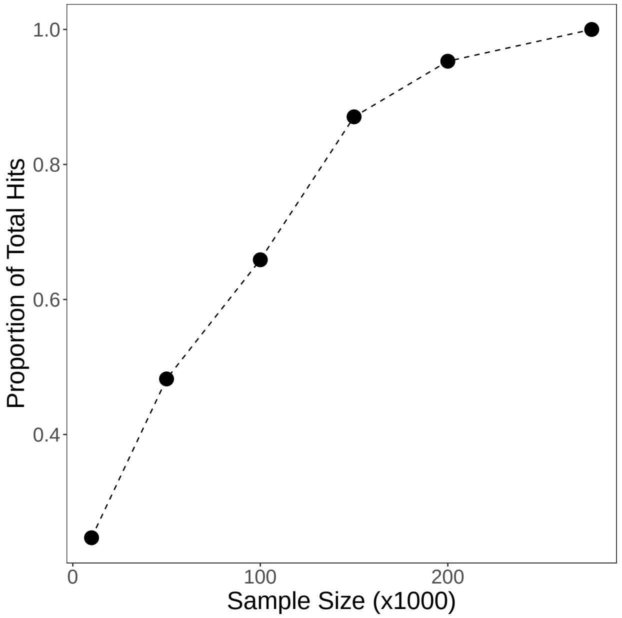
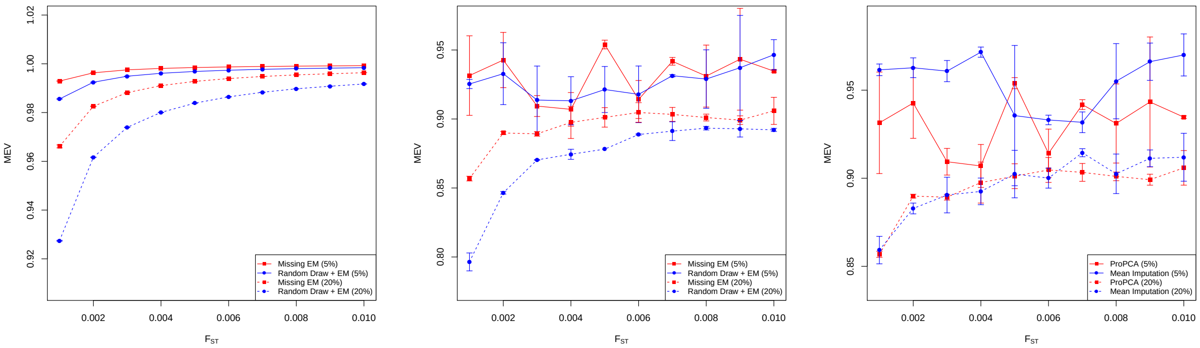
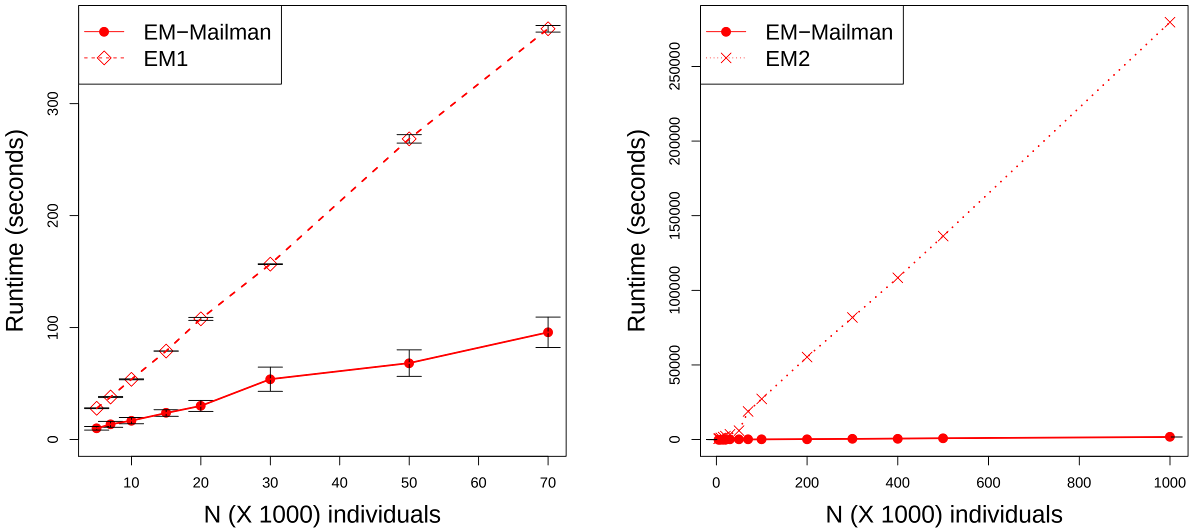


Figure A.10: **Proportion of total number of selection signal hits as a function of sample size**: To further illustrate the importance of large sample sizes for biological discovery, we analyzed how many selection signals we could discover as a function of sample size. We randomly subsampled 10,000, 50,000, 100,000, and 200,000 individuals from the White British populations and performed our selection scan. The x-axis denotes sample size in thousands and the y-axis denotes the proportions of total hits discovered.



(a) (b) (c)

Figure A.11: **ProPCA infers more accurate principal components (PCs) in the presence of missing data compared to imputed genotypes**: We compared the MEV from eigenvectors calculated from both modes of ProPCA with ground truth eigenvectors from performing a full SVD. We evaluated performance at 5% and 20% random missing values at 5 (A.11a) and 10 principal components (A.11b). We additionally compared ProPCA to mean imputation followed by a full SVD (A.11c). The data consists of simulated genotype data of $50,000$ SNPs from $10,000$ individuals from 5 populations for 5 PCs and 10 populations for 10 PCs separated by a range of F_{st} values. This process was repeated ten times to measure variability. Error bars denoting one standard deviation are shown for each point.



(a)

(b)

Figure A.12: **The Mailman matrix-vector multiplication contributes to the scalability of ProPCA**: We compare the time taken to compute the top five principal components by the EM algorithm underlying ProPCA when used in conjunction with the Mailman algorithm and without. We performed these comparisons on simulated genotype data containing 100, 000 SNPs, six subpopulations, $F_{st} = 0.10$ and individuals varying from 10, 000 to 1, 000, 000. Figure A.12a compares the runtime of the EM algorithm with the Mailman matrix-vector multiplication to an EM algorithm where the genotypes are represented as a matrix of doubles (EM1). With this representation, the EM algorithm could only be applied to sample sizes of at most 70, 000 individuals due to memory constraints. Figure A.12b compares the runtime of the EM algorithm with the Mailman matrix-vector multiplication to an EM algorithm where genotypes are represented in a compact representation (EM2).

	PC5	**PC10**	**PC15**	**PC20**	**PC25**	**PC30**	**PC35**	**PC40**
bigsnpr	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
FlashPCA2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
PLINK2	0.9816	1.0000	0.9990	0.9999	0.9981	0.9998	0.9999	1.0000
ProPCA	0.9825	0.9994	0.9932	0.9975	0.9991	0.9999	1.0000	1.0000
TeraPCA	0.9996	0.9943	0.6714	0.5000	0.9247	0.9505	0.9429	0.9548
FastPCA	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	NA	NA

Table A.1: **Comparison of accuracy of methods to estimate principal components on the genotype data from the 1000 Genomes Phase 1 project**. We compared the accuracy of the ProPCA algorithm, bigsnpr, FlashPCA2, PLINK2, TeraPCA, and FastPCA when applied to 1092 individuals in the 1000 Genomes Phase 1 project. We report MEV averaged over ten trials. FastPCA gave us a segmentation fault for estimation of ≥ 35 PCs. We ran all methods using default settings.

	PC1	**PC2**	**PC3**	**PC4**	**PC5**
Longitude	-0.39 (0)	-0.06 (6.29e-188)	-0.18 (0)	0.03 (4.88e-42)	-0.11 (0)
Latitude	0.38 (0)	-0.41 (0)	0.16 (0)	-0.05 (1.06e-141)	0.32 (0)

Table A.2: **Pearson correlation between principal components and birth location coordinates in the unrelated White British.** Pearson correlation between the principal components and birth location coordinates reveals that the principal components unveil geographic variation. P-values from Pearson correlation *t*-test is shown in parentheses on the right.

λ_{GC}	**PC1**	**PC2**	**PC3**	**PC4**	**PC5**	**Combined**
Statistic	0.961	0.970	0.979	0.955	0.958	0.962
Galinsky	1.017	0.904	0.900	0.791	0.794	0.877

Table A.3: **Selection statistics are not substantially inflated.** We calculated the λ_{GC} values for each principal component and the combined statistic to check for inflation. In the unrelated White British set, the calculated values show that our selection statistics are not substantially inflated (top row). Furthermore, we show that the previously related statistic proposed by Galinsky et al. 2016 does not calibrate as well as our statistics based on λ_{GC} values (bottom row).

Table A.4: **Table of significant SNPs found by selection scan on unrelated White British.** Our selection scan on the unrelated White British population resulted in 59 significant SNPs. Our significance threshold was a Bonferroni corrected $p < 0.05$. Bonferroni corrected p-values are shown.

SNP	**PC1**	**PC2**	**PC3**	**PC4**	**PC5**
rs79907870	1	0.00799562	1	1	1
rs6670894	1	0.01540734	1	1	1
rs7570971	2.64E-09	1	0.05250026	1	1
Affx-17900027	4.70E-05	1	0.94915971	1	1
rs1375132	0.00063193	1	0.92775578	1	1
rs1446585	1.70E-07	1	0.00158508	1	1
rs2322659	0.00089443	1	0.22033712	1	1
rs2236783	3.40E-06	1	0.03538101	1	1
rs309125	3.38E-06	1	0.01891111	1	1
rs6716536	2.43E-05	1	0.05178926	1	1
rs13131593	1	1	1	1	4.93E-05
rs7660102	1	1	1	1	2.39E-05
rs11729638	1	1	0.16162614	1	8.50E-06
rs6853255	1	1	0.70035187	1	0.00103318
Affx-35294751	0.00073162	1	0.00026216	1	2.53E-07
rs10776483	0.00046787	0.02786821	0.00024183	1	3.89E-07
rs11096955	0.00030116	1	4.18E-08	1	1.14E-07
rs11096956	0.000265	0.01418271	0.00029435	1	2.50E-07
rs11096957	0.00038658	1	4.61E-08	1	1.21E-07
rs73236616	7.69E-05	0.80738365	4.96E-06	1	3.14E-09
rs5743614	2.80E-09	0.00085802	3.82E-09	1	1.75E-18
rs4833095	1.80E-09	0.00034064	1.56E-09	1	4.64E-18
rs5743566	8.29E-05	1	6.60E-06	1	5.37E-09
rs5743560	7.09E-05	1	7.21E-06	1	1.23E-09
rs5743810	1	1	1	1	1.23E-05
rs6531668	1	1	1	1	4.23E-05
rs73236633	9.99E-05	0.9429865	0.0002819	1	3.26E-08
rs73236649	0.00022931	1	0.00014038	1	8.51E-08
rs6851685	4.57E-08	0.0160796	4.90E-06	1	8.14E-15
rs4833106	1	1	1	1	0.01083497
rs112873858	1	1	1	8.29E-05	1
rs79194719	1	0.02475552	1	0.0193468	1
rs77635680	1	0.00068717	1	0.03656091	1
rs7773997	1.34E-06	1	1	1	1
rs3778607	1.53E-05	1	1	1	1
rs872071	4.99E-05	1	1	1	1

Continued on next page

Continued on previous page

SNP	**PC1**	**PC2**	**PC3**	**PC4**	**PC5**
rs9378805	0.00018458	1	1	1	1
rs62389423	8.80E-36	1	1	1	0.00696174
rs1473909	0.00289641	1	1	1	1
rs6918152	0.00291115	1	1	1	1
rs9267810	0.0330364	1	1	1	1
rs2269424	0.02367807	1	1	1	1
rs1061807	0.02433184	1	1	1	1
rs9267817	0.0191211	1	1	1	1
rs1035798	0.04542733	1	1	1	1
rs12380860	1	0.02669599	1	1	1
rs1129038	1	0.00245893	1	1	1
rs12913832	1	0.00161319	1	1	1
rs62048361	1	0.01231594	1	1	1
rs61747071	1	0.00648571	1	1	1
rs516246	0.00267139	1	1	1	1
rs492602	0.00217383	1	1	1	1
rs681343	0.00231481	1	1	1	1
rs601338	0.00234875	1	1	1	1
rs602662	0.01882034	1	1	1	1
rs485186	0.02395844	1	1	1	1
rs504963	0.01839202	1	1	1	1
rs503279	0.0163532	1	1	1	1
rs676388	0.04348389	1	1	1	1

CHR	**POS**	**rsid**	**Gene**	**PC1**	**Other Gene Hits in Window**
2	135837906	rs7570971	RAB3GAP1	2.64E-09	RAB3GAP1,R3HDM1,LCT
4	38799710	rs4833095	TLR1	1.80E-09	TLR10,TLR1,TLR6,FAM114A1
6	421281	rs62389423		8.80E-36	IRF4,EXOC2
6	32139813	rs9267817		0.019121	HLA
19	49206417	rs492602	FUT2	0.002174	FUT2
CHR	**POS**	**rsid**	**Gene**	**PC2**	**Other Gene Hits in Window**
1	2240074	rs79907870	SKI	0.007996	SKI
1	116977051	rs6670894		0.015407	
4	38799710	rs4833095	TLR1	0.000341	TLR10,TLR1,FAM114A1
5	164861910	rs77635680		0.000687	
9	13954710	rs12380860		0.026696	
15	28365618	rs12913832	HERC2	0.001613	HERC2
16	53720436	rs61747071	RPGRIP1L	0.006486	RPGRIP1L
CHR	**POS**	**rsid**	**Gene**	**PC3**	**Other Gene Hits in Window**
2	136407479	rs1446585	R3HDM1	0.001585	R3HDM1,LCT
4	38799710	rs4833095	TLR1	1.56E-09	TLR10,TLR1,TLR6,FAM114A1
CHR	**POS**	**rsid**	**Gene**	**PC4**	**Other Gene Hits in Window**
4	89323743	rs112873858	HERC6	8.29E-05	HERC6
5	164847509	rs79194719		0.019347	
CHR	**POS**	**rsid**	**Gene**	**PC5**	**Other Gene Hits in Window**
4	38798935	rs5743614	TLR1	1.75E-18	TLR10,TLR1,TLR6,FAM114A1
6	421281	rs62389423		0.006962	

Table A.5: **Principal component selection scan reveals 12 unique loci under selection across the top five principal components.** We obtained 59 selection hits across the first five principal components of the unrelated White British subset of the UK Biobank. We clustered these hits into 12 unique loci by aggregating all significant hits into 1 Mb windows centered around the most significant hits. Other genes with significant hits that are within the 1 Mb window are listed in the last column.

Region	**rs79907870**	**rs6670894**	**rs9856661**	**rs112873858**	**rs116352364**	**rs77635680**	**rs118079376**	**rs12380860**	**rs4986790**	**rs12913832**	**rs61747071**
Cardiff and Vale of Glamorgan	1.45E-05	0.1969244	1	1	1	0.01750364	1	0.00700611	0.41743406	5.48E-08	3.95E-08
Central Valleys	3.70E-08	1.09E-12	1	1	1	9.32E-05	1	4.86E-14	0.05081134	4.46E-07	1.96E-09
Gwent Valleys	9.68E-05	1	1	1	1	1	1	1.06E-09	1	0.00194891	1.67E-11
Bridgend and Neath Port Talbot	5.05E-09	2.15E-06	1	1	1	0.42357784	1	6.56E-05	1	0.0187777	8.11E-07
South West Wales	8.00E-07	4.27E-07	1	1	1	0.00616006	1	2.09E-07	1	6.69E-06	3.64E-11
Flintshire and Wrexham	0.00857823	1	0.00108963	0.08038258	0.00050635	9.87E-16	5.11E-07	0.0284304	1	1	1
Swansea	6.66E-06	7.73E-06	1	1	0.6076096	0.01796713	1	0.00011823	1	2.27E-07	8.98E-05
Conwy and Denbighshire	3.98E-05	1	1	2.72E-08	3.14E-05	6.35E-10	0.07236678	1	1	1	1
Wirral	1	1	1	9.19E-06	0.01890679	0.21626338	1	1	1	1	1
Gwynedd	1	1	0.15212484	1	1	1.70E-07	0.64368483	0.00047818	1	1	1
Calderdale and Kirklees	1	1	1	1	1	0.00013276	1	1	1	1	1
Stoke-on-Trent	1	1	1	1	1	0.01602871	1	1	1	1	1
Shropshire CC	1	1	1	1	1	7.97E-10	1	1	1	1	1
Powys	1	1	1	1	1	1	1	0.00278546	1	1	1
Tyneside	1	1	0.42547449	1	0.86933517	1	1	1	2.80E-07	0.25743043	0.01778222
Liverpool	1	0.59063574	1	1	1	0.40703851	1	1	0.0217454	1	1
Aberdeen City and Aberdeenshire	1	1	1	1	1	1	1	1	0.0138983	1	1
Glasgow City	1	1	1	1	1	1	1	1	5.78E-36	1	1
Bristol, City of	1	1	1	1	1	1	1	1	1.39E-05	0.02274959	1
Bath and North East Somerset, North Somerset and South Gloucestershire	1	1	1	1	1	1	1	1	4.27E-05	0.20396962	1
Angus and Dundee City	1	1	1	1	1	1	1	1	0.00142624	1	1
North Lanarkshire	1	1	0.71591595	1	1	1	1	1	0.00874928	1	1
Edinburgh, City of	1	1	1	1	1	1	1	1	3.10E-18	1	1
Perth & Kinross and Stirling	1	1	1	1	1	1	1	1	1.49E-05	1	1
Sefton	1	1	1	1	1	1	1	1	0.00819986	1	1
South Lanarkshire	1	1	1	1	1	1	1	1	0.00037579	1	1
Inverclyde, East Renfrewshire and Renfrewshire	1	1	1	1	1	1	1	1	0.00021506	1	1
Clackmannanshire and Fife	1	1	1	1	1	1	1	1	2.40E-08	1	1
Falkirk	1	1	1	1	1	1	1	1	0.03926723	1	1
Na h-Eileanan Siar (Western Isles)	1	1	1	1	1	1	1	1	0.00188611	1	1

Table A.6: **Allele frequency tests between NUTS3 regions at novel loci confirms differences between geographic regions.** We performed a two-tailed proportion test for our novel loci between the allele frequency in each individual region from the NUTS3 classification of the United Kingdom against the frequency from every other region. We corrected the p-values using the Bonferroni correction (11 loci × 163 regions). The corrected p-values for regions passing the significance threshold are shown in the table.

CHR	POS	rsid	Gene	P	Other Gene Hits in Window	PC1	PC2	PC3	PC4	PC5
1	2240074	rs79907870	SKI	0.046830	SKI	1.27	35.48	4.63	2.15	1.38
2	136407479	rs1446585	R3HDM1	1.66E-14	RAB3GAP1,R3HDM1,UBXN4,LCT	56.55	2.40	38.63	2.12	5.02
3	54077256	rs9856661*		0.020002		0.94	9.18	12.43	0.72	23.45
4	89323743	rs112873858	HERC6	2.18E-05	HERC6	0.02	7.65	2.70	44.40	6.36
4	38799710	rs4833095	TLR1	6.22E-53	TLR10,TLR1,TLR6,FAM114A1	65.50	41.64	65.78	7.13	104.60
5	162948205	rs116352364*		0.000162		0.00	13.68	21.83	16.15	5.27
5	164861910	rs77635680		3.81E-11		2.81	40.26	5.00	32.52	8.16
6	32526736	rs111586361	HLA-DRB5	0.003826	HLA-DRB5	24.43	0.29	11.32	1.25	12.94
6	421281	rs62389423		7.11E-47	IRF4,EXOC2	185.64	13.97	12.09	8.97	35.75
7	38463542	rs118079376*	AMPH	0.000817	AMPH	1.08	0.01	6.36	18.80	27.26
9	120475302	rs4986790*	TLR4	2.35E-05	TLR4	20.37	25.98	1.68	0.67	12.29
15	28365618	rs12913832	HERC2	0.012037	HERC2	0.07	38.60	5.57	3.42	0.16
16	53720436	rs61747071	RPGRIP1L	0.012396	RPGRIP1L	0.76	35.88	4.67	6.39	0.05
19	49206417	rs492602	FUT2	0.000505	FUT2	38.02	12.08	0.33	3.03	1.08

Table A.7: **Combined selection statistic across the top five principal components reveals four additional novel loci.** We discover four additional novel loci using our combined selection statistic from the first five principal components. Loci not found in the individual PC selection statistics are denoted by an asterik in the rsid column. The chi-squared statistic (one degree of freedom) for each principal component is shown in the last five columns of the table.

SNP	Genes in Window	P	Phenotype
rs12913832	HERC2	0	pigment_HAIR_blackmale
		0	pigment_HAIR_blonde
		0	pigment_HAIR_darkbrown
		0	pigment_HAIR
		9.70E-103	pigment_HAIR_red
		0	pigment_SKIN
		1.50E-138	pigment_SUNBURN
		0	pigment_TANNING
rs492602	FUT2	2.50E-09	blood_HIGH_LIGHT_SCATTER_RETICULOCYTE_COUNT
		1.80E-53	blood_MEAN_PLATELET_VOL
		5.20E-11	blood_MEAN_SPHERED_CELL_VOL
		9.70E-18	blood_PLATELET_COUNT
		1.10E-08	body_HEIGHTz
		7.50E-13	bp_DIASTOLICadjMEDz
		1.20E-12	bp_SYSTOLICadjMEDz
		9.40E-19	disease_CARDIOVASCULAR
		2.60E-21	disease_HI_CHOL_SELF_REP
		1.60E-09	disease_HYPERTENSION_DIAGNOSED
		8.80E-12	lung_FEV1FVCzSMOKE
rs62389423	IRF4,EXOC2	6.80E-29	blood_EOSINOPHIL_COUNT
		4.70E-19	blood_LYMPHOCYTE_COUNT
		2.20E-16	blood_WHITE_COUNT
		2.40E-68	body_BALDING1
		2.00E-66	body_BALDING4
		3.20E-31	cancer_ALL
		0	pigment_HAIR_blackmale
		0	pigment_HAIR_blonde
		0	pigment_HAIR_darkbrown
		0	pigment_HAIR
		1.90E-33	pigment_HAIR_red
		0	pigment_SKIN
		0	pigment_SUNBURN
		0	pigment_TANNING
rs7570971	RAB3GAP1,R3HDM1,LCT	1.90E-08	blood_EOSINOPHIL_COUNT
		1.70E-09	blood_RED_COUNT
		2.60E-15	lung_FVCzSMOKE
rs9267817	HLA	2.50E-10	blood_MEAN_PLATELET_VOL
		6.30E-13	blood_MONOCYTE_COUNT
		3.70E-16	blood_RBC_DISTRIB_WIDTH
		1.00E-13	body_HEIGHTz
		7.00E-10	bp_SYSTOLICadjMEDz
		2.40E-13	impedance_BASAL_METABOLIC_RATEz
		6.10E-27	lung_FEV1FVCzSMOKE

Table A.8: **Selection hits are associated with phenotypes in the UK Biobank.** We ran genome-wide association tests for 64 phenotypes in the full release of the UK Biobank for each of our loci. Phenotypes shown reached a *p*-value of genome-wide significance level (0.05×10^{-6}).

SNP	Genes in Window	Phenotype Code	P	Phenotype
rs12913832	HERC2	INI1737	6.61E-24	Childhood_sunburn_occasions
rs492602	FUT2	INI3064	1.28E-09	Peak_expiratory_flow_(PEF)
		INI50	2.00E-11	Standing_height
		HC269	1.55E-14	high_cholesterol
		HC273	4.35E-08	essential_hypertension
		HC357	3.96E-10	duodenal_ulcer
		INI1289	4.42E-09	Cooked_vegetable_intake
		INI20015	1.51E-08	Sitting_height
		INI24019	3.36E-09	Particulate_matter_air_pollution_(pm10);_2007
		HC188	4.85E-17	cholelithiasis/gall_stones
		HC215	8.22E-13	hypertension
		HC225	6.19E-14	cholecystitis
rs5743614	TLR10,TLR1,TLR6,FAM114A1	HC382	4.97E-11	asthma
		HC49	1.18E-14	hayfever/allergic_rhinitis
		INI24019	1.25E-64	Particulate_matter_air_pollution_(pm10);_2007
rs62389423	IRF4,EXOC2	INI30120	1.62E-11	Lymphocyte_count
		INI30150	1.17E-14	Eosinophill_count
		INI30210	4.74E-08	Eosinophill_percentage
		INI50	1.26E-08	Standing_height
		INI134	3.57E-09	Number_of_self-reported_cancers
		INI1737	8.62E-164	Childhood_sunburn_occasions
		INI1873	6.06E-18	Number_of_full_brothers
		INI24004	1.66E-12	Nitrogen_oxides_air_pollution;_2010
		INI24006	7.33E-17	Particulate_matter_air_pollution_(pm2.5);_2010
		INI24017	6.46E-13	Nitrogen_dioxide_air_pollution;_2006
		cancer1003	2.41E-87	skin_cancer
		cancer1060	2.05E-99	non-melanoma_skin_cancer
		FH1001	3.98E-18	Lung_cancer
rs7570971	RAB3GAP1,R3HDM1,LCT	INI3062	1.11E-08	Forced_vital_capacity_(FVC)
		INI23100	1.37E-08	Whole_body_fat_mass
		INI24019	5.36E-12	Particulate_matter_air_pollution_(pm10);_2007
rs9267817	HLA	INI30100	1.36E-10	Mean_platelet_(thrombocyte)_volume
		INI30150	3.16E-16	Eosinophill_count
		INI46	1.31E-09	Hand_grip_strength_(left)
		INI50	1.51E-24	Standing_height
		HC303	5.00E-49	malabsorption/coeliac_disease
		INI20015	1.63E-14	Sitting_height
		INI21002	7.44E-12	Weight
		INI23098	2.31E-11	Weight
		INI24019	1.33E-10	Particulate_matter_air_pollution_(pm10);_2007
		FH1065	7.46E-11	High_blood_pressure
		HC215	3.20E-10	hypertension

Table A.9: **Selection hits are associated with phenotypes from the Global Biobank Engine.** We queried the Global Biobank Engine for associations from our loci. The Global Biobank Engine contains GWAS results for many more phenotypes than those available in the UK Biobank. Phenotypes shown are significant at genome-wide significance level (0.05×10^{-6}).

APPENDIX B

Supplementary Material - Inferring population structure in biobank-scale genomic data

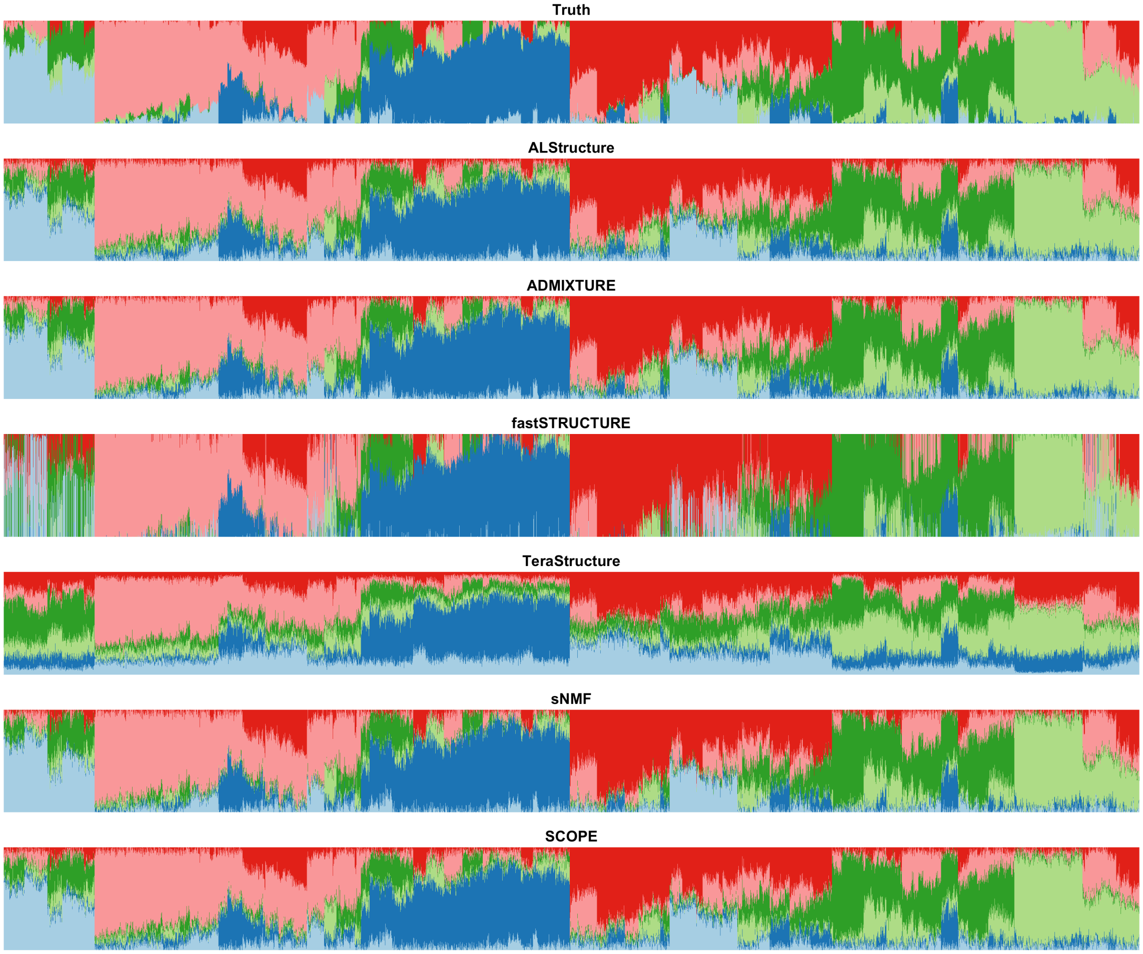


Figure B.1: **Population structure inference for simulations under PSD model generated using Human Genomes Diversity Project data.** PSD model parameters were drawn from HGDP data to generate a simulation dataset with 10,000 samples and 10,000 SNPs. The true admixture proportions and resulting inferred admixture proportions from each method are shown. Colors and order of samples are matched between each method to the truth.

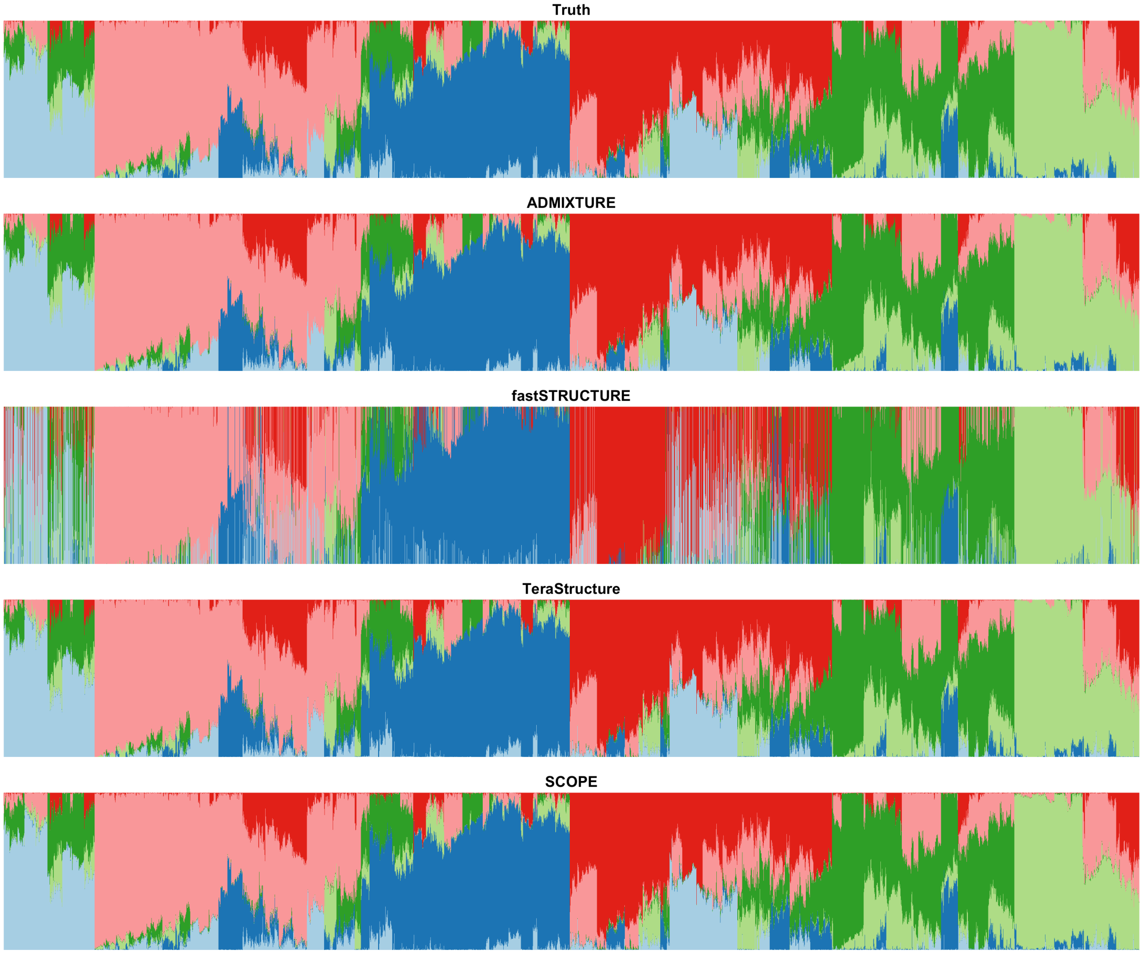


Figure B.2: **Population structure inference for simulations under PSD model generated using 1000 Genomes Phase 3 data.** PSD model parameters were drawn from TGP data to generate a simulation dataset with 10,000 samples and 1 million SNPs. The true admixture proportions and resulting inferred admixture proportions from each method are shown. Colors and order of samples are matched between each method to the truth.

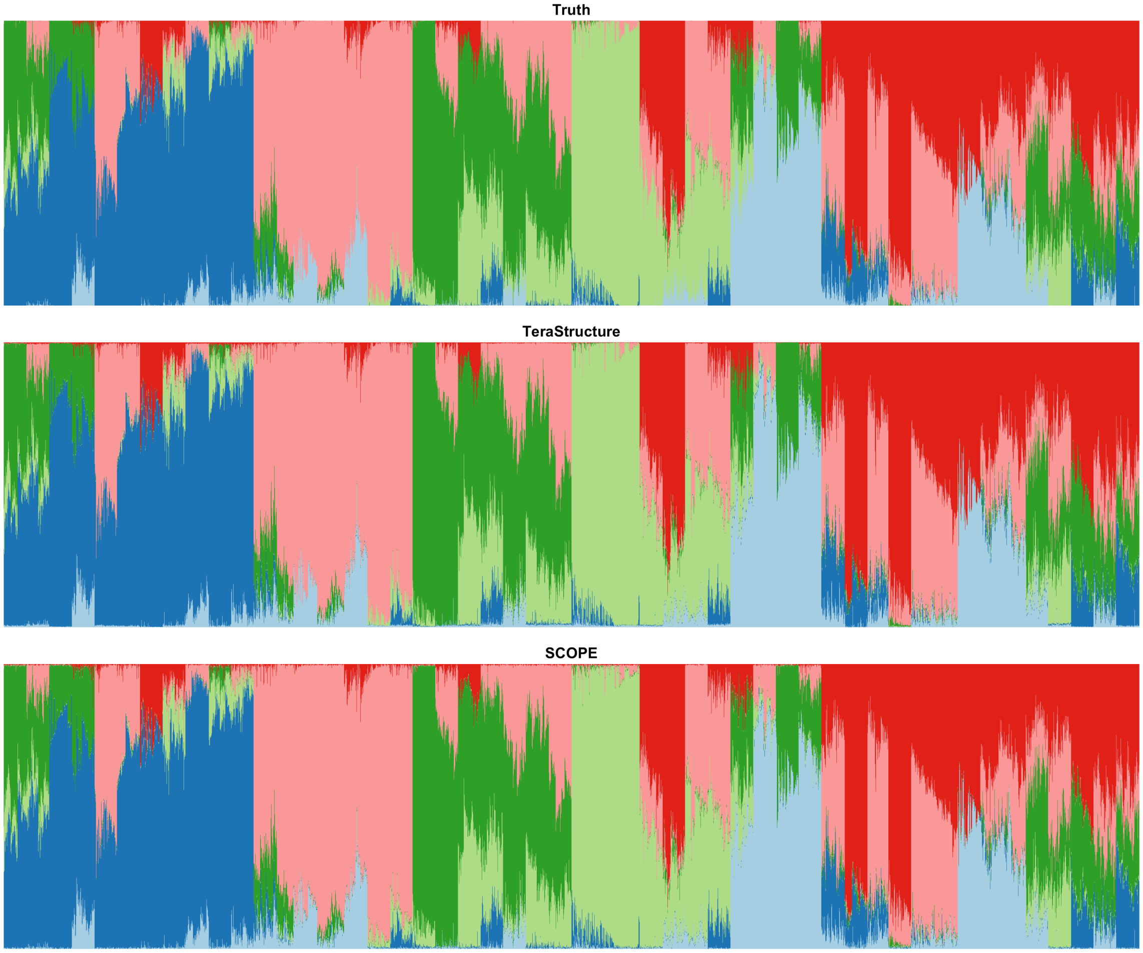


Figure B.3: **Population structure inference for simulations under PSD model generated using 1000 Genomes Phase 3 data.** PSD model parameters were drawn from TGP data to generate a simulation dataset with 100,000 samples and 1 million SNPs. The true admixture proportions and resulting inferred admixture proportions from each method are shown. Colors and order of samples are matched between each method to the truth.

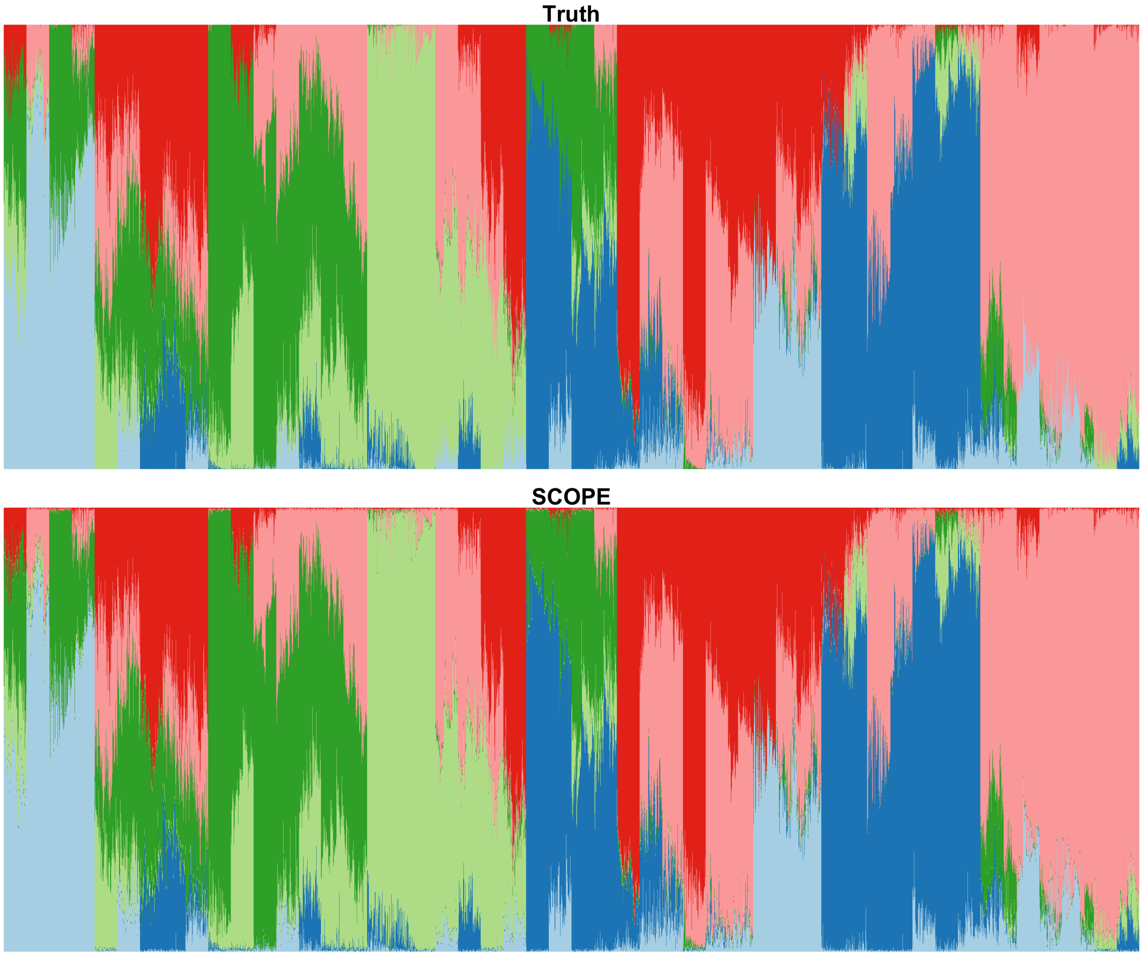


Figure B.4: **Population structure inference for simulations under PSD model generated using 1000 Genomes Phase 3 data.** PSD model parameters were drawn from TGP data to generate a simulation dataset with 1 million samples and SNPs. The true admixture proportions and resulting inferred admixture proportions are shown. Colors and order of samples are matched between SCOPE and the true admixture proportions.

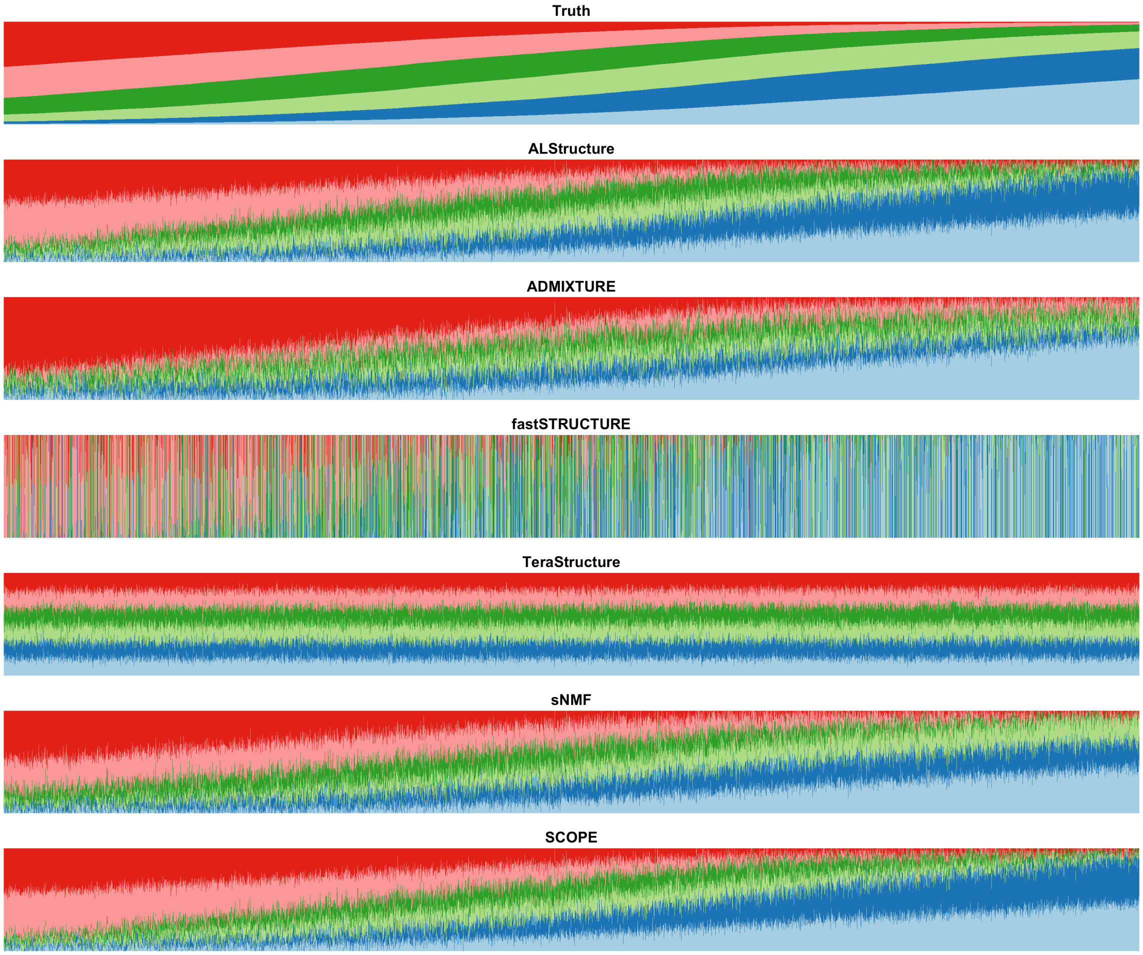


Figure B.5: **Population structure inference for simulations under a spatial model generated using Human Genome Diversity Project data.** Model parameters were drawn from HGDP data to generate a simulation dataset with 10,000 samples and 10,000 SNPs under a spatial model (see Methods). The true admixture proportions and resulting inferred admixture proportions from each method are shown. Colors and order of samples are matched between each method to the truth.

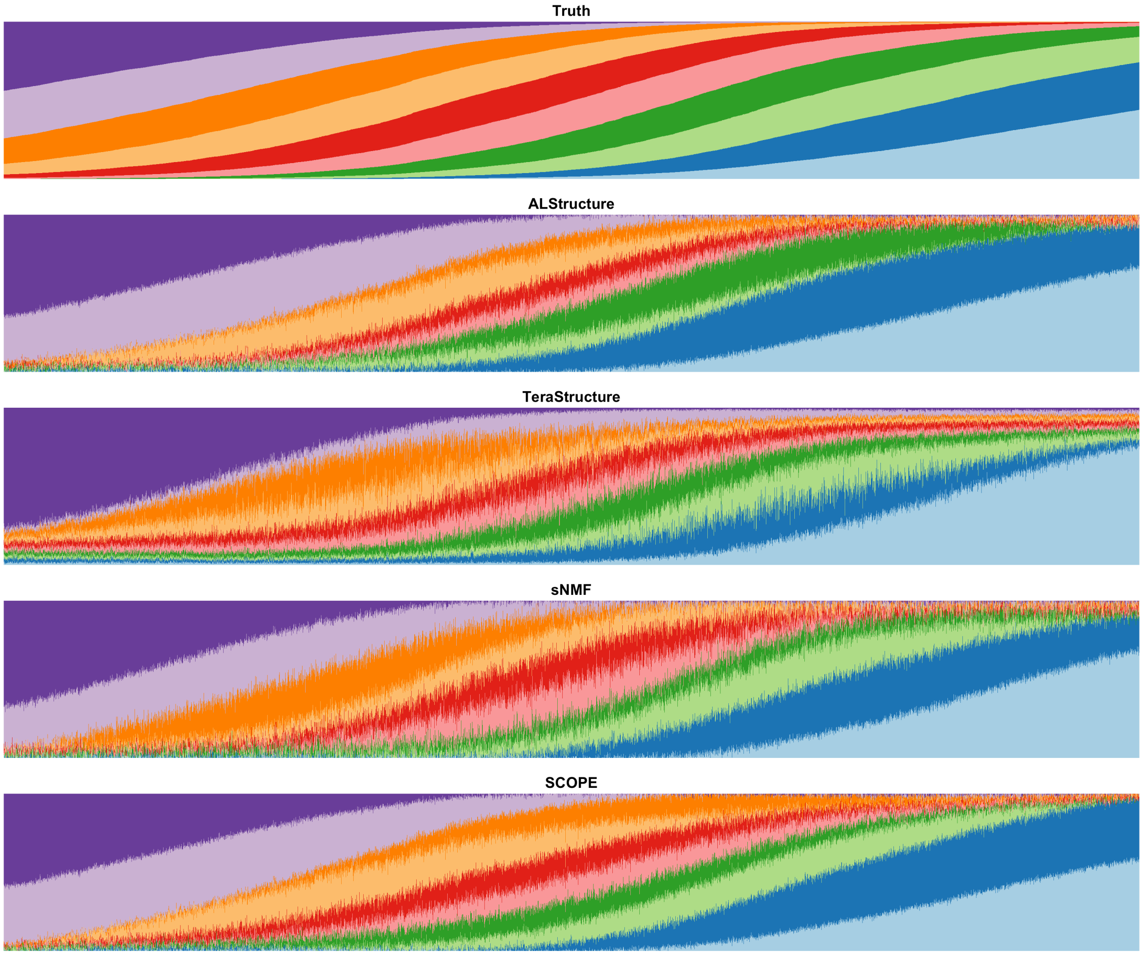


Figure B.6: **Population structure inference for simulations under a spatial model generated using 1000 Genomes Phase 3 data.** Model parameters were drawn from TGP data to generate a simulation dataset with 10,000 samples and 100,000 SNPs under a spatial model (see Methods). The true admixture proportions and resulting inferred admixture proportions from each method are shown. Colors and order of samples are matched between each method to the truth.

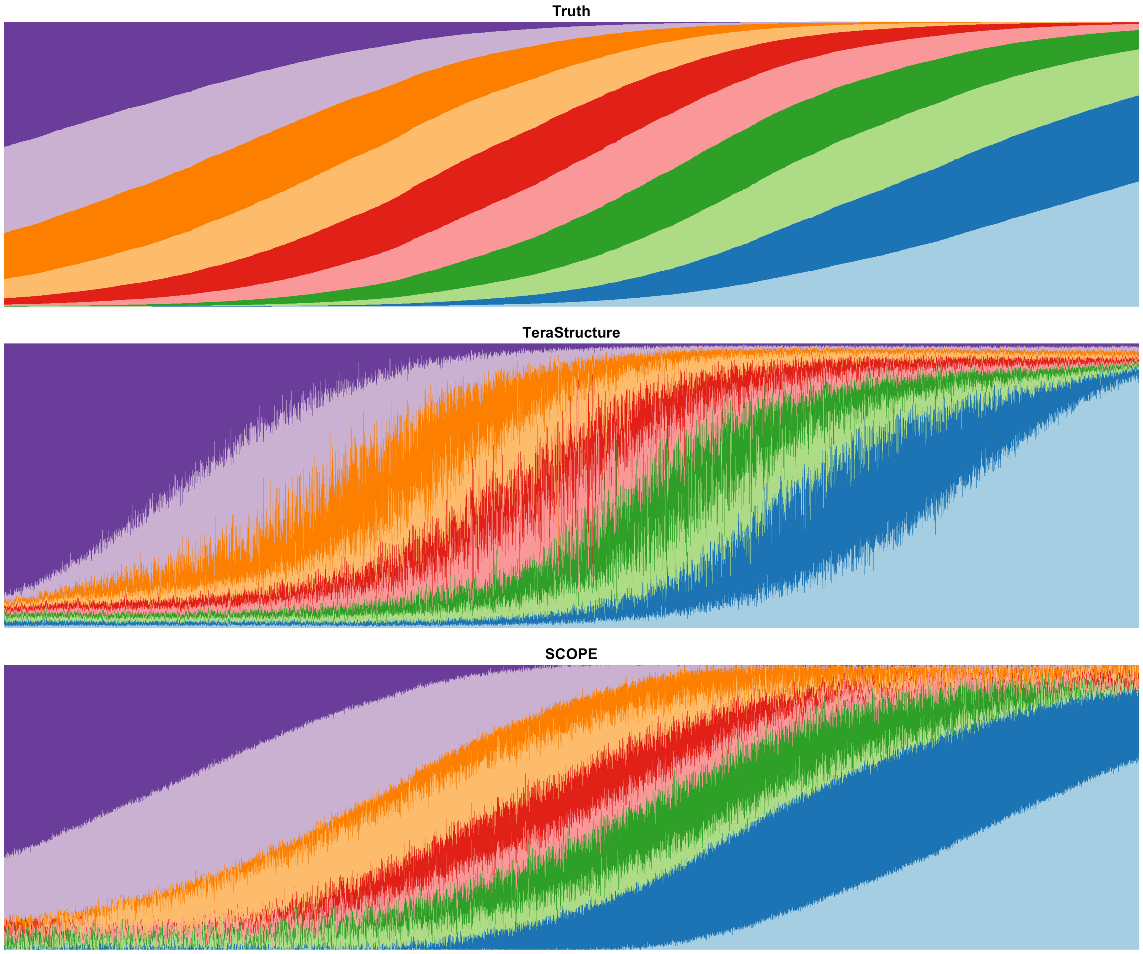
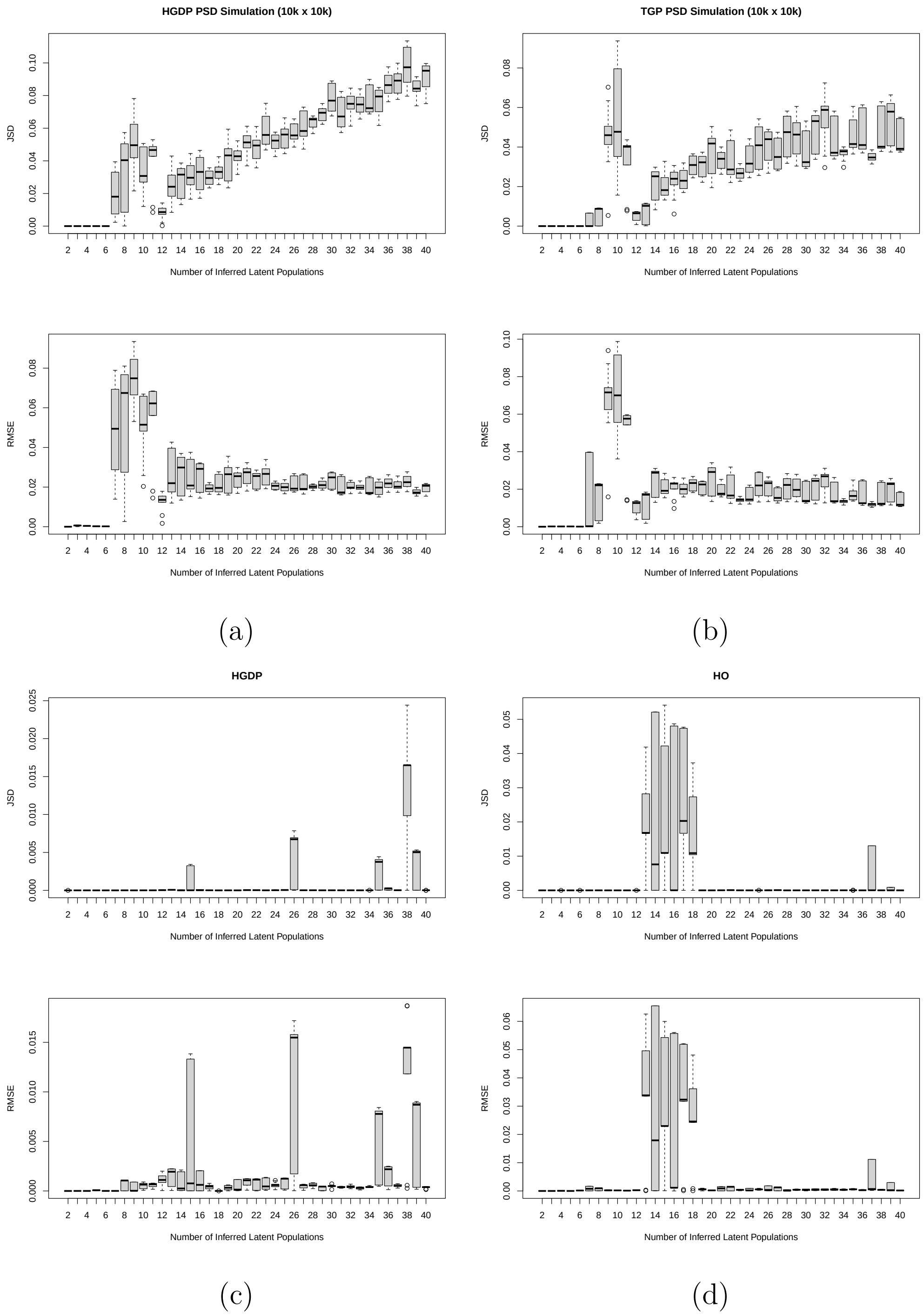


Figure B.7: **Population structure inference for simulations under a spatial model generated using 1000 Genomes Phase 3 data.** Model parameters were drawn from TGP data to generate a simulation dataset with 10,000 samples and 1 millions SNPs under a spatial model (see Methods). The true admixture proportions and resulting inferred admixture proportions from each method are shown. Colors and order of samples are matched between each method to the truth.



(a)

(b)

(c)

(d)

Figure B.8: **Agreement between different runs of SCOPE.** We ran five replicates of SCOPE on our 6 population HGDP PSD simulation (B.8a), our 6 population TGP PSD simulation (B.8b), the HGDP dataset (B.8c), and the HO dataset (B.8d) from 2 to 40 inferred populations. Each boxplot is created from the 10 possible combinations of the five replicates. Jensen-Shannon divergence (top) and root-mean-square error (bottom) are calculated for each of combination.

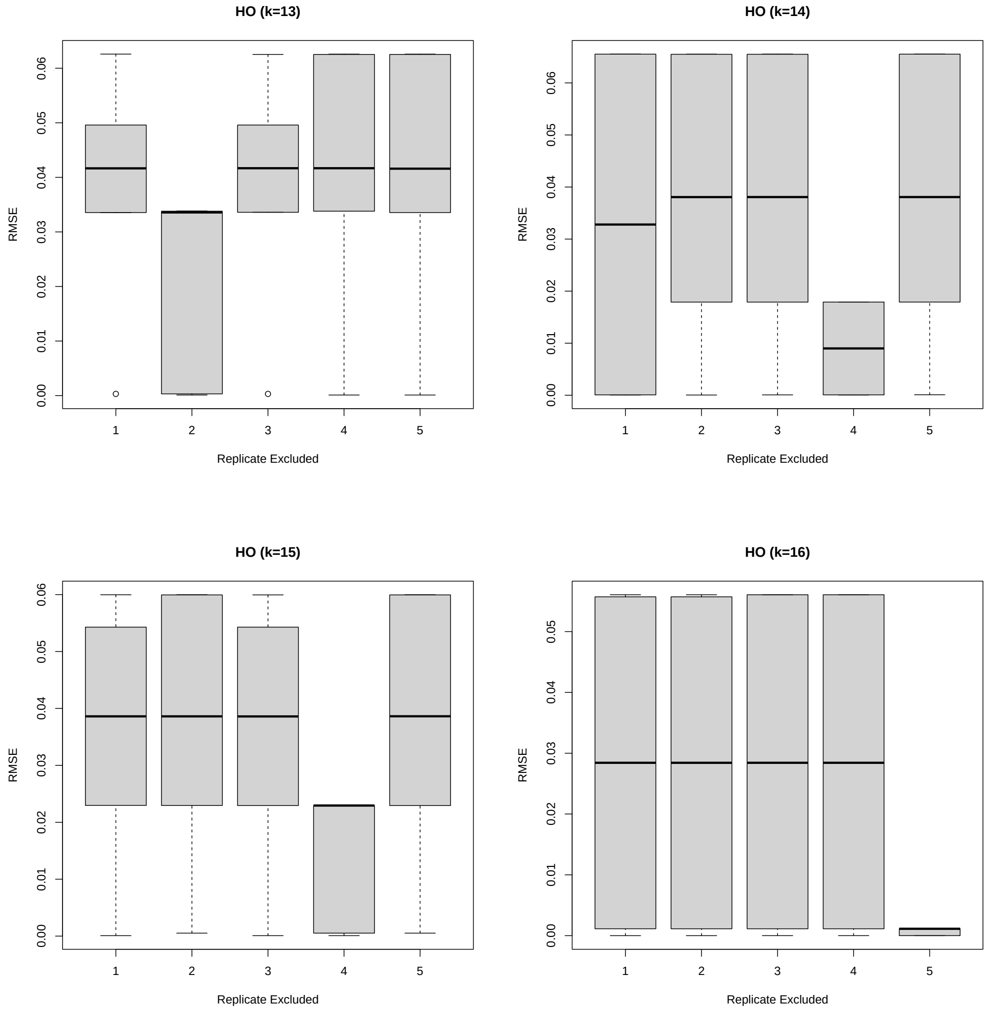
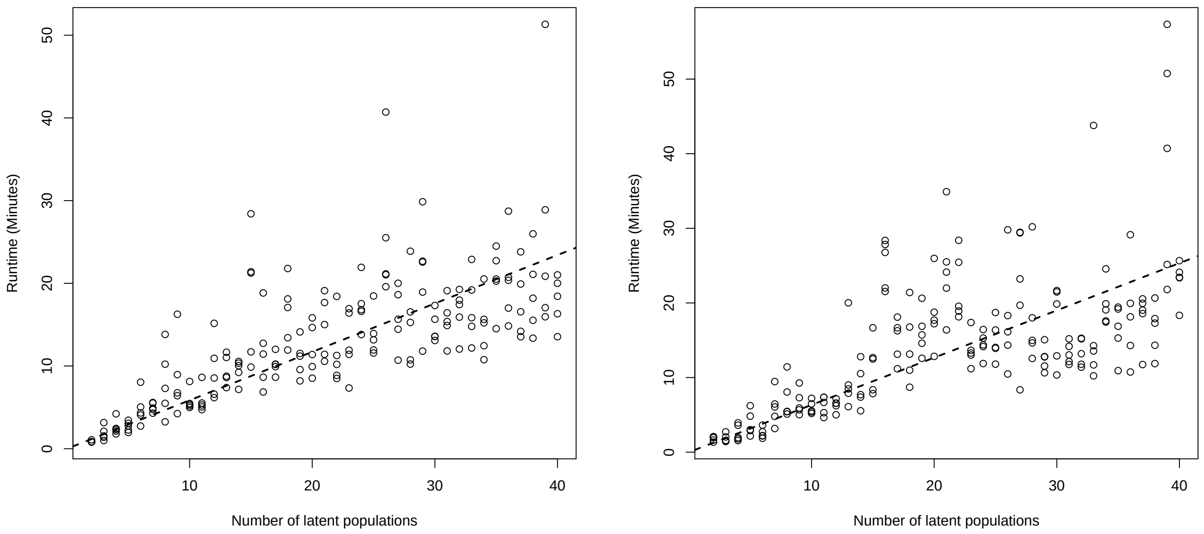


Figure B.9: **Excluding one replicate decreases variability between runs.** We repeated the calculations as in Figure B.8, but excluded one replicate. When excluding one of the five replicates, the variability between different runs of SCOPE decreases.



(a) (b)

Figure B.10: **Runtime scales linearly with increasing number of latent populations.** SCOPE was run on the HGDP (B.10a) and HO (B.10b) datasets with 2 to 40 latent populations (k). We ran five replicates for each value of k. The dashed line represents the least squares estimate for each dataset. Each run of SCOPE was performed using 8 threads.

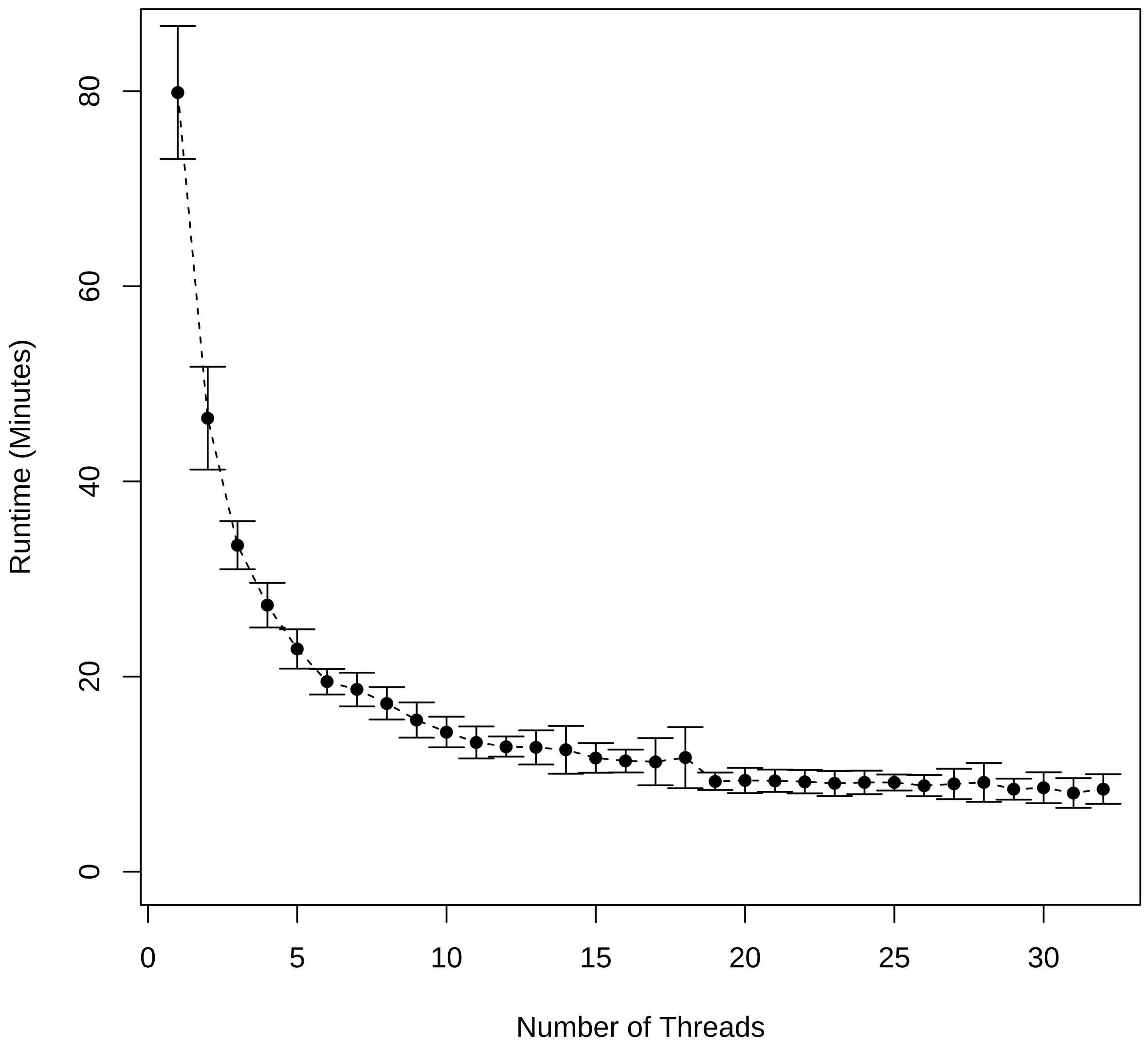


Figure B.11: **Runtime scales sublinearly with number of threads.** SCOPE was run on our PSD simulation dataset with 10,000 individuals, 1 million SNPs, and 6 latent populations. We varied the number of threads used from 1-32 and repeated the experiment 5 times for each number of threads. Means and one standard deviation are shown in the figure.

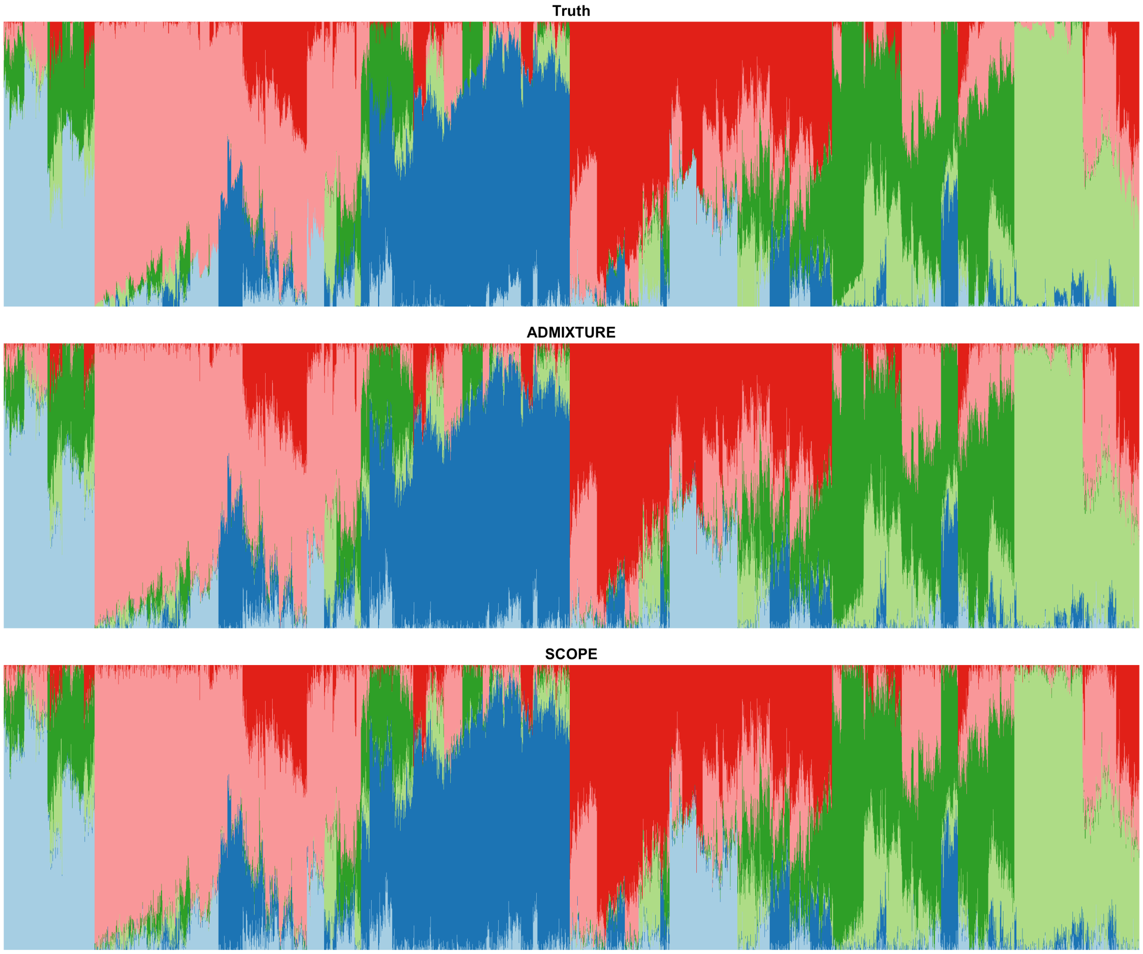


Figure B.12: **Supervised population structure inference for simulations under the PSD model generated using 1000 Genomes Phase 3 data.** PSD model parameters were drawn from TGP data to generate a simulation dataset with 10,000 samples and 10,000 SNPs. Both were methods provided the true population allele frequencies as input. Colors and order of samples are matched between each method to the truth.

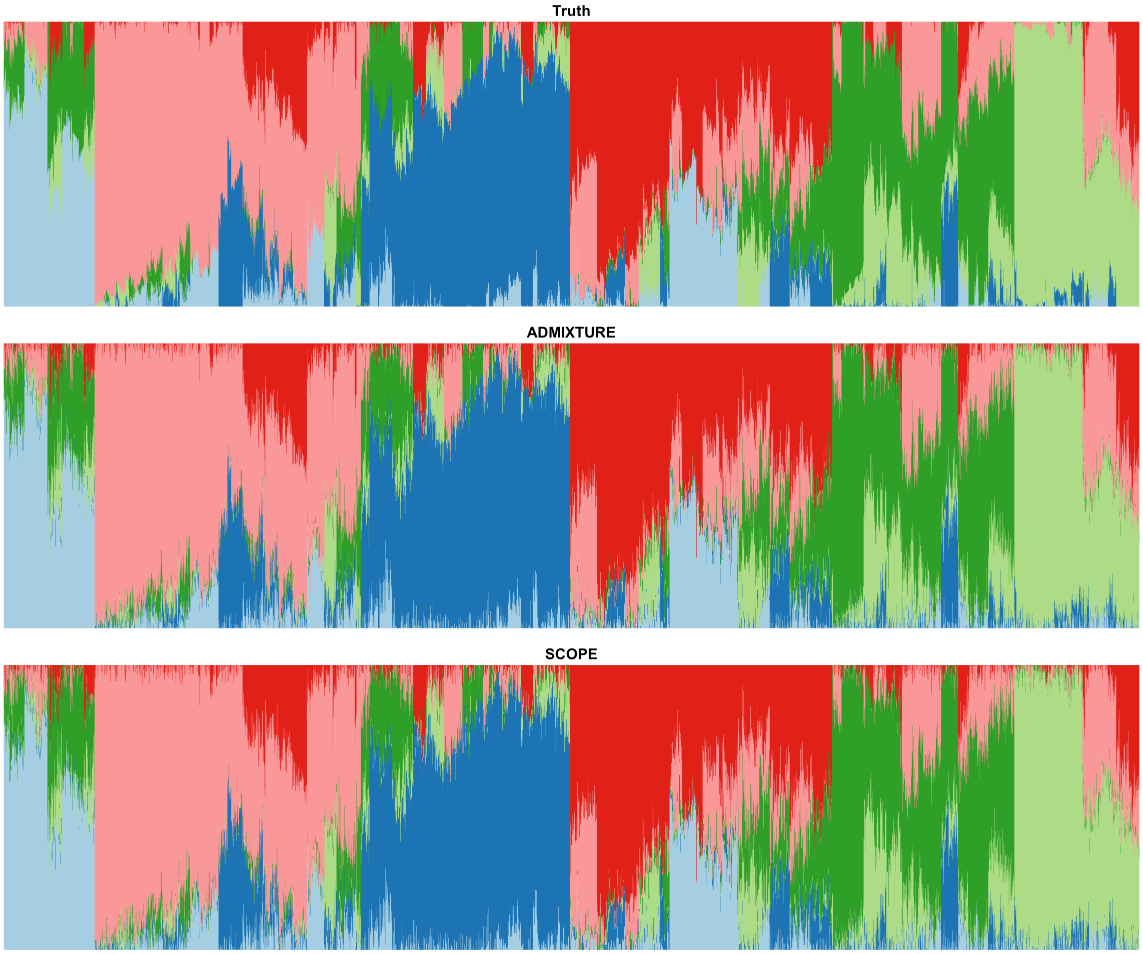


Figure B.13: **Supervised population structure inference for simulations under the PSD model generated using Human Genome Diversity data.** PSD model parameters were drawn from HGDP data to generate a simulation dataset with 10,000 samples and 10,000 SNPs. Both were methods provided the true population allele frequencies as input. Colors and order of samples are matched between each method to the truth.

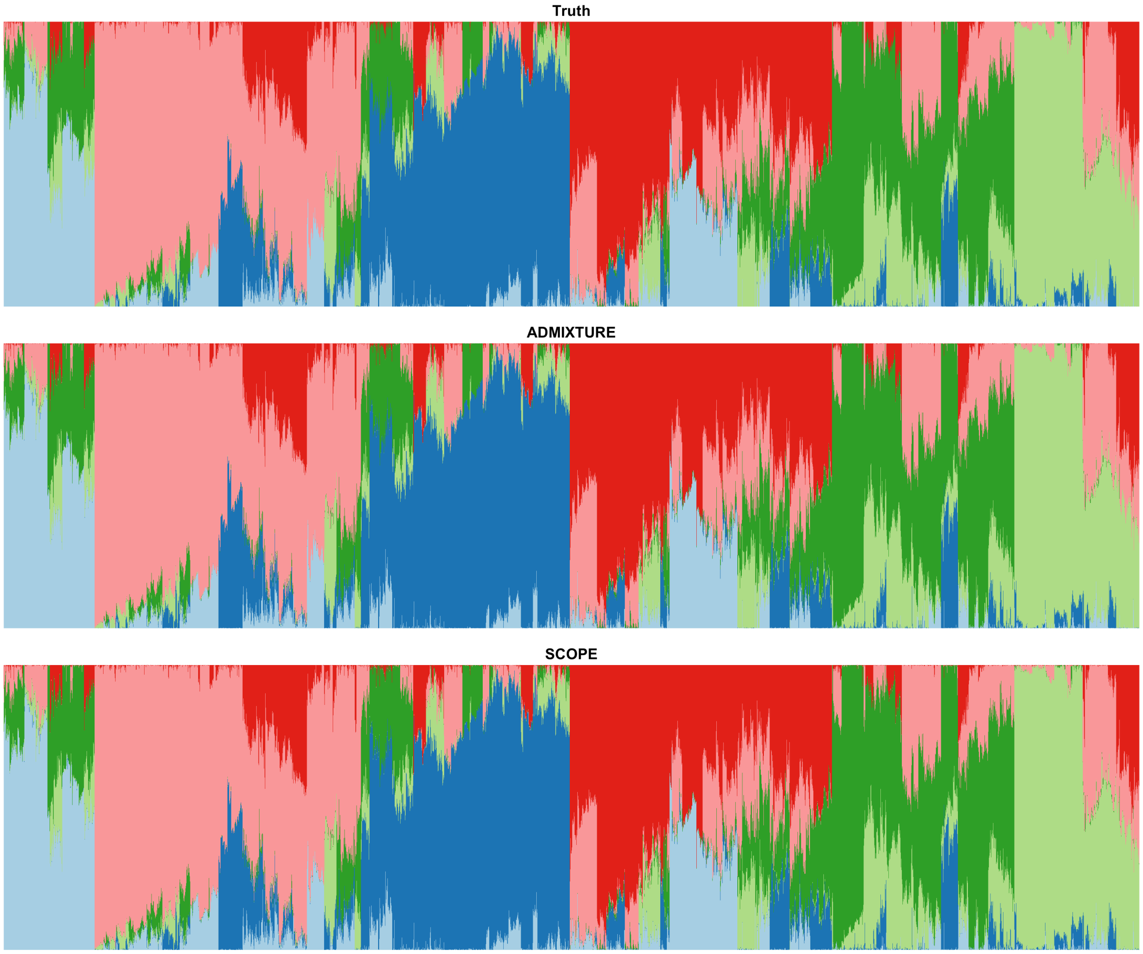


Figure B.14: **Supervised population structure inference for simulations under the PSD model generated using 1000 Genomes Phase 3 data.** PSD model parameters were drawn from TGP data to generate a simulation dataset with 10,000 samples and 1 million SNPs. Both were methods provided the true population allele frequencies as input. Colors and order of samples are matched between each method to the truth.

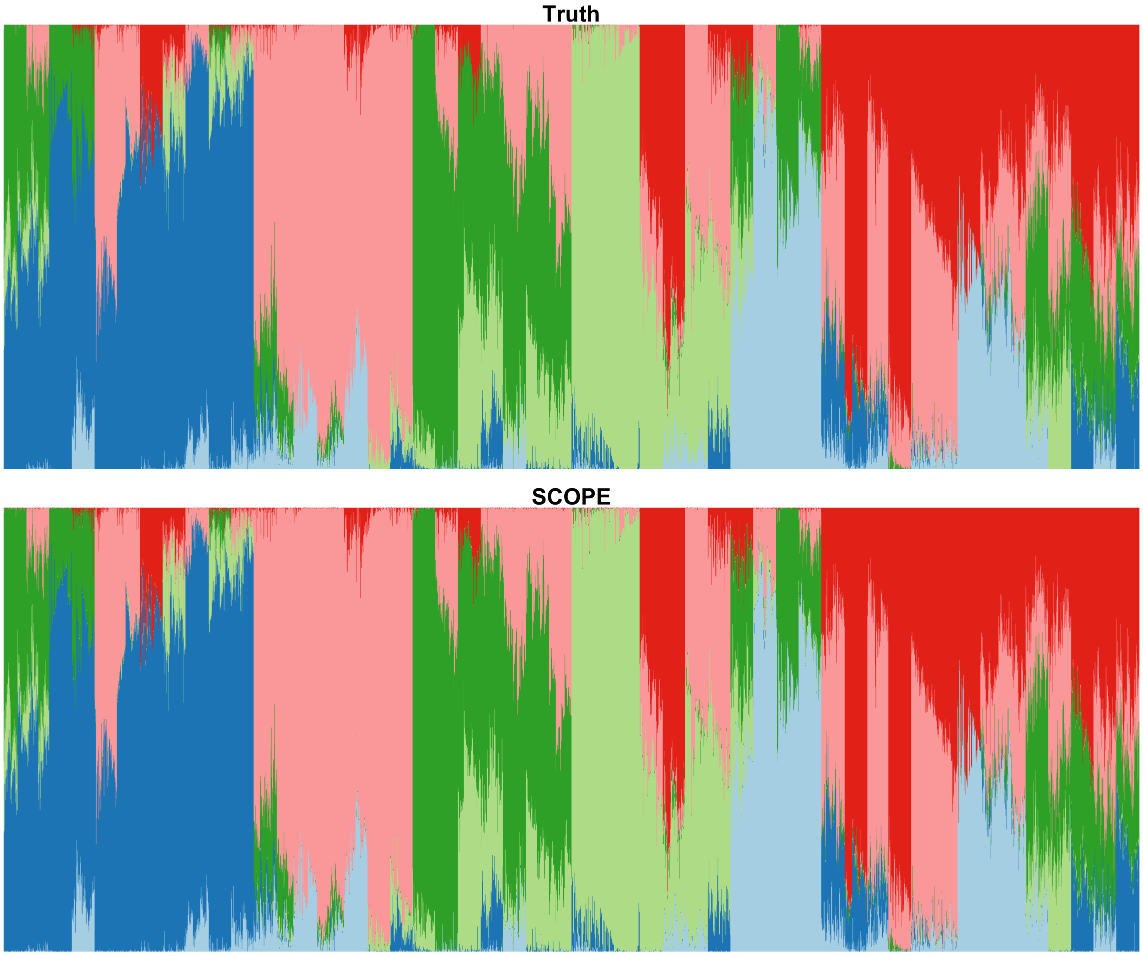


Figure B.15: **Supervised population structure inference for simulations under the PSD model generated using 1000 Genomes Phase 3 data.** PSD model parameters were drawn from TGP data to generate a simulation dataset with 100,000 samples and 1 million SNPs. Both were methods provided the true population allele frequencies as input. Colors and order of samples are matched between each method to the truth.

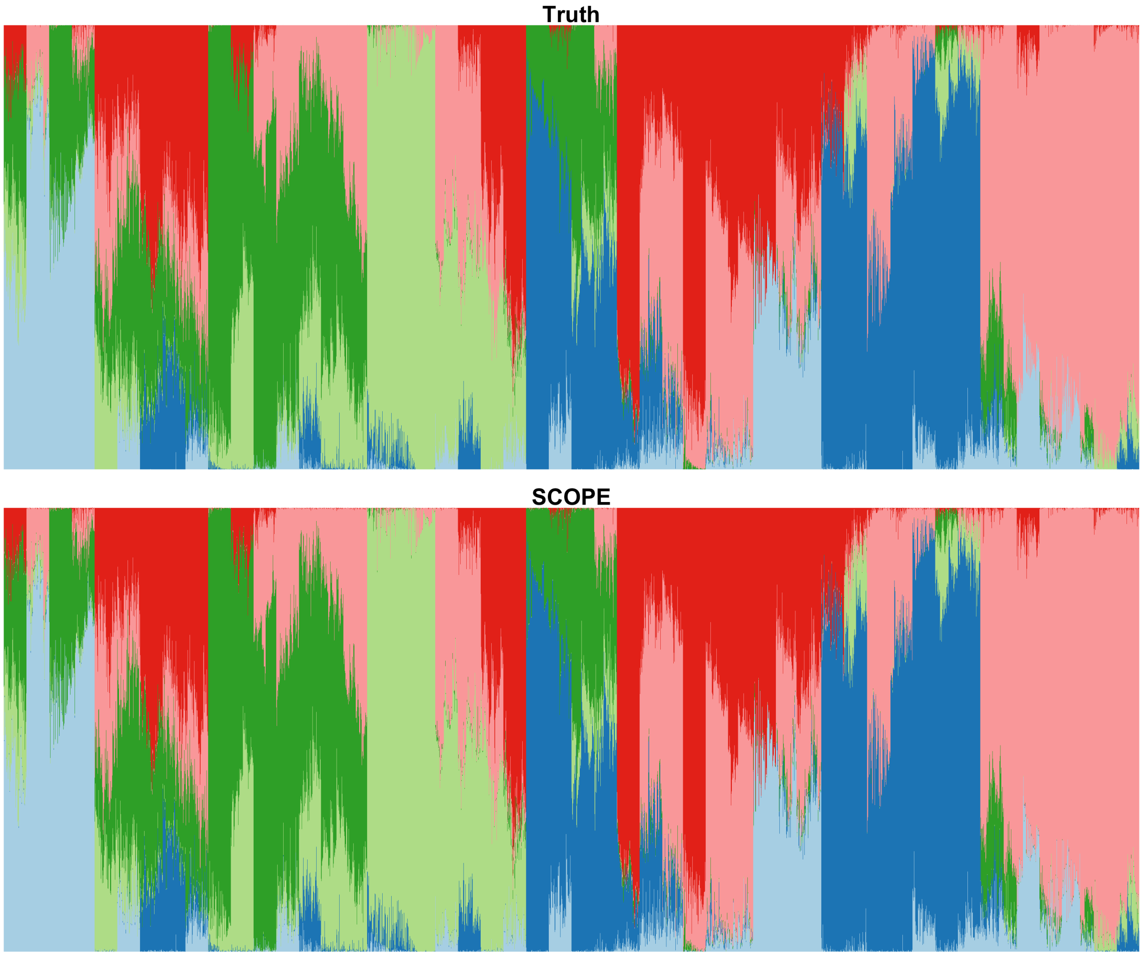


Figure B.16: **Supervised population structure inference for simulations under the PSD model generated using 1000 Genomes Phase 3 data.** PSD model parameters were drawn from TGP data to generate a simulation dataset with 1 million individuals SNPs. SCOPE was provided the true population allele frequencies as input. Colors and order of samples are matched between SCOPE and the truth.

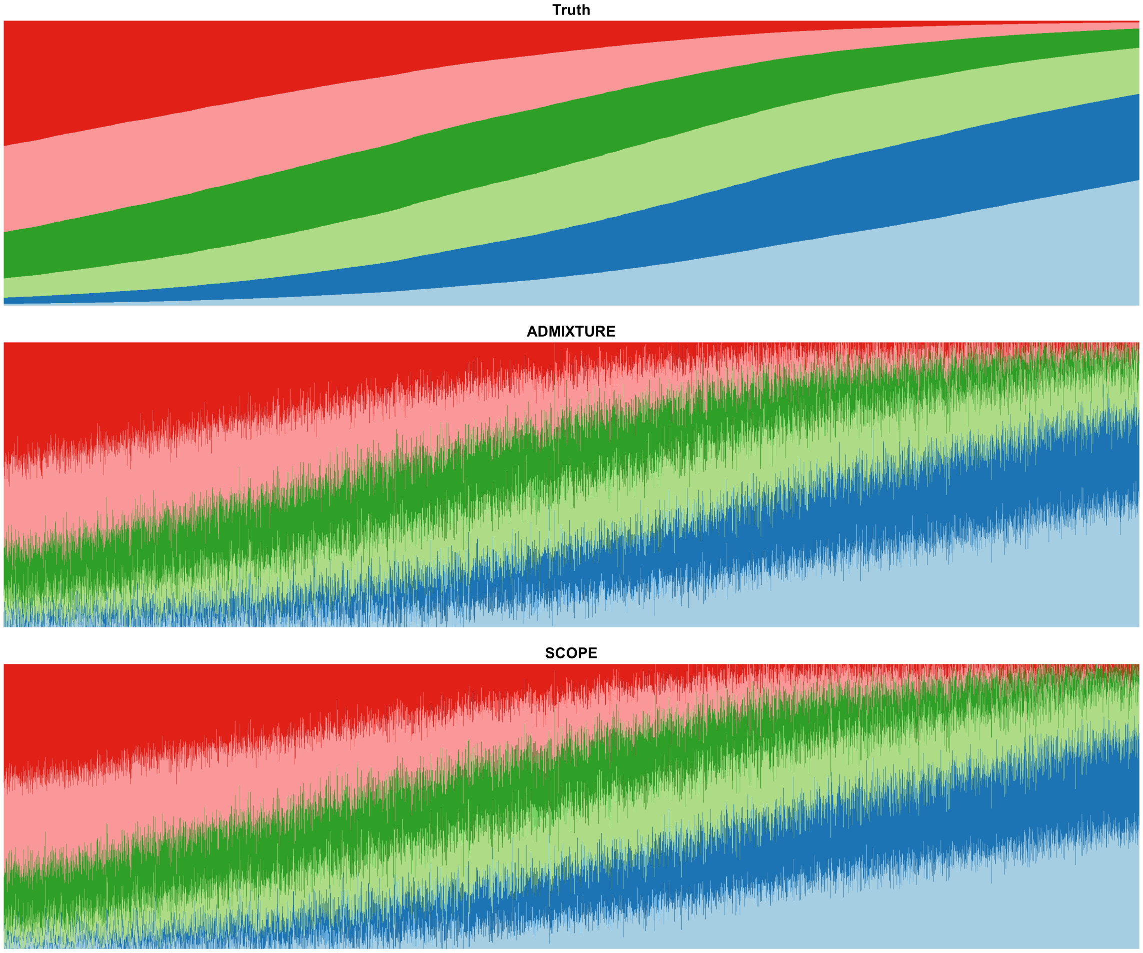


Figure B.17: **Supervised population structure inference for simulations under a spatial model generated using Human Genome Diversity Project data.** Model parameters were drawn from HGDP data to generate a simulation dataset with 10,000 samples and 10,000 SNPs under a spatial model. Both methods were provided the true population allele frequencies as input. Colors and order of samples are matched between each method to the truth.

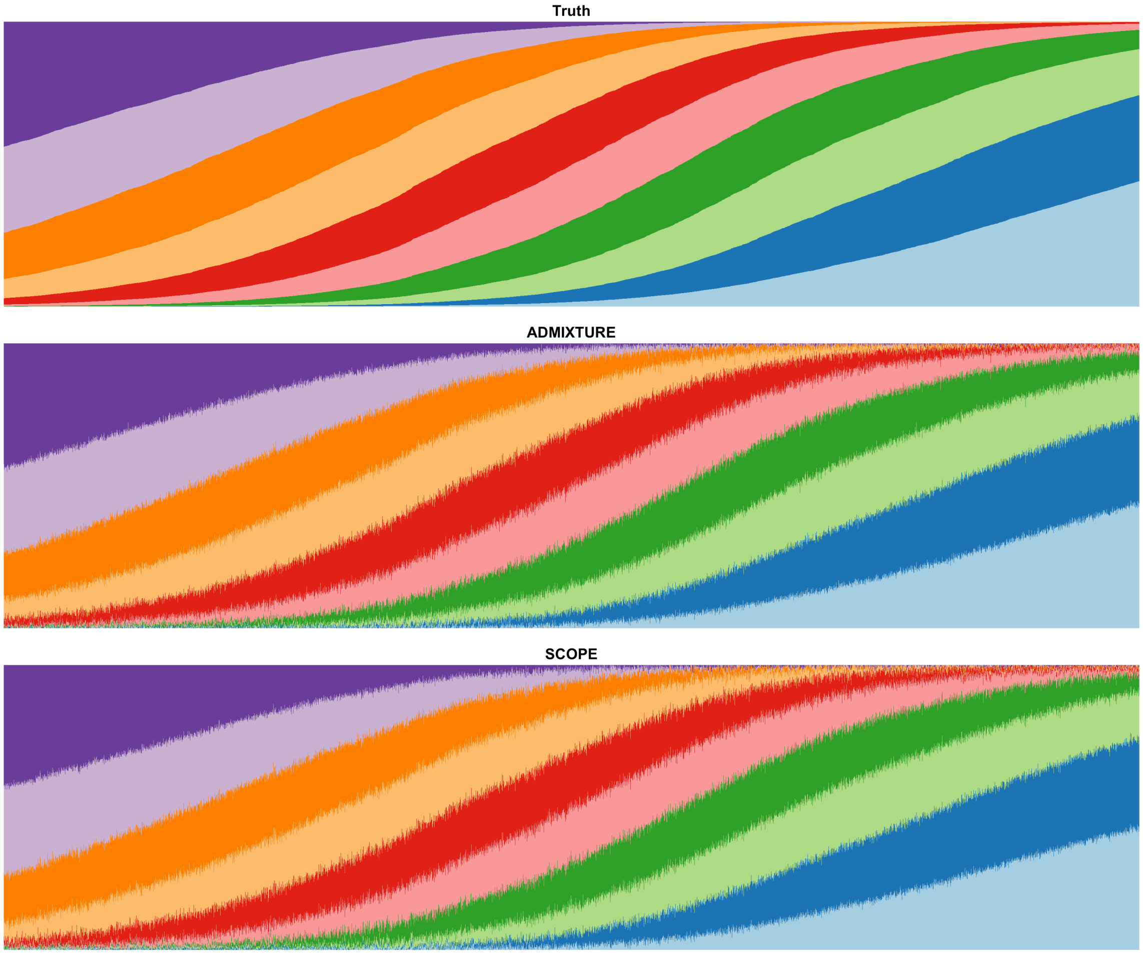


Figure B.18: **Supervised population structure inference for simulations under a spatial model generated using 1000 Genomes Phase 3 data.** Model parameters were drawn from TGP data to generate a simulation dataset with 10,000 samples and 100,000 SNPs under a spatial model. Both methods were provided the true population allele frequencies as input. Colors and order of samples are matched between each method to the truth.

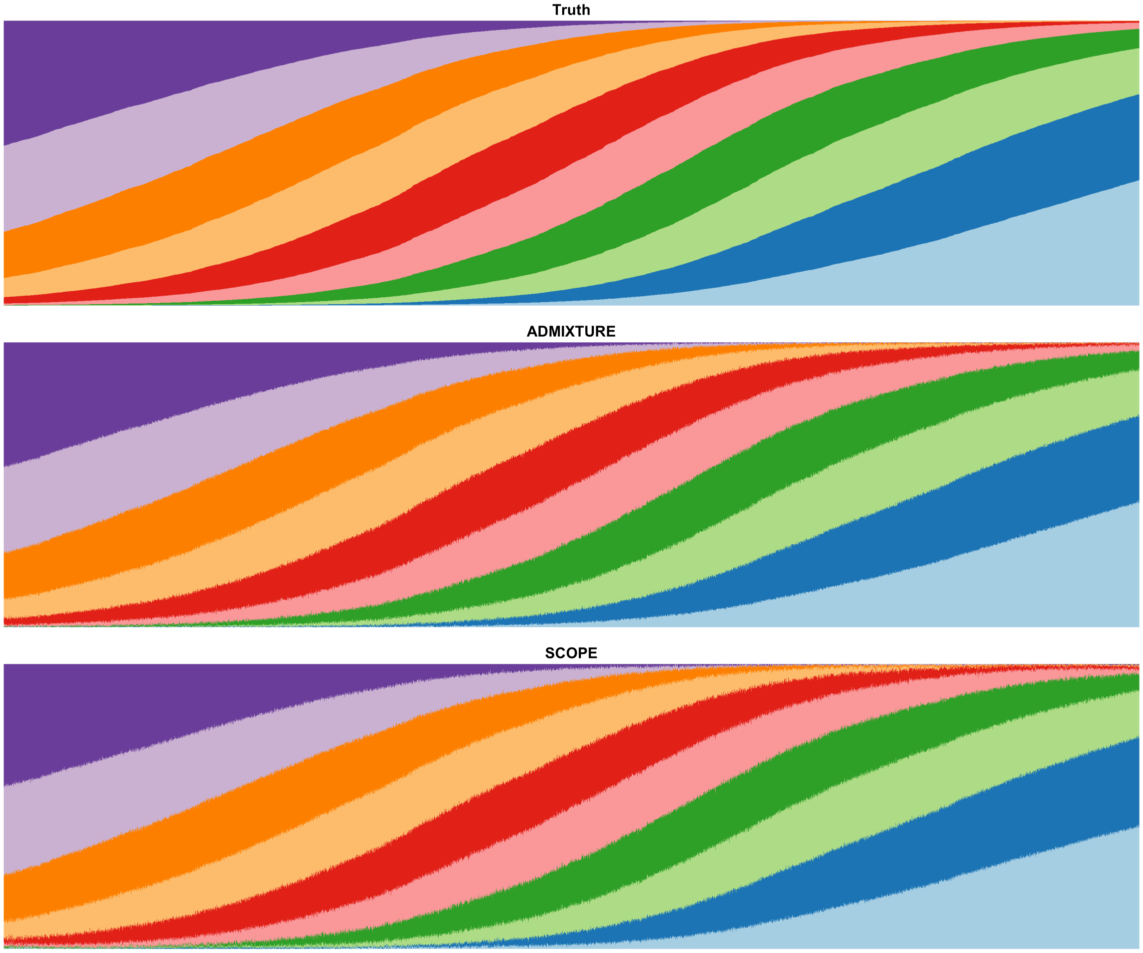


Figure B.19: **Supervised population structure inference for simulations under a spatial model generated using 1000 Genomes Phase 3 data.** Model parameters were drawn from TGP data to generate a simulation dataset with 10,000 samples and 1 million SNPs under a spatial model. Both methods were provided the true population allele frequencies as input. Colors and order of samples are matched between each method to the truth.

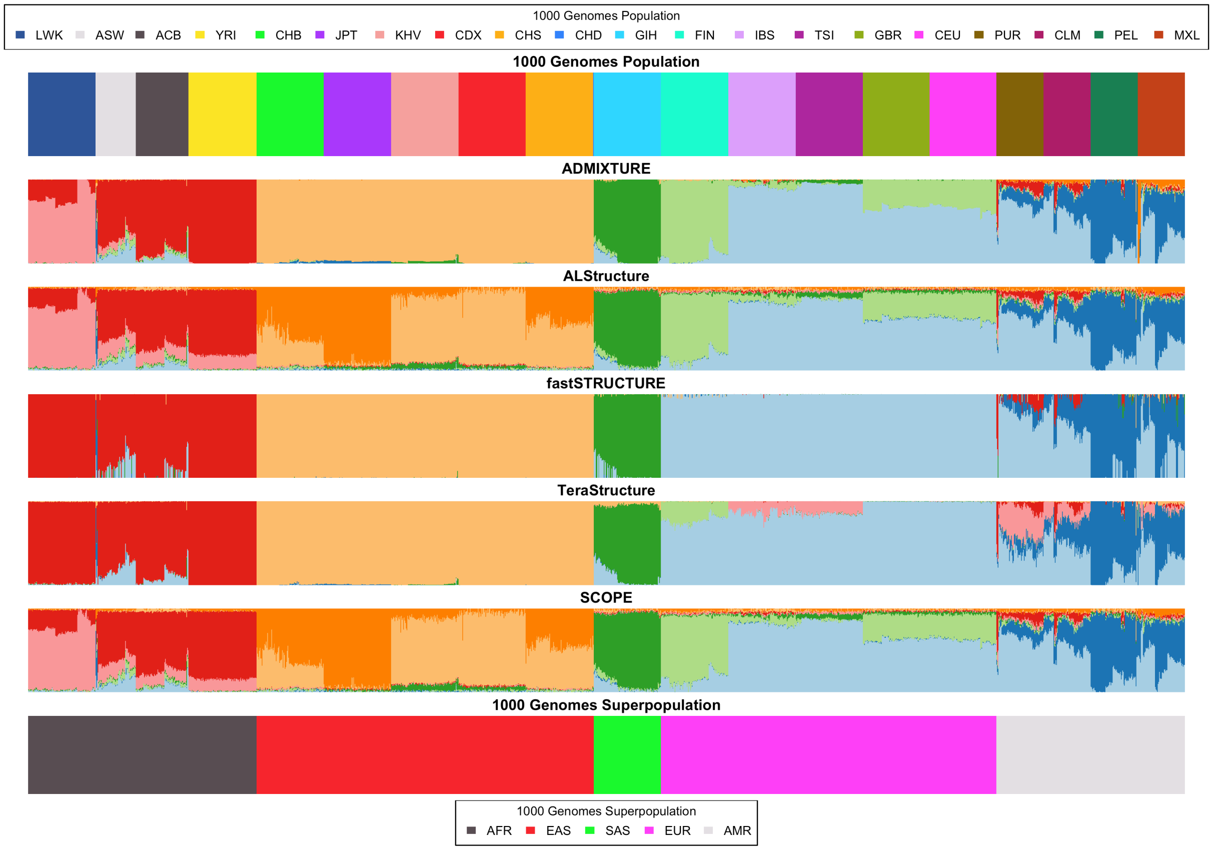


Figure B.20: **Population structure inference of 1000 Genomes Phase 3 data using 8 latent populations.** Colors are matched between each method and ADMIXTURE. Samples are ordered through hierarchical clustering (see Methods). The superpopulations and superpopulations are shown for reference.

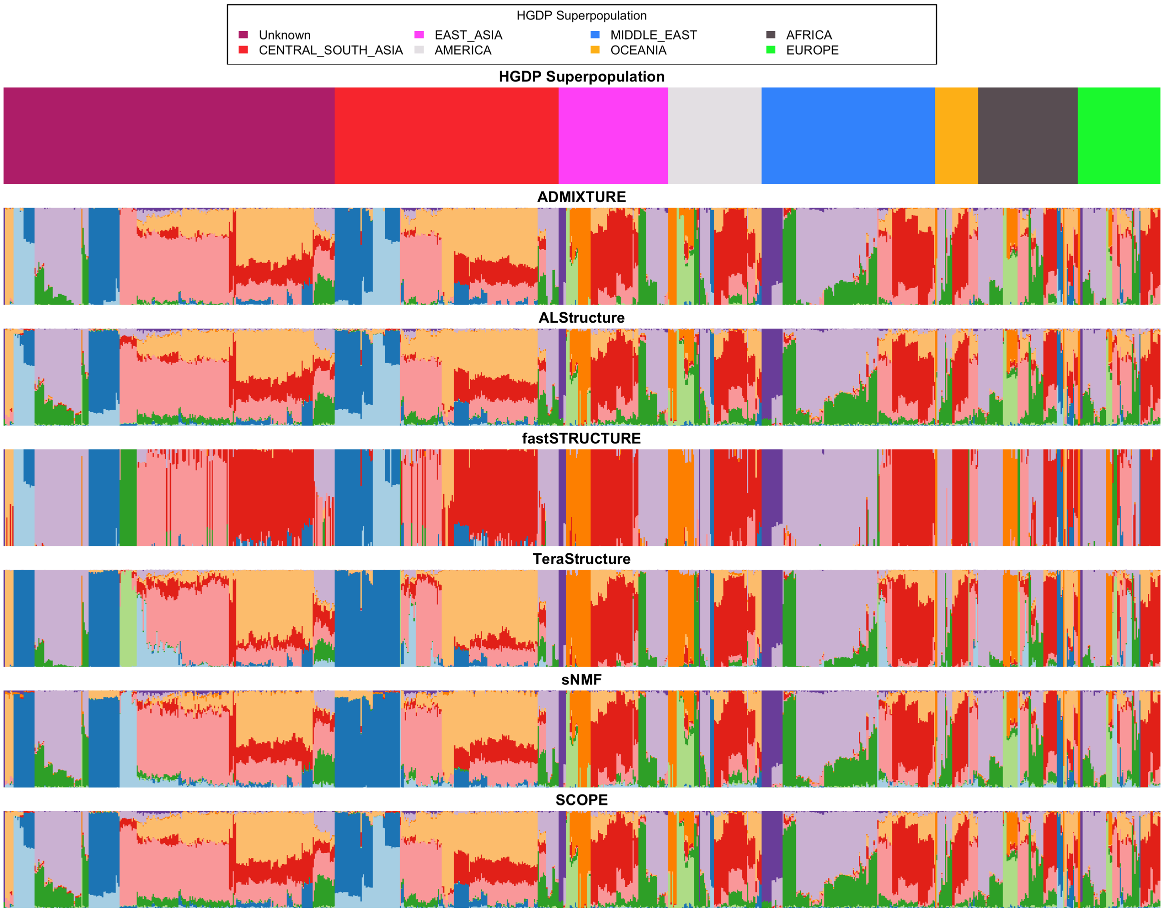


Figure B.21: **Population structure inference of Human Genomes Diversity Population data using 10 latent populations.** Colors are matched between each method and ADMIXTURE. Samples are ordered through hierarchical clustering (see Methods). HGDP superpopulation is shown for reference.

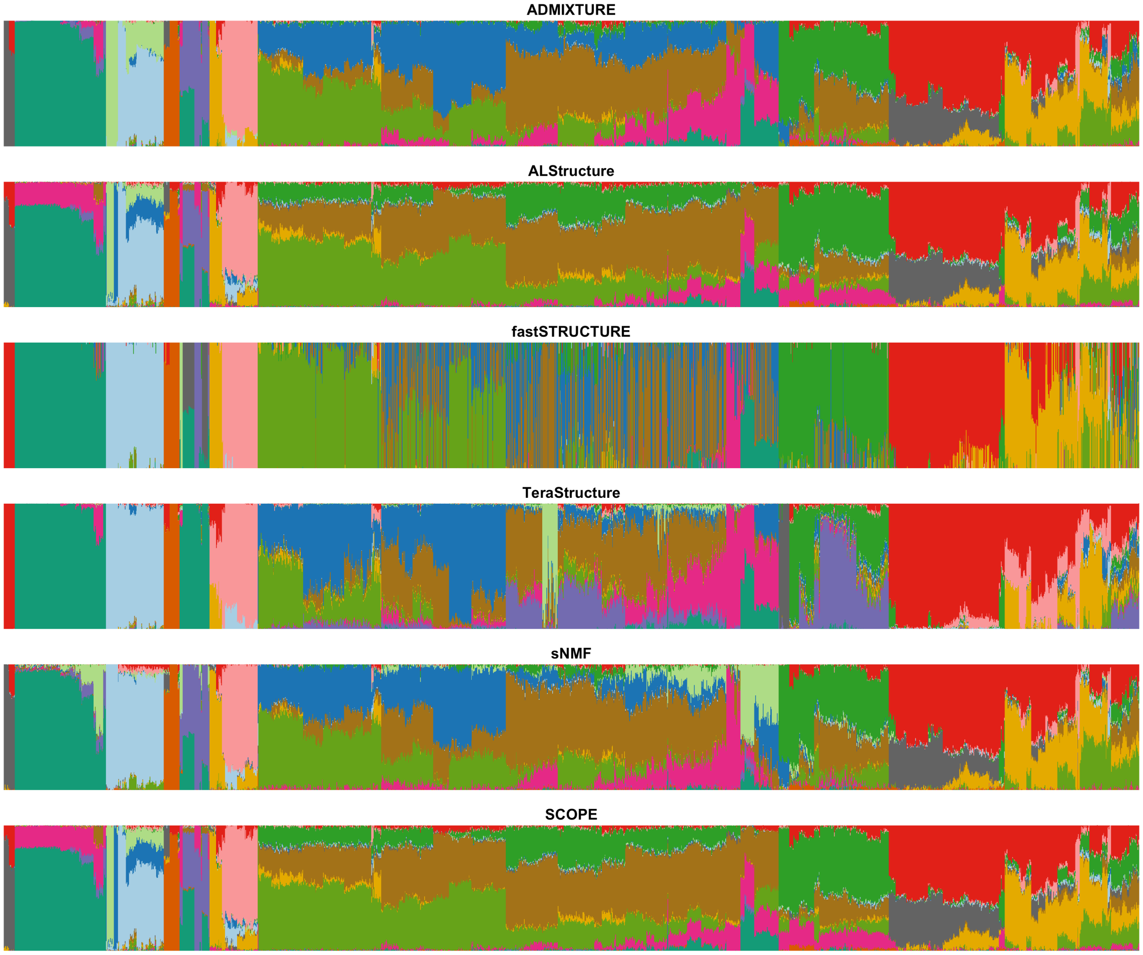


Figure B.22: **Population structure inference of Human Origins data using 14 latent populations.** Colors and order of samples are matched between each method and ADMIXTURE. ADMIXTURE was ordered through hierarchical clustering (see Methods).

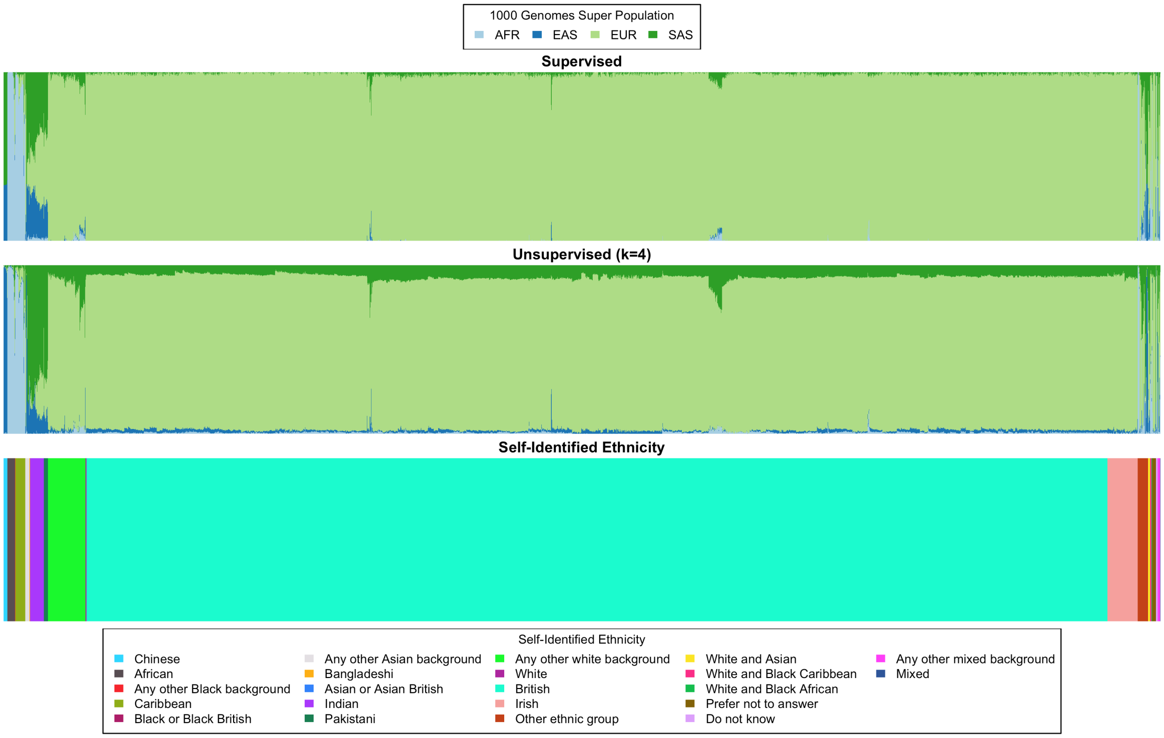


Figure B.23: **Population structure inference on the UK Biobank with all individuals.** We ran population structure inference using SCOPE (488,363 individuals and 569,346 SNPs) in both supervised mode using 1000 Genomes Phase 3 allele frequencies (top) and unsupervised with 4 latent populations (middle). For reference, we plot the self-identified race/ethnicity (bottom). Colors and order of samples are matched between each row of the figure. This is an extended version of Figure 3.4 that includes all self-identified British samples.

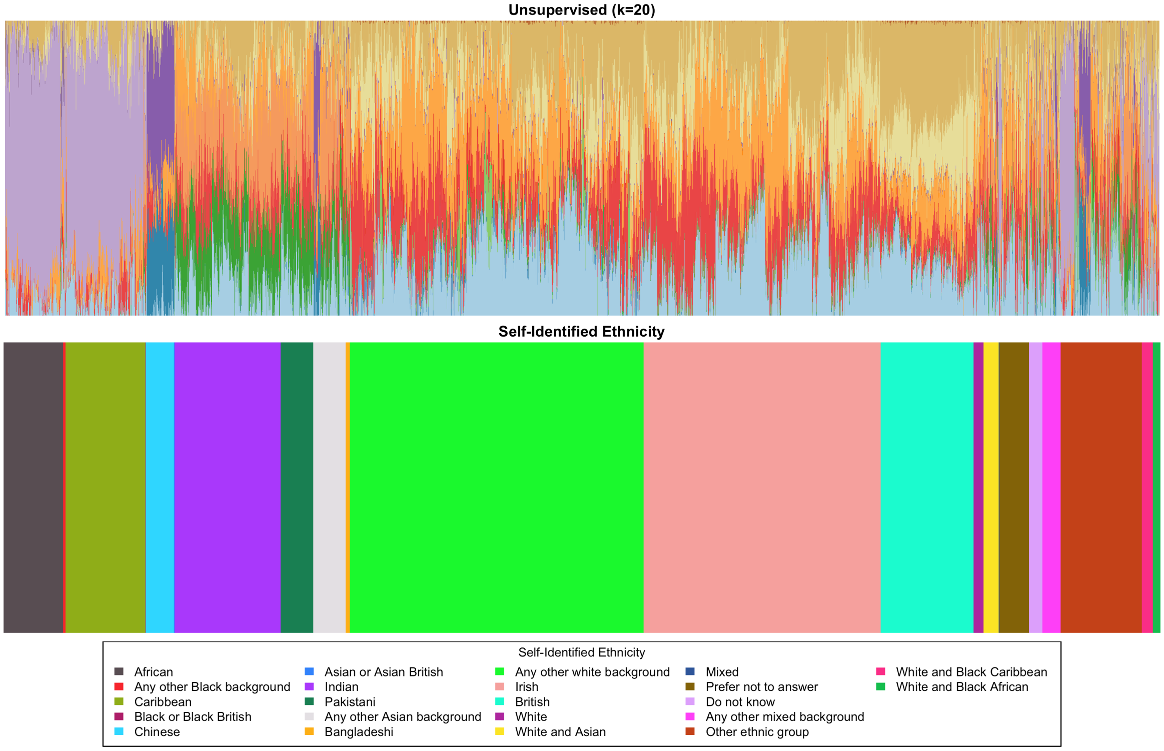


Figure B.24: **Population structure inference on the UK Biobank with 20 latent populations.** We ran population structure inference using SCOPE unsupervised with 20 latent populations on the UK Biobank (488,363 individuals and 147,604 SNPs) (top). For reference, we plot the self-identified race/ethnicity (bottom). For visualization purposes, we reduced the number of self-identified British individuals to a random subset of $5,000$ individuals. Colors and order of samples are matched between each row of the figure.

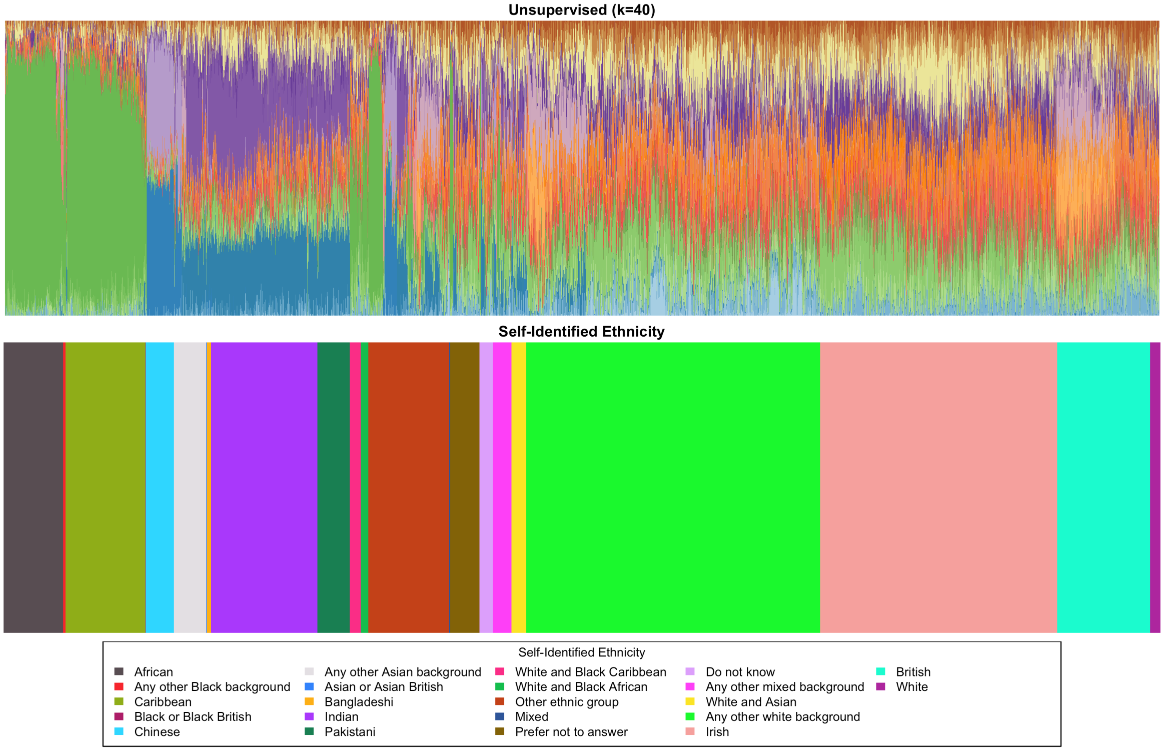


Figure B.25: **Population structure inference on the UK Biobank with 40 latent populations.** We ran population structure inference using SCOPE unsupervised with 40 latent populations on the UK Biobank (488,363 individuals and 147,604 SNPs) (top). For reference, we plot the self-identified race/ethnicity (bottom). For visualization purposes, we reduced the number of self-identified British individuals to a random subset of $5,000$ individuals. Colors and order of samples are matched between each row of the figure.

Table B.1: **Kullback-Leibler divergence measurements for methods on simulated data with truth as first input.** Kullback-Leibler divergence (KLD) was computed against the ground truth admixture proportions for each simulation using truth as first input. Values are displayed as percentages rounded to one decimal place. Estimated proportions of 0 were set to 1×10^{-9} (see Methods). A '-' denotes that the method was not run due to projected time or memory usage. Bold values denote the best value for each dataset.

Dataset Type	Base Dataset	k	n	m	ADMIXTURE	fastStructure	TeraStructure	ALStructure	sNMF	SCOPE
PSD	HGDP	6	10,000	10,000	**8.3**	124.6	48.4	12.3	8.8	12.3
PSD	TGP	6	10,000	10,000	**3.4**	233.5	35.5	7.1	8.8	7.1
PSD	TGP	6	10,000	1,000,000	**0.2**	320.8	0.9	-	-	0.5
PSD	TGP	6	100,000	1,000,000	-	-	1.1	-	-	**0.6**
PSD	TGP	6	1,000,000	1,000,000	-	-	-	-	-	**0.7**
Spatial	HGDP	6	10,000	10,000	49.22	630.6	20.9	25.6	**15.3**	31.5
Spatial	TGP	6	10,000	10,000	62.8	596.7	**9.25**	60.6	25.7	58.6
Spatial	TGP	10	10,000	100,000	134.0	778.1	**27.2**	116.9	47.91	85.2
Spatial	TGP	10	10,000	1,000,000	-	-	**30.5**	-	-	85.6

Table B.2: **Kullback-Leibler divergence measurements for methods on simulated data with truth as second input.** Kullback-Leibler (KLD) was computed against the ground truth admixture proportions for each simulation using truth as second input. Values are displayed as percentages rounded to one decimal place. Estimated proportions of 0 were set to 1×10^{-9} (see Methods). A '-' denotes that the method was not run due to projected time or memory usage. Bold values denote the best value for each dataset.

Dataset Type	Base Dataset	k	n	m	ADMIXTURE	fastStructure	TeraStructure	ALStructure	sNMF	SCOPE
PSD	HGDP	6	10,000	10,000	313.5	**219.8**	1560.7	476.9	311.5	476.0
PSD	TGP	6	10,000	10,000	**91.84**	197.9	769.7	260.8	311.5	259.3
PSD	TGP	6	10,000	1,000,000	**1.6**	175.9	16.0	-	-	25.87
PSD	TGP	6	100,000	1,000,000	-	-	40.4	-	-	**35.6**
PSD	TGP	6	1,000,000	1,000,000	-	-	-	-	-	**38.3**
Spatial	HGDP	6	10,000	10,000	24.9	127.2	30.4	**8.0**	8.8	9.9
Spatial	TGP	6	10,000	10,000	25.1	111.0	**10.8**	12.3	15.0	11.8
Spatial	TGP	10	10,000	100,000	56.8	136.8	33.7	32.3	23.9	**22.9**
Spatial	TGP	10	10,000	1,000,000	-	-	**29.2**	-	-	29.5

Table B.3: **Memory usage of methods on simulated and real datasets.** ADMIXTURE, TeraStructure, sNMF, and SCOPE were run using 8 threads. ALStructure and fastStructure were run on a single thread due to their lack of multithreading implementations. TeraStructure's '-rfreq' parameter was set to 10% of the number of SNPs. A '-' denotes that the method was not run due to projected time or memory usage. Default parameters were used otherwise. Memory is displayed in gigabytes (GB). Bold values denote the best value for each dataset.

Dataset Type	Base Dataset	k	n	m	ADMIXTURE	fastStructure	TeraStructure	ALStructure	sNMF	SCOPE
PSD	HGDP	6	10,000	10,000	0.12	0.17	0.12	7.30	**0.04**	0.14
PSD	TGP	6	10,000	10,000	0.12	0.16	0.12	7.30	**0.04**	0.14
PSD	TGP	6	10,000	1,000,000	10.66	10.66	**9.96**	-	-	12.60
PSD	TGP	6	100,000	1,000,000	-	-	94.38	-	-	**93.47**
PSD	TGP	6	1,000,000	1,000,000	-	-	-	-	-	**746.19**
Spatial	HGDP	6	10,000	10,000	0.12	0.17	0.12	7.30	**0.04**	0.14
Spatial	TGP	6	10,000	10,000	0.12	0.16	0.12	7.30	**0.04**	0.14
Spatial	TGP	10	10,000	100,000	1.17	1.33	1.05	33.20	**0.38**	1.28
Spatial	TGP	10	10,000	1,000,000	-	-	**10.30**	-	-	12.69
Real	HGDP	10	940	642,951	1.94	1.99	1.17	24.38	**0.36**	1.30
Real	HO	14	1,931	385,089	1.83	1.89	1.21	27.45	**0.38**	1.53
Real	TGP	8	1,718	1,854,622	6.20	6.18	**4.44**	145.49	-	6.34
Real	UKB	4	488,363	569,346	-	-	-	-	-	**230.57**
Real	UKB	20	488,363	147,604	-	-	-	-	-	**60.92**
Real	UKB	40	488,363	147,604	-	-	-	-	-	**62.01**

Table B.4: **Accuracy of supervised population structure inference for SCOPE and ADMIXTURE using supplied allele frequencies on simulations.** True allele frequencies were supplied to each method. Root-mean-square error (RMSE) and Jensen-Shannon Divergence (JSD) were computed against the true admixture proportions. Estimated proportions of 0 were set to 1×10^{-9} for JSD calculations (see Methods). A "-" denotes that the method was not run for that dataset due to time or memory constraints. Values are displayed as percentages. Bold values denote the best value for each dataset.

					SCOPE		ADMIXTURE	
Dataset Type	Base Dataset	k	n	m	RMSE	JSD	RMSE	JSD
PSD	HGDP	6	10,000	10,000	2.9	1.5	**2.6**	**1.2**
PSD	TGP	6	10,000	10,000	2.0	0.9	**1.6**	**0.6**
PSD	TGP	6	10,000	1,000,000	**0.2**	0.1	**0.2**	**0.03**
PSD	TGP	6	100,000	1,000,000	**0.2**	0.1	**0.2**	**0.03**
PSD	TGP	6	1,000,000	1,000,000	**0.2**	**0.1**	-	-
Spatial	HGDP	6	10,000	10,000	**2.4**	**0.6**	3.2	0.9
Spatial	TGP	6	10,000	10,000	**1.7**	**0.3**	2.2	0.4
Spatial	TGP	10	10,000	100,000	**0.6**	**0.3**	0.7	**0.3**
Spatial	TGP	10	10,000	1,000,000	0.3	**0.1**	**0.2**	**0.1**

Table B.5: **Prediction accuracy of self-identified race and ethnicity using inferred admixture proportions.** We trained multinomial logistic regression models using the inferred admixture proportions from each method to predict SIRE labels. For TGP, we predicted 5 superpopulation labels corresponding to continental ancestry from 8 inferred latent populations. For HGDP, we predicted 7 continental ancestry populations from 10 inferred latent populations. Training accuracy as a percentage is reported. sNMF was not able to be run on TGP due to its disk space requirements.

Method	TGP	HGDP
ADMIXTURE	100	46.4
ALStructure	100	47.6
fastStructure	99.4	41.8
TeraStructure	100	47.8
sNMF	-	47.6
SCOPE	100	47.2

Table B.6: **Prediction accuracy of birth location GPS coordinates for British individuals in the UK Biobank.** We trained ordinary least squares models using admixture proportions inferred by SCOPE from the three different runs on the UK Biobank. Two separate models were trained to predict the longitude coordinate and latitude coordinate. Quantiles of the difference between predicted birth location and reported birth location are displayed after the two R^2 columns and are reported in kilometers.

Number of Latent Populations	R^2 (Latitude)	R^2 (Longitude)	Minimum	25%	50%	75%	90%	95%	99%	Maximum
4	0.007	0.008	0.989	66.859	159.390	211.687	287.527	336.069	382.546	854.593
20	0.300	0.150	0.028	60.358	108.489	181.209	241.689	292.441	386.268	892.224
40	0.230	0.149	0.079	63.429	117.495	189.312	252.232	297.463	392.643	871.836

APPENDIX C

Supplementary Material - A simple statistical testing framework for detecting differences in variance and covariance in gene expression networks

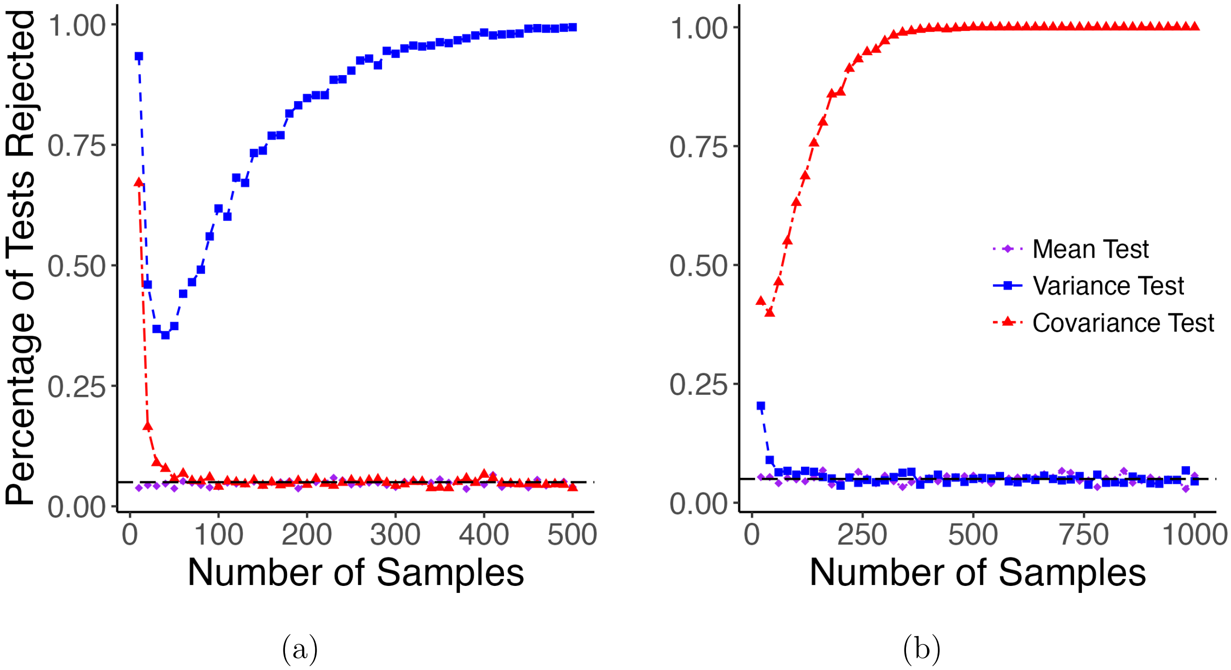
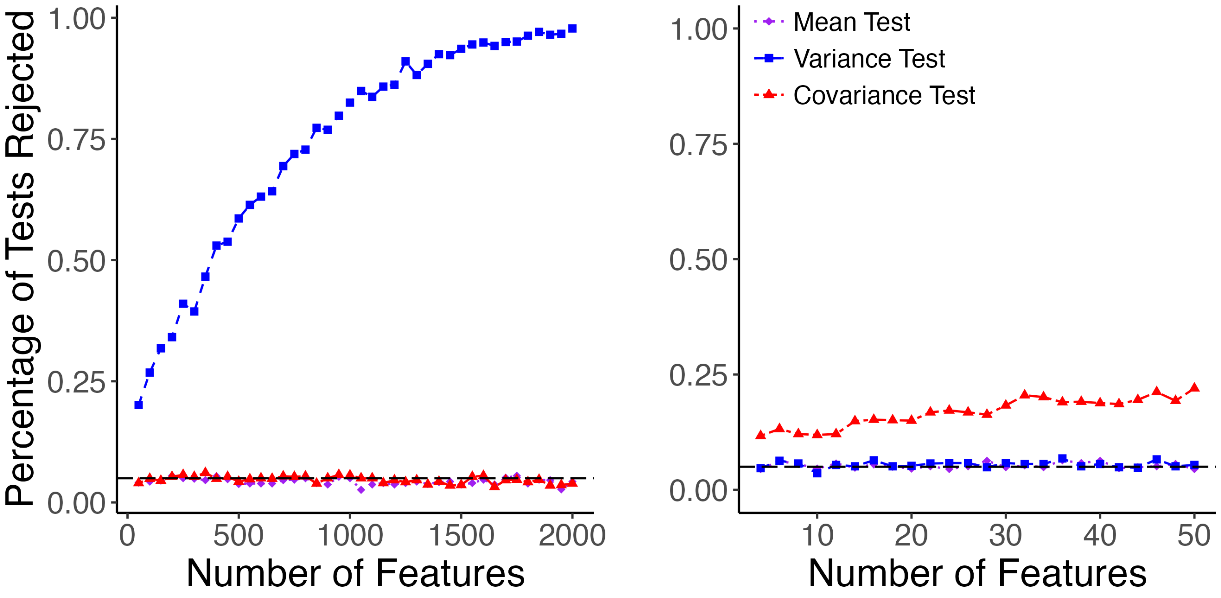


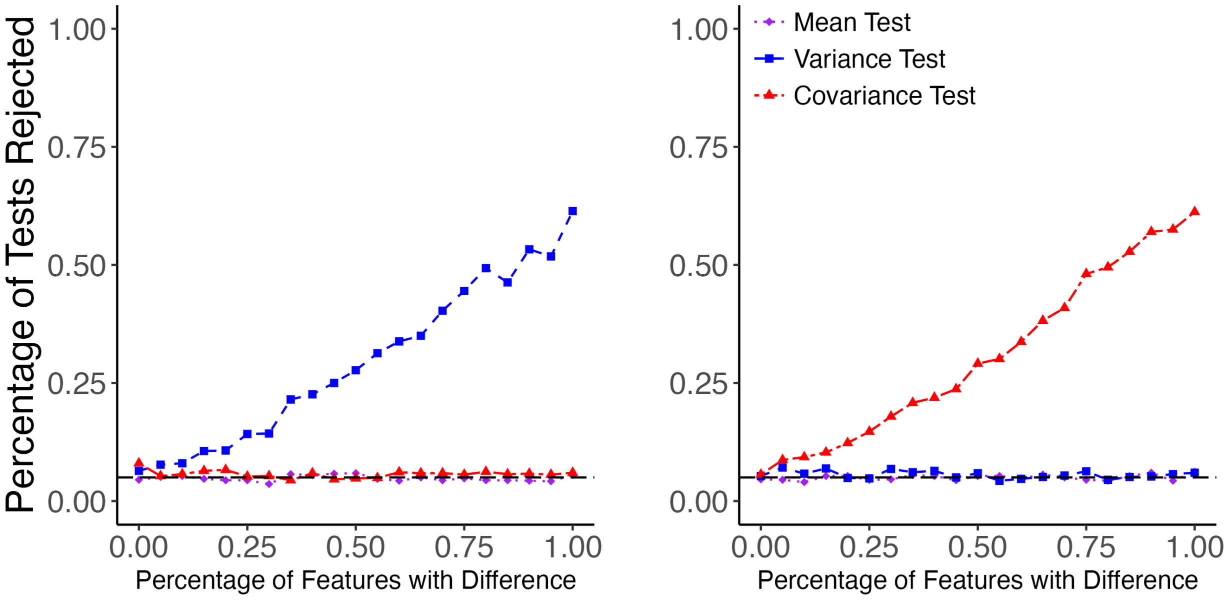
Figure C.1: **Tests increase in power as sample sizes increase.** We performed simulations containing either variance differences of 0.1 (Figure C.1a) or covariance differences of 0.2 (Figure C.1b). 1,000 simulations were performed on each point using a simulation dataset of 500 genes. The dashed line represents $p = 0.05$.



(a)

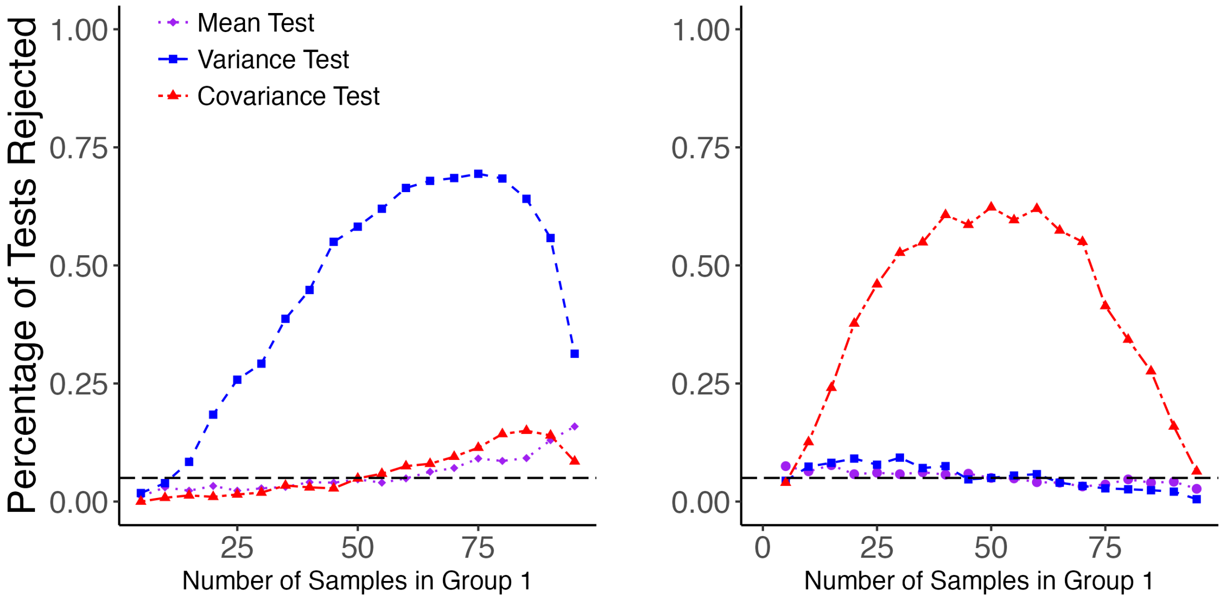
(b)

Figure C.2: **Tests increase in power as number of features increases.** We performed simulations containing either variance differences of 0.1 (Figure C.2a) or covariance differences of 0.2 (Figure C.2b). $1,000$ simulations were performed on each point using a simulation dataset of 100 samples. The dashed line represents $p = 0.05$.



(a) (b)

Figure C.3: **Tests increase in power as number of relevant genes/features increases.** We performed simulations containing either variance differences of 0.1 (Figure C.3a) or covariance differences of 0.2 (Figure C.3b). $1,000$ simulations were performed on each point using a simulation dataset of 100 samples and 500 genes. The dashed line represents $p = 0.05$.



(a) (b)

Figure C.4: **Tests have improved power when there are more even sample sizes between groups.** We performed simulations containing either variance differences of 0.1 (Figure C.4a) or covariance differences of 0.2 (Figure C.4b). $1,000$ simulations were performed on each point using a simulation dataset of 100 samples and 500 genes. The dashed line represents $p = 0.05$.

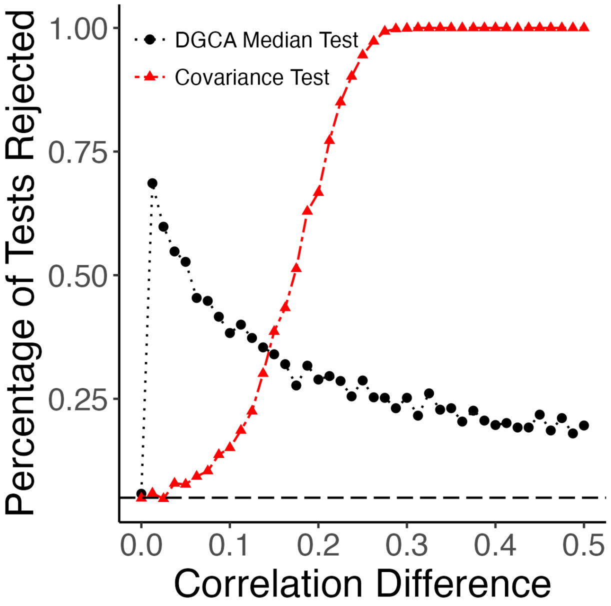
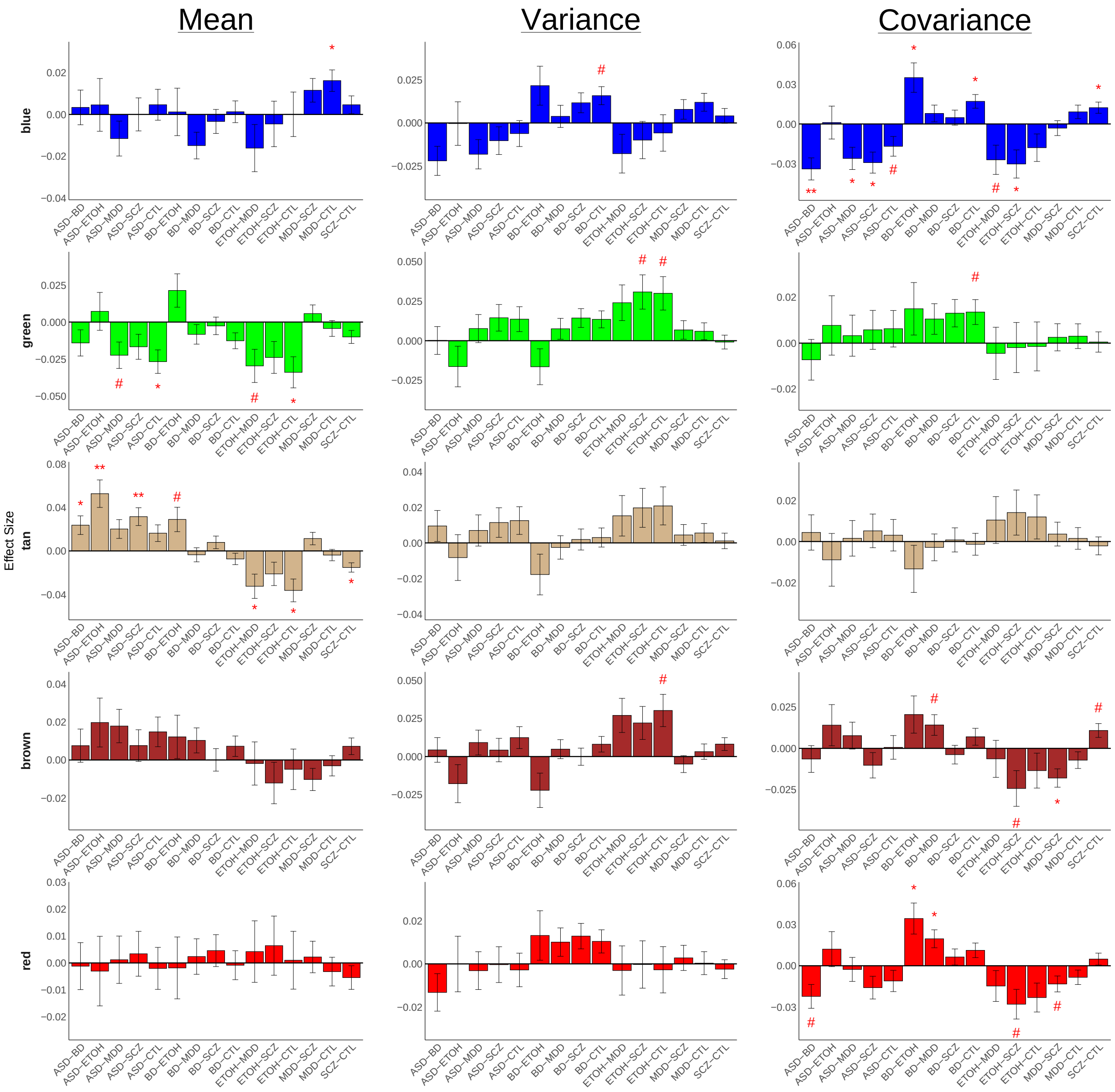
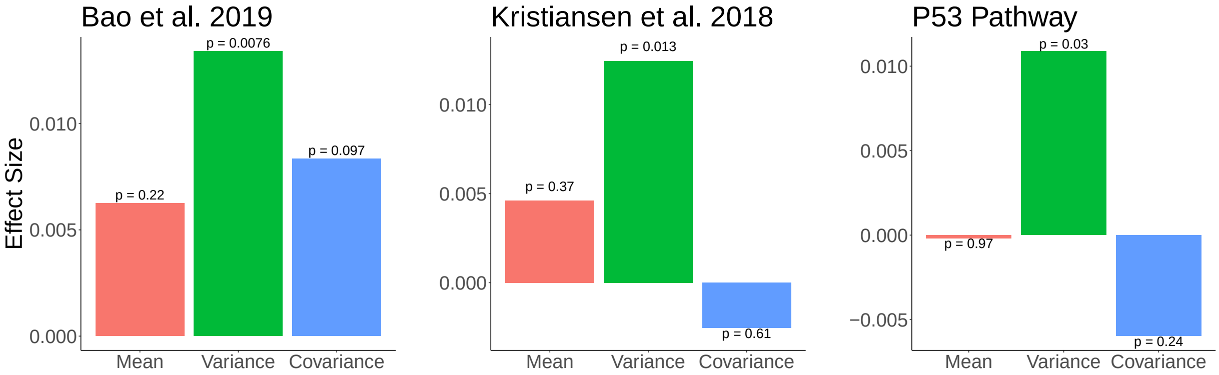


Figure C.5: **DGCA median test does not calibrate in simulations.** We applied the DGCA median test on our simulated frameworks and found that it does not calibrate as expected compared to our proposed covariance test. Each point represents the percentage of tests rejected out of a thousand simulations. The black dashed line represents $p = 0.05$.



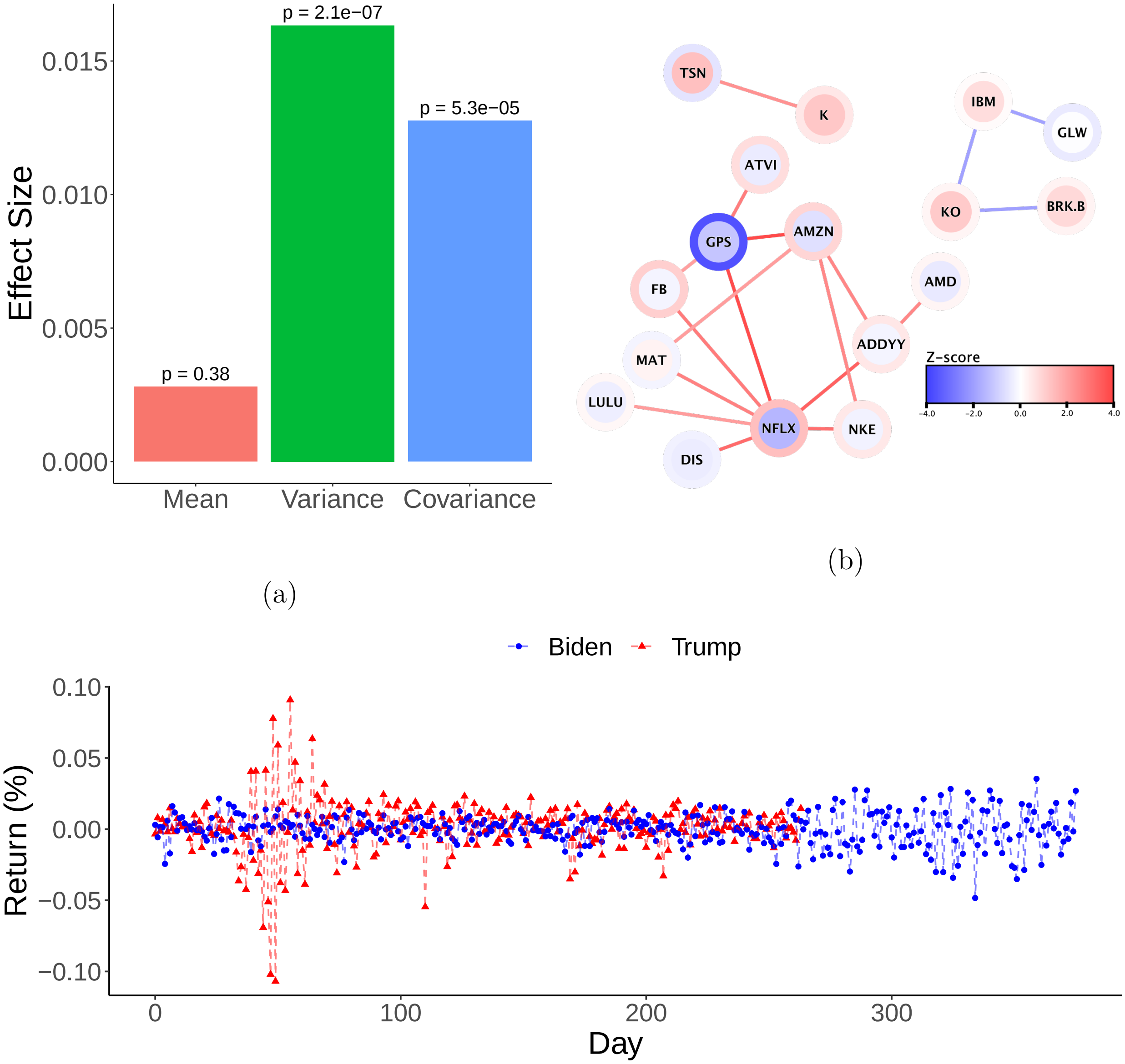
(a) (b) (c)

Figure C.6: **Gene expression differences in remaining modules in psychiatric disorders.** We tested for mean differences (Figure C.6a), variance differences (Figure C.6b), and covariance differences (Figure C.6c). The second group in each label denotes the reference group in the comparison. A pound sign, single asterisk, double asterisks, and triple asterisks denote significance at 0.1, 0.05, 0.01, and 0.001 after multiple testing correction, respectively.



(a) (b) (c)

Figure C.7: **Increased variance in methylation probe sets in dead breast cancer patients.** We applied our testing framework to three methylation probe sets. All found increased variance in dead patients. Text above or below each bar represents *p*-values for the respective test.



(a)

(b)

(c)

Figure C.8: **Application of testing framework to stock data reveals variance and covariance differences between presidencies.** We applied our testing framework to the daily percentage returns of 30 stocks from 01/03/2020 to 7/19/2022. Figure C.8a displays the overall effect size and corresponding *p*-values. Figure C.8b is a Cytoscape network plot of stocks with an absolute Z-scored difference in covariance greater than 1.96. Node color represents mean value, border color represents variance value, and edge color represents covariance value. Figure C.8c shows the mean return of an equally weighted portfolio over time.

Control Type	Method	Number of Features	Significant Tests (0.05)	Significant Tests (0.1)
Positive	Variance (MDSeq)	50-4,978	100%	100%
Positive	Variance (F-test)	50-2,000	91%	98%
Positive	Covariance (DiffCor)	50-30,628	99%	100%
Positive	Covariance (CILP)	190-44,850	100%	100%
Negative	Variance (MDSeq)	50-2,000	0%	0%
Negative	Variance (F-test)	50-2,000	0%	0%
Negative	Covariance (DiffCor)	190-44,850	0%	0%
Negative	Covariance (CILP)	190-44,850	0%	0%

Table C.1: **Positive and negative controls generated from individual feature methods on real data.** We generated $1,000$ artificial gene sets composed all positive or all negative features from individual feature or pair methods for variance or covariance, respectively. MDSeq and an F-test were applied to GTEx sun-exposed and non-exposed skin cells for generating variance gene sets. DiffCor and CILP were used to generate covariance gene sets. The number of features refers to the number of columns in the matrix from which eigengenes were obtained. The significant test columns denote the percentage of tests that were significant by applying our variance or covariance testing framework.

REFERENCES

[1] Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T. Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared O'Connell, Adrian Cortes, Samantha Welsh, Alan Young, Mark Effingham, Gil McVean, Stephen Leslie, Naomi Allen, Peter Donnelly, and Jonathan Marchini. The uk biobank resource with deep phenotyping and genomic data. *Nature*, 562:203–209, 2018.

[2] Richard M. Durbin and 1000 Genomes Project Consortium. A map of human genome variation from population-scale sequencing. *NATURE*, 467(7319):1061–1073, OCT 28 2010.

[3] John Michael Gaziano, John Concato, Mary Brophy, Louis Fiore, Saiju Pyarajan, James Breeling, Stacey Whitbourne, Jennifer Deen, Colleen Shannon, Donald Humphries, et al. Million veteran program: a mega-biobank to study genetic influences on health and disease. *Journal of clinical epidemiology*, 70:214–223, 2016.

[4] John Novembre, Toby Johnson, Katarzyna Bryc, Zoltán Kutalik, Adam R Boyko, Adam Auton, Amit Indap, Karen S King, Sven Bergmann, Matthew R Nelson, et al. Genes mirror geography within europe. *Nature*, 456(7218):98–101, 2008.

[5] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The cancer genome atlas pan-cancer analysis project. *Nat Genet*, 45(10):1113–1120, Oct 2013.

[6] Traver Hart, Megha Chandrashekhar, Michael Aregger, Zachary Steinhart, Kevin R. Brown, Graham MacLeod, Monika Mis, Michal Zimmermann, Amelie Fradet-Turcotte, Song Sun, Patricia Mero, Peter Dirks, Sachdev Sidhu, Frederick P. Roth, Olivia S. Rissland, Daniel Durocher, Stephane Angers, and Jason Moffat. High-resolution crispr screens reveal fitness genes and genotype-specific cancer liabilities. *Cell*, 163(6):1515–1526, 2015.

[7] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[8] Thomas Rolland, Murat Taşan, Benoit Charloteaux, Samuel J. Pevzner, Quan Zhong, Nidhi Sahni, Song Yi, Irma Lemmens, Celia Fontanillo, Roberto Mosca, Atanas Kamburov, Susan D. Ghiassian, Xinping Yang, Lila Ghamsari, Dawit Balcha, Bridget E.

Begg, Pascal Braun, Marc Brehme, Martin P. Broly, Anne-Ruxandra Carvunis, Dan Convery-Zupan, Roser Corominas, Jasmin Coulombe-Huntington, Elizabeth Dann, Matija Dreze, Amélie Dricot, Changyu Fan, Eric Franzosa, Fana Gebreab, Bryan J. Gutierrez, Madeleine F. Hardy, Mike Jin, Shuli Kang, Ruth Kiros, Guan Ning Lin, Katja Luck, Andrew MacWilliams, Jörg Menche, Ryan R. Murray, Alexandre Palagi, Matthew M. Poulin, Xavier Rambout, John Rasla, Patrick Reichert, Viviana Romero, Elien Ruyssinck, Julie M. Sahalie, Annemarie Scholz, Akash A. Shah, Amitabh Sharma, Yun Shen, Kerstin Spirohn, Stanley Tam, Alexander O. Tejeda, Shelly A. Trigg, Jean-Claude Twizere, Kerwin Vega, Jennifer Walsh, Michael E. Cusick, Yu Xia, Albert-László Barabási, Lilia M. Iakoucheva, Patrick Aloy, Javier De Las Rivas, Jan Tavernier, Michael A. Calderwood, David E. Hill, Tong Hao, Frederick P. Roth, and Marc Vidal. A proteome-scale map of the human interactome network. *Cell*, 159(5):1212–1226, 2014.

[9] Danielle Welter, Jacqueline MacArthur, Joannella Morales, Tony Burdett, Peggy Hall, Heather Junkins, Alan Klemm, Paul Flicek, Teri Manolio, Lucia Hindorff, and Helen Parkinson. The nhgri gwas catalog, a curated resource of snp-trait associations. *Nucleic Acids Research*, 42(D1):D1001–D1006, 8/12/2022 2014.

[10] Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 8/12/2022 2010.

[11] Guillaume Lettre et al. Genome-wide association study of coronary heart disease and its risk factors in 8,090 African Americans: The NHLBI CARe Project. *PLoS Genet*, in press.

[12] Lucia A. Hindorff, Praveen Sethupathy, Heather A. Junkins, Erin M. Ramos, Jayashri P. Mehta, Francis S. Collins, and Teri A. Manolio. Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *Proceedings of the National Academy of Sciences*, 106(23):9362–9367, 2009.

[13] Loic Yengo, Julia Sidorenko, Kathryn E Kemper, Zhili Zheng, Andrew R Wood, Michael N Weedon, Timothy M Frayling, Joel Hirschhorn, Jian Yang, Peter M Visscher, and the GIANT Consortium. Meta-analysis of genome-wide association studies for height and body mass index in 700000 individuals of european ancestry. *Human Molecular Genetics*, 27(20):3641–3649, 8/12/2022 2018.

[14] Andrew R Wood, Tonu Esko, Jian Yang, Sailaja Vedantam, Tune H Pers, Stefan Gustafsson, Audrey Y Chu, Karol Estrada, Jian'an Luan, Zoltán Kutalik, Najaf Amin, Martin L Buchkovich, Damien C Croteau-Chonka, Felix R Day, Yanan Duan, Tove Fall, Rudolf Fehrmann, Teresa Ferreira, Anne U Jackson, Juha Karjalainen, Ken Sin Lo, Adam E Locke, Reedik Mägi, Evelin Mihailov, Eleonora Porcu, Joshua C Randall, André Scherag, Anna A E Vinkhuyzen, Harm-Jan Westra, Thomas W Winkler, Tsegaselassie Workalemahu, Jing Hua Zhao, Devin Absher, Eva Albrecht, Denise Anderson, Jeffrey Baron, Marian Beekman, Ayse Demirkan, Georg B Ehret, Bjarke

Feenstra, Mary F Feitosa, Krista Fischer, Ross M Fraser, Anuj Goel, Jian Gong, Anne E Justice, Stavroula Kanoni, Marcus E Kleber, Kati Kristiansson, Unhee Lim, Vaneet Lotay, Julian C Lui, Massimo Mangino, Irene Mateo Leach, Carolina Medina-Gomez, Michael A Nalls, Dale R Nyholt, Cameron D Palmer, Dorota Pasko, Sonali Pechlivanis, Inga Prokopenko, Janina S Ried, Stephan Ripke, Dmitry Shungin, Alena Stancáková, Rona J Strawbridge, Yun Ju Sung, Toshiko Tanaka, Alexander Teumer, Stella Trompet, Sander W van der Laan, Jessica van Setten, Jana V Van Vliet-Ostaptchouk, Zhaoming Wang, Loïc Yengo, Weihua Zhang, Uzma Afzal, Johan Ärnlöv, Gillian M Arscott, Stefania Bandinelli, Amy Barrett, Claire Bellis, Amanda J Bennett, Christian Berne, Matthias Blüher, Jennifer L Bolton, Yvonne Böttcher, Heather A Boyd, Marcel Bruinenberg, Brendan M Buckley, Steven Buyske, Ida H Caspersen, Peter S Chines, Robert Clarke, Simone Claudi-Boehm, Matthew Cooper, E Warwick Daw, Pim A De Jong, Joris Deelen, Graciela Delgado, Josh C Denny, Rosalie Dhonukshe-Rutten, Maria Dimitriou, Alex S F Doney, Marcus Dörr, Niina Eklund, Elodie Eury, Lasse Folkersen, Melissa E Garcia, Frank Geller, Vilmantas Giedraitis, Alan S Go, Harald Grallert, Tanja B Grammer, Jürgen Gräßler, Henrik Grönberg, Lisette C P G M de Groot, Christopher J Groves, Jeffrey Haessler, Per Hall, Toomas Haller, Goran Hallmans, Anke Hannemann, Catharina A Hartman, Maija Hassinen, Caroline Hayward, Nancy L Heard-Costa, Quinta Helmer, Gibran Hemani, Anjali K Henders, Hans L Hillege, Mark A Hlatky, Wolfgang Hoffmann, Per Hoffmann, Oddgeir Holmen, Jeanine J Houwing-Duistermaat, Thomas Illig, Aaron Isaacs, Alan L James, Janina Jeff, Berit Johansen, Åsa Johansson, Jennifer Jolley, Thorhildur Juliusdottir, Juhani Junttila, Abel N Kho, Leena Kinnunen, Norman Klopp, Thomas Kocher, Wolfgang Kratzer, Peter Lichtner, Lars Lind, Jaana Lindström, Stéphane Lobbens, Mattias Lorentzon, Yingchang Lu, Valeriya Lyssenko, Patrik K E Magnusson, Anubha Mahajan, Marc Maillard, Wendy L McArdle, Colin A McKenzie, Stela McLachlan, Paul J McLaren, Cristina Menni, Sigrun Merger, Lili Milani, Alireza Moayyeri, Keri L Monda, Mario A Morken, Gabriele Müller, Martina Müller-Nurasyid, Arthur W Musk, Narisu Narisu, Matthias Nauck, Ilja M Nolte, Markus M Nöthen, Laticia Oozageer, Stefan Pilz, Nigel W Rayner, Frida Renstrom, Neil R Robertson, Lynda M Rose, Ronan Roussel, Serena Sanna, Hubert Scharnagl, Salome Scholtens, Fredrick R Schumacher, Heribert Schunkert, Robert A Scott, Joban Sehmi, Thomas Seufferlein, Jianxin Shi, Karri Silventoinen, Johannes H Smit, Albert Vernon Smith, Joanna Smolonska, Alice V Stanton, Kathleen Stirrups, David J Stott, Heather M Stringham, Johan Sundström, Morris A Swertz, Ann-Christine Syvänen, Bamidele O Tayo, Gudmar Thorleifsson, Jonathan P Tyrer, Suzanne van Dijk, Natasja M van Schoor, Nathalie van der Velde, Diana van Heemst, Floor V A van Oort, Sita H Vermeulen, Niek Verweij, Judith M Vonk, Lindsay L Waite, Melanie Waldenberger, Roman Wennauer, Lynne R Wilkens, Christina Willenborg, Tom Wilsgaard, Mary K Wojczynski, Andrew Wong, Alan F Wright, Qunyuan Zhang, Dominique Arveiler, Stephan J L Bakker, John Beilby, Richard N Bergman, Sven Bergmann, Reiner Biffar, John Blangero, Dorret I Boomsma, Stefan R Bornstein, Pascal Bovet, Paolo Brambilla, Morris J Brown, Harry Campbell, Mark J Caulfield, Aravinda Chakravarti, Rory Collins, Francis S Collins, Dana C Crawford, L Adrienne Cupples, John Danesh, Ulf

de Faire, Hester M den Ruijter, Raimund Erbel, Jeanette Erdmann, Johan G Eriksson, Martin Farrall, Ele Ferrannini, Jean Ferrières, Ian Ford, Nita G Forouhi, Terrence Forrester, Ron T Gansevoort, Pablo V Gejman, Christian Gieger, Alain Golay, Omri Gottesman, Vilmundur Gudnason, Ulf Gyllensten, David W Haas, Alistair S Hall, Tamara B Harris, Andrew T Hattersley, Andrew C Heath, Christian Hengstenberg, Andrew A Hicks, Lucia A Hindorff, Aroon D Hingorani, Albert Hofman, G Kees Hovingh, Steve E Humphries, Steven C Hunt, Elina Hypponen, Kevin B Jacobs, Marjo-Riitta Jarvelin, Pekka Jousilahti, Antti M Jula, Jaakko Kaprio, John J P Kastelein, Manfred Kayser, Frank Kee, Sirkka M Keinanen-Kiukaanniemi, Lambertus A Kiemeney, Jaspal S Kooner, Charles Kooperberg, Seppo Koskinen, Peter Kovacs, Aldi T Kraja, Meena Kumari, Johanna Kuusisto, Timo A Lakka, Claudia Langenberg, Loic Le Marchand, Terho Lehtimäki, Sara Lupoli, and Pamela A F Madden. Defining the role of common variation in the genomic and biological architecture of adult human height. *Nature Genetics*, 46(11):1173–1186, 2014.

[15] Joel N. Hirschhorn and Mark J. Daly. Genome-wide association studies for common diseases and complex traits. *Nature Reviews Genetics*, 6(2):95–108, 2005.

[16] L. J. Scott, K. L. Mohlke, L. L. Bonnycastle, C. J. Willer, Y. Li, W. L. Duren, M. R. Erdos, H. M. Stringham, P. S. Chines, A. U. Jackson, L. Prokunina-Olsson, C. J. Ding, A. J. Swift, N. Narisu, T. Hu, R. Pruim, R. Xiao, X. Y. Li, K. N. Conneely, N. L. Riebow, A. G. Sprau, M. Tong, P. P. White, K. N. Hetrick, M. W. Barnhart, C. W. Bark, J. L. Goldstein, L. Watkins, F. Xiang, J. Saramies, T. A. Buchanan, R. M. Watanabe, T. T. Valle, L. Kinnunen, G. R. Abecasis, E. W. Pugh, K. F. Doheny, R. N. Bergman, J. Tuomilehto, F. S. Collins, and M. Boehnke. A genome-wide association study of type 2 diabetes in finns detects multiple susceptibility variants. *Science*, 316(5829):1341–5, 2007.

[17] Nadav Brandes, Omer Weissbrod, and Michal Linial. Open problems in human trait genetics. *Genome Biology*, 23(1):131, 2022.

[18] Peter M. Visscher, Naomi R. Wray, Qian Zhang, Pamela Sklar, Mark I. McCarthy, Matthew A. Brown, and Jian Yang. 10 years of gwas discovery: Biology, function, and translation. *The American Journal of Human Genetics*, 101(1):5–22, 2017.

[19] Frank Dudbridge. Power and predictive accuracy of polygenic risk scores. *PLOS Genetics*, 9(3):1–17, 03 2013.

[20] Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat Rev Genet*, 11(10):733–739, Oct 2010.

[21] Lon R Cardon and Lyle J Palmer. Population stratification and spurious allelic association. *The Lancet*, 361(9357):598–604, 2003.

[22] Jonathan Marchini, Lon R. Cardon, Michael S. Phillips, and Peter Donnelly. The effects of human population structure on large genetic association studies. *Nat Genet*, 36(5):512–517, May 2004.

[23] Jae Hoon Sul, Lana S Martin, and Eleazar Eskin. Population structure in genetic studies: Confounding factors and mixed models. *PLoS Genet*, 14(12):e1007309, Dec 2018.

[24] W. Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 8/12/2022 2007.

[25] David H. Alexander, John Novembre, and Kenneth Lange. Fast model-based estimation of ancestry in unrelated individuals. *Genome Research*, 19(9):1655–1664, 2009.

[26] Yuqing Zhang, Giovanni Parmigiani, and W Evan Johnson. Combat-seq: batch effect adjustment for rna-seq count data. *NAR Genomics and Bioinformatics*, 2(3):lqaa078, 8/12/2022 2020.

[27] Alkes L Price, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38(8):904–909, 2006.

[28] Abdel Abdellaoui, Karin J.H. Verweij, and Michel G. Nivard. Geographic confounding in genome-wide association studies. *bioRxiv*, 2021.

[29] Mashaal Sohail, Robert M Maier, Andrea Ganna, Alex Bloemendal, Alicia R Martin, Michael C Turchin, Charleston WK Chiang, Joel Hirschhorn, Mark J Daly, Nick Patterson, Benjamin Neale, Iain Mathieson, David Reich, and Shamil R Sunyaev. Polygenic adaptation on height is overestimated due to uncorrected stratification in genome-wide association studies. *eLife*, 8:e39702, mar 2019.

[30] Nathalie Pochet, Frank De Smet, Johan A. K. Suykens, and Bart L. R. De Moor. Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction. *Bioinformatics*, 20(17):3185–3195, 8/15/2022 2004.

[31] Vivien Marx. The big challenges of big data. *Nature*, 498(7453):255–260, 2013.

[32] Oriol Canela-Xandri, Andy Law, Alan Gray, John A Woolliams, and Albert Tenesa. A new tool called dissect for analysing large genomic data sets using a big data approach. *Nature communications*, 6:10162, 2015.

[33] Inès Krissaane, Carlos De Niz, Alba Gutiérrez-Sacristán, Gabor Korodi, Nneka Ede, Ranjay Kumar, Jessica Lyons, Arjun Manrai, Chirag Patel, Isaac Kohane, and Paul Avillach. Scalability and cost-effectiveness analysis of whole genome-wide association studies on google cloud platform and amazon web services. *Journal of the American Medical Informatics Association*, 27(9):1425–1430, 8/12/2022 2020.

[34] Edo Liberty and Steven W Zucker. The mailman algorithm: A note on matrix–vector multiplication. *Information Processing Letters*, 109(3):179–182, 2009.

[35] Iain Mathieson and Gil McVean. Differential confounding of rare and common variants in spatially structured populations. *Nature genetics*, 44(3):243, 2012.

[36] Garrett Hellenthal, Adam Auton, and Daniel Falush. Inferring human colonization history using a copying model. *PLoS Genet*, 4(5):e1000078, 05 2008.

[37] Irineo Cabreros and John D. Storey. A likelihood-free estimator of population structure bridging admixture models and principal components analysis. *Genetics*, 212(4):1009–1029, 2019.

[38] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[39] W Li, JE Cerise, Y Yang, and H Han. Application of t-sne to human genetic data. *J Bioinform Comput Biol.*, 15(4):1750017, 2017.

[40] E Becht, L McInnes, J Healy, CA Dutertre, IWH Kwok, LG Ng, F Ginhoux, and EW Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nat Biotechnol.*, 37:38–44, 2019.

[41] Peter Langfelder and Steve Horvath. Wgcna: an r package for weighted correlation network analysis. *BMC Bioinformatics*, 9:559, 2008.

[42] John Novembre and Sohini Ramachandran. Perspectives on human population structure at the cusp of the sequencing era. *Annual review of genomics and human genetics*, 12:245–274, 2011.

[43] J Novembre, T Johnson, K Bryc, Z Kutalik, AR Boyko, A Auton, A Indap, KS King, S Bergmann, MR Nelson, M Stephens, and Bustamante CD. Genes mirror geography within europe. *Nature*, 456(7219):274, 2008.

[44] Wen-Yun Yang, John Novembre, Eleazar Eskin, and Eran Halperin. A model-based approach for analysis of spatial structure in genetic data. *Nature genetics*, 44(6):725–731, 2012.

[45] Yael Baran, Inés Quintela, Ángel Carracedo, Bogdan Pasaniuc, and Eran Halperin. Enhanced localization of genetic samples through linkage-disequilibrium correction. *The American Journal of Human Genetics*, 92(6):882–894, 2013.

[46] Alkes L Price, Noah A Zaitlen, David Reich, and Nick Patterson. New approaches to population stratification in genome-wide association studies. *Nature reviews. Genetics*, 11(7):459, 2010.

[47] Nick Patterson, Alkes L. Price, and David Reich. Population structure and eigenanalysis. *PLoS Genetics*, 2(12):e190+, December 2006.

[48] C. L. Hanis, R. Chakraborty, R. E. Ferrell, and W. J. Schull. Individual admixture estimates: disease associations and individual risk of diabetes and gallbladder disease among mexican-americans in starr county, texas. *Am J Phys Anthropol*, 70(4):433–441, August 1986.

[49] J. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959, 2000.

[50] Chibiao Chen, Eric Durand, Florence Forbes, and Olivier François. Bayesian clustering algorithms ascertaining spatial population structure: a new computer program and a comparison study. *Molecular Ecology Resources*, 7(5):747–756, 2007.

[51] Barbara E Engelhardt and Matthew Stephens. Analysis of population structure: a unifying framework and novel methods based on sparse factor analysis. *PLoS genetics*, 6(9):e1001117, 2010.

[52] Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.

[53] Kevin J Galinsky, Gaurav Bhatia, Po-Ru Loh, Stoyan Georgiev, Sayan Mukherjee, Nick J Patterson, and Alkes L Price. Fast principal-component analysis reveals convergent evolution of adh1b in europe and east asia. *The American Journal of Human Genetics*, 98(3):456–472, 2016.

[54] Gad Abraham, Yixuan Qiu, and Michael Inouye. Flashpca2: principal component analysis of biobank-scale genotype datasets. *Bioinformatics*, 2017.

[55] F Prive, H Aschard, A Ziyatdinov, and MGB Blum. Efficient analysis of large-scale genome-wide data with two r packages: bigstatsr and bigsnpr. *Bioinformatics.*, 34(16):2781–2787, 2018.

[56] Aritra Bose, Vassilis Kalantzis, Eugenia-Maria Kontopoulou, Mai Elkady, Peristera Paschou, and Petros Drineas. Terapca: a fast and scalable software package to study genetic variation in tera-scale genotypes. *Bioinformatics.*, 35(19):3679–3683, 2019.

[57] CC Chang, CC Chow, LC Tellier, S Vattikuti, SM Purcell, and JJ Lee. Second-generation plink: rising to the challenge of larger and richer datasets. *Gigascience.*, 4:7, 2015.

[58] A. Price, N. Patterson, R. Plenge, M. Weinblatt, N. Shadick, and D. Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38:904–909, 2006.

[59] Sam T Roweis. Em algorithms for pca and spca. In *Advances in neural information processing systems*, pages 626–632, 1998.

[60] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

[61] Theodore W Anderson and Herman Rubin. Statistical inference in factor analysis. In *Proceedings of the third Berkeley symposium on mathematical statistics and probability*, volume 5, pages 111–150, 1956.

[62] Arthur Szlam, Andrew Tulloch, and Mark Tygert. Accurate low-rank approximations via a few iterations of alternating least squares. *SIAM Journal on Matrix Analysis and Applications*, 38(2):425–433, 2017.

[63] Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel AR Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul IW De Bakker, Mark J Daly, et al. Plink: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007.

[64] Richard B Lehoucq and Danny C Sorensen. Deflation techniques for an implicitly restarted arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications*, 17(4):789–821, 1996.

[65] A Manichaikul, JC Mychaleckyj, SS Rich, K Daly, M Sale, and WM Chen. Robust relationship inference in genome-wide association studies. *Bioinformatics*, 26(22):2867–2873, 2010.

[66] The 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65, Nov 2012.

[67] Mark D Shriver, Giulia C Kennedy, Esteban J Parra, Heather A Lawson, Vibhor Sonpar, Jing Huang, Joshua M Akey, and Keith W Jones. The genomic distribution of population substructure in four populations using 8,525 autosomal snps. *Human genomics*, 1(4):274, 2004.

[68] Chao Tian, Robert M Plenge, Michael Ransom, Annette Lee, Pablo Villoslada, Carlo Selmi, Lars Klareskog, Ann E Pulver, Lihong Qi, Peter K Gregersen, et al. Analysis and application of european genetic substructure using 300 k snp information. *PLoS genetics*, 4(1):e4, 2008.

[69] A Wiegering, U Ruther, and C Gerhardt. The ciliary protein rpgrip1l in development and disease. *Dev Biol*, 442(1):60–68, 2018.

[70] Marion Delous, Lekbir Baala, Rémi Salomon, Christine Laclef, Jeanette Vierkotten, Kàlmàn Tory, Christelle Golzio, Tiphanie Lacoste, Laurianne Besse, Catherine Ozilou, Imane Moutkine, Nathan E Hellman, Isabelle Anselme, Flora Silbermann, Christine Vesque, Christoph Gerhardt, Eleanor Rattenberry, Matthias T F Wolf, Marie Claire Gubler, Jéléna Martinovic, Féréchté Encha-Razavi, Nathalie Boddaert, Marie Gonzales, Marie Alice Macher, Hubert Nivet, Gérard Champion, Jean Pierre Berthélémé, Patrick Niaudet, Fiona McDonald, Friedhelm Hildebrandt, Colin A Johnson, Michel Vekemans, Corinne Antignac, Ulrich Rüther, Sylvie Schneider-Maunoury, Tania Attié-Bitach, and Sophie Saunier. The ciliary gene rpgrip1l is mutated in cerebello-oculo-renal syndrome (joubert syndrome type b) and meckel syndrome. *Nature Genetics*, 39:875–881, 2007.

[71] Oliver Devuyst and Veronique J. Arnould. Mutations in rpgrip1l : extending the clinical spectrum of ciliopathies. *Nephrology Dialysis Transplantation*, 23(5):1500–1503, 2008.

[72] Hemant Khanna, Erica E Davis, Carlos A Murga-Zamalloa, Alejandro Estrada-Cuzcano, Irma Lopez, Anneke I den Hollander, Marijke N Zonneveld, Mohammad I Othman, Naushin Waseem, Christina F Chakarova, Cecilia Maubaret, Anna Diaz-Font, Ian MacDonald, Donna M Muzny, David A Wheeler, Margaret Morgan, Lora R Lewis, Clare V Logan, Perciliz L Tan, Michael A Beer, Chris F Inglehearn, Richard A Lewis, Samuel G Jacobson, Carsten Bergmann, Philip L Beales, Tania Attié-Bitach, Colin A Johnson, Edgar A Otto, Shomi S Bhattacharya, Friedhelm Hildebrandt, Richard A Gibbs, Robert K Koenekoop, Anand Swaroop, and Nicholas Katsanis. A common allele in rpgrip1l is a modifier of retinal degeneration in ciliopathies. *Nature Genetics*, 41(6):739–45, 2009.

[73] H Aschard, BJ Vilhjálmsson, N Greliche, PE Morange, DA Trégouët, and P Kraft. Maximizing the power of principal-component analysis of correlated phenotypes in genome-wide association studies. *AJHG*, 94(5):662–76, 2014.

[74] KV Korneev, KN Atretkhany, MS Drutskaya, SI Grivennikov, DV Kuprash, and SA Nedospasov. Tlr-signaling and proinflammatory cytokines as drivers of tumorigenesis. *Cytokine*, 89:127–135, 2017.

[75] FP Mockenhaupt, JP Cramer, L Hamann, MS Stegemann, J Eckert, NR Oh, RN Otchwemah, E Dietz, S Ehrhardt, NW Schröder, U Bienzle, and RR Schumann. Toll-like receptor (tlr) polymorphisms in african children: Common tlr-4 variants predispose to severe malaria. *PNAS*, 103(1):177–182, 2006.

[76] CA Van der Graaf, MG Netea, SA Morré, M Den Heijer, PE Verweij, JW Van der Meer, and BJ Kullberg. Toll-like receptor 4 asp299gly/thr399ile polymorphisms are a risk factor for candida bloodstream infection. *European Cytokine Network*, 17(1):29–34, 2006.

[77] Yair Field, Evan A Boyle, Natalie Telis, Ziyue Gao, Kyle J. Gaulton, David Golan, Loic Yengo, Ghislain Rocheleau, Philippe Froguel, Mark I. McCarthy, and Jonathan K. Pritchard. Detection of human adaptation during the past 2000 years. *Science*, 354(6313):760–764, 2016.

[78] Albers and McVean. Dating genomic variants and shared ancestry in population-scale sequencing data. *bioRxiv*, 2019.

[79] Yue Wu and Sriram Sankararaman. A scalable estimator of snp heritability for biobank-scale data. *Bioinformatics*, 34(13):i187–i194, 2018.

[80] Na Li and Matthew Stephens. Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data. *Genetics*, 165(4):2213–2233, 2003.

[81] Xiaoquan Wen and Matthew Stephens. Using linear predictors to impute allele frequencies from summary or pooled genotype data. *The annals of applied statistics*, 4(3):1158, 2010.

[82] Andrew I Schein, Lawrence K Saul, and Lyle H Ungar. A generalized linear model for principal component analysis of binary data. In *AISTATS*, volume 3, page 10, 2003.

[83] Jade Yu Cheng, Thomas Mailund, and Rasmus Nielsen. Fast admixture analysis and population tree estimation for SNP and NGS data. *Bioinformatics*, 33(14):2148–2155, 01 2017.

[84] Anil Raj, Matthew Stephens, and Jonathan K Pritchard. faststructure: variational inference of population structure in large snp data sets. *Genetics*, 197(2):573–589, 2014.

[85] Prem Gopalan, Wei Hao, David M Blei, and John D Storey. Scaling probabilistic models of genetic variation to millions of humans. *Nature Genetics*, 48:1587–1590, 11 2016.

[86] X. Chen and J. D. Storey. Consistent estimation of low dimensional latent structure in high-dimensional data. *arXiv*, page 1510.03497v1, 2015.

[87] 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491:56–65, 2012.

[88] Aaron A Behr, Katherine Z Liu, Gracie Liu-Fang, Priyanka Nakka, and Sohini Ramachandran. Pong: fast analysis and visualization of latent clusters in population genetic data. *Bioinformatics*, 32(18):2817–2823, 2016.

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

[90] D. J. Balding and R. A. Nichols. A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. *Genetica*, 96:3–12, 1995.

[91] Alejandro Ochoa and John D Storey. Estimating fst and kinship for arbitrary population structures. *PLoS genetics*, 17(1):e1009241–e1009241, 01 2021.

[92] I. et al. Lazardis. Ancient human genomes suggest three ancestral populations for present-day europeans. *Nature*, 513:409–413, 2014.

[93] H.M. et al. Cann. A human genome diversity cell line panel. *Science*, 296:261–262, 2002.

[94] L.L. Cavalli-Sforza. The human genome diversity project: past, present and future. *Nat. Rev. Genet.*, 6:333–340, 2005.

[95] N.A. Rosenberg. Standardized subsets of the hgdp-ceph human genome diversity cell line panel, accounting for atypical and duplicated samples and pairs of close relatives. *Ann. Hum. Genet.*, 70:841–847, 2006.

[96] Marek Gagolewski, Maciej Bartoszuk, and Anna Cena. Genie: A new, fast, and outlier-resistant hierarchical clustering algorithm. *Information Sciences*, 363:8–23, 2016.

[97] Eric Frichot, François Mathieu, Théo Trouillon, Guillaume Bouchard, and Olivier François. Fast and efficient estimation of individual ancestry coefficients. *Genetics*, 196(4):973–983, 2014.

[98] Jun Z. Li, Devin M. Absher, Hua Tang, Audrey M. Southwick, Amanda M. Casto, Sohini Ramachandran, Howard M. Cann, Gregory S. Barsh, Marcus Feldman, Luigi L. Cavalli-Sforza, and Richard M. Myers. Worldwide Human Relationships Inferred from Genome-Wide Patterns of Variation. *Science*, 319(5866):1100–1104, 2008.

[99] Liat Shenhav, Mike Thompson, Tyler A. Joseph, Leah Briscoe, Ori Furman, David Bogumil, Itzhak Mizrahi, Itsik Pe'er, and Eran Halperin. Feast: fast expectation-maximization for microbial source tracking. *Nature Methods*, 16(7):627–632, 2019.

[100] Christa Caggiano, Barbara Celona, Fleur Garton, Joel Mefford, Brian Black, Catherine Lomen-Hoerth, Andrew Dahl, and Noah Zaitlen. Estimating the rate of cell type degeneration from epigenetic sequencing of cell-free dna. *bioRxiv*, 2020.

[101] Daniel John Lawson, Garrett Hellenthal, Simon Myers, and Daniel Falush. Inference of population structure using dense haplotype data. *PLOS Genetics*, 8(1):1–16, 01 2012.

[102] Alex Diaz-Papkovich, Luke Anderson-Trocmé, Chief Ben-Eghan, and Simon Gravel. Umap reveals cryptic population structure and phenotype heterogeneity in large genomic cohorts. *PLOS Genetics*, 15(11):1–24, 11 2019.

[103] Juba Nait Saada et al. Identity-by-descent detection across 487,409 british samples reveals fine scale population structure and ultra-rare variant associations. *Nature Communications*, 11(1), 2020.

[104] Wohns A.W. et al. Kelleher J., Wong Y. Inferring whole-genome histories in large population datasets. *Nat Genet*, 51:1330–1338, 2019.

[105] Arjun Raj, Scott A Rifkin, Erik Andersen, and Alexander van Oudenaarden. Variability in gene expression underlies incomplete penetrance. *Nature*, 463(7283):913–918, Feb 2010.

[106] Yu Hasegawa, Deanne Taylor, Dmitry A Ovchinnikov, Ernst J Wolvetang, Laurence de Torrenté, and Jessica C Mar. Variability of gene expression identifies transcriptional regulators of early human embryonic development. *PLoS Genet*, 11(8):e1005428, Aug 2015.

[107] Tristan V. de Jong, Yuri M. Moshkin, and Victor Guryev. Gene expression variability: the other dimension in transcriptome analysis. *Physiological Genomics*, 51(5):145–158, 2019.

[108] David J Green, Shalaw R Sallah, Jamie M Ellingford, Simon C Lovell, and Panagiotis I Sergouniotis. Variability in gene expression is associated with incomplete penetrance in inherited eye disorders. *Genes (Basel)*, 11(2), Feb 2020.

[109] Anna A. Igolkina, Chris Armoskus, Jeremy R. B. Newman, Oleg V. Evgrafov, Lauren M. McIntyre, Sergey V. Nuzhdin, and Maria G. Samsonova. Analysis of gene expression variance in schizophrenia using structural equation modeling. *Frontiers in Molecular Neuroscience*, 11, 2018.

[110] Fuquan Zhang, Yin Yao Shugart, Weihua Yue, Zaohuo Cheng, Guoqiang Wang, Zhenhe Zhou, Chunhui Jin, Jianmin Yuan, Sha Liu, and Yong Xu. Increased variability of genomic transcription in schizophrenia. *Scientific Reports*, 5(1):17995, 2015.

[111] Jessica C Mar, Nicholas A Matigian, Alan Mackay-Sim, George D Mellick, Carolyn M Sue, Peter A Silburn, John J McGrath, John Quackenbush, and Christine A Wells. Variance of gene expression identifies altered network constraints in neurological disease. *PLoS Genet*, 7(8):e1002207, Aug 2011.

[112] Simone Ecker, Vera Pancaldi, Daniel Rico, and Alfonso Valencia. Higher gene expression variability in the more aggressive subtype of chronic lymphocytic leukemia. *Genome Medicine*, 7(1):8, 2015.

[113] Luise Wolf, Olin K Silander, and Erik van Nimwegen. Expression noise facilitates the evolution of gene regulation. *eLife*, 4:e05856, jun 2015.

[114] Zoltán Bódi, Zoltán Farkas, Dmitry Nevozhay, Dorottya Kalapis, Viktória Lázár, Bálint Csörgő, Ákos Nyerges, Béla Szamecz, Gergely Fekete, Balázs Papp, Hugo Araújo, José L. Oliveira, Gabriela Moura, Manuel A. S. Santos, Tamás Székely Jr, Gábor Balázsi, and Csaba Pál. Phenotypic heterogeneity promotes adaptive evolution. *PLOS Biology*, 15(5):1–26, 05 2017.

[115] Eyal Simonovsky, Ronen Schuster, and Esti Yeger-Lotem. Large-scale analysis of human gene expression variability associates highly variable drug targets with lower drug effectiveness and safety. *Bioinformatics*, 35(17):3028–3037, 01 2019.

[116] Dennis Kostka and Rainer Spang. Finding disease specific alterations in the co-expression of genes. *Bioinformatics*, 20:i194–i199, 08 2004.

[117] Lea A et al. Genetic and environmental perturbations lead to regulatory decoherence. *eLife*, 8:e40538, 2019.

[118] Jung Kyoon Choi, Ungsik Yu, Ook Joon Yoo, and Sangsoo Kim. Differential coexpression analysis using microarray data and its application to human cancer. *Bioinformatics*, 21(24):4348–4355, 10 2005.

[119] Ed Reznik and Chris Sander. Extensive decoupling of metabolic genes in cancer. *PLoS Comput Biol*, 11(5):e1004176, May 2015.

[120] Emma L Bailey, Martin W McBride, Wendy Beattie, John D McClure, Delyth Graham, Anna F Dominiczak, Cathie L M Sudlow, Colin Smith, and Joanna M Wardlaw. Differential gene expression in multiple neurological, inflammatory and connective tissue pathways in a spontaneous model of human small vessel stroke. *Neuropathol Appl Neurobiol*, 40(7):855–872, Dec 2014.

[121] C. Gaiteri, Y. Ding, B. French, G. C. Tseng, and E. Sibille. Beyond modules and hubs: the potential of gene coexpression networks for investigating molecular mechanisms of complex brain disorders. *Genes, Brain and Behavior*, 13(1):13–24, 2014.

[122] Simone Ecker, Lu Chen, Vera Pancaldi, Frederik O. Bagger, JoséMaría Fernández, Enrique Carrillo de Santa Pau, David Juan, Alice L. Mann, Stephen Watt, Francesco Paolo Casale, Nikos Sidiropoulos, Nicolas Rapin, Angelika Merkel, Cornelis A. Albers, Vyacheslav Amstislavskiy, Sofie Ashford, Lorenzo Bomba, David Bujold, Frances Burden, Stephan Busche, Maxime Caron, Shu-Huang Chen, Warren A. Cheung, Laura Clarke, Irina Colgiu, Avik Datta, Oliver Delaneau, Heather Elding, Samantha Farrow, Diego Garrido-Martín, Bing Ge, Roderic Guigo, Valentina Iotchkova, Kousik Kundu, Tony Kwan, John J. Lambourne, Ernesto Lowy, Daniel Mead, Farzin Pourfarzad, Adriana Redensek, Karola Rehnstrom, Augusto Rendon, David Richardson, Thomas Risch, Sophia Rowlston, Xiaojian Shao, Marie-Michelle Simon, Marc Sultan, Klaudia Walter, Steven P. Wilder, Ying Yan, Stylianos E. Antonarakis, Guillaume Bourque, Emmanouil T. Dermitzakis, Paul Flicek, Hans Lehrach, Joost H. A. Martens, Marie-Laure Yaspo, Willem H. Ouwehand, Hendrik G. Stunnenberg, Oliver Stegle, Mattia Frontini, Kate Downes, Tomi Pastinen, Taco W. Kuijpers, Daniel Rico, Alfonso Valencia, Stephan Beck, Nicole Soranzo, Dirk S. Paul, and BLUEPRINT Consortium. Genome-wide analysis of differential transcriptional and epigenetic variability across human immune cell types. *Genome Biology*, 18(1):18, 2017.

[123] Anders S Love MI, Huber W. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biology*, 15:550, 2014.

[124] Smyth GK McCarthy DJ, Chen Y. Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic Acids Research*, 40(10):4288–4297, 2012.

[125] Di Ran and Z John Daye. Gene expression variability and the analysis of large-scale rna-seq studies with the mdseq. *Nucleic Acids Res*, 45(13):e127, Jul 2017.

[126] Atsushi Fukushima. Diffcorr: An r package to analyze and visualize differential correlations in biological networks. *Gene*, 518:209–214, 2013.

[127] Orly Alter, Patrick O. Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106, 2000.

[128] Steve Horvath and Jun Dong. Geometric interpretation of gene coexpression network analysis. *PLOS Computational Biology*, 4(8):1–27, 08 2008.

[129] Peter Langfelder and Steve Horvath. Eigengene networks for studying the relationships between co-expression modules. *BMC Systems Biology*, 1(1):54, 2007.

[130] James Baglama and Loath Reichel. Augmented implicitly restarted lanczos bidiagonalization methods. *SIAM Journal on Scientific Computing*, 27(1):19–42, 2005.

[131] Alan Genz and Frank Bretz. *Computation of Multivariate Normal and t Probabilities*. Lecture Notes in Statistics. Springer-Verlag, Heidelberg, 2009.

[132] Michael J Gandal, Jillian R Haney, Neelroop N Parikshak, Virpi Leppa, Gokul Ramaswami, Chris Hartl, Andrew J Schork, Vivek Appadurai, Alfonso Buil, Thomas M Werge, Chunyu Liu, Kevin P White, Steve Horvath, and Daniel H Geschwind. Shared molecular neuropathology across major psychiatric disorders parallels polygenic overlap. *Science*, 359(6376):693–697, Feb 2018.

[133] Antonio Colaprico, Tiago C. Silva, Catharina Olsen, Luciano Garofano, Claudia Cava, Davide Garolini, Thais S. Sabedot, Tathiane M. Malta, Stefano M. Pagnotta, Isabella Castiglioni, Michele Ceccarelli, Gianluca Bontempi, and Houtan Noushmehr. Tcgabiolinks: an r/bioconductor package for integrative analysis of tcga data. *Nucleic Acids Research*, 44(8):e71–e71, 8/10/2022 2016.

[134] John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, Barbara Foster, Mike Moser, Ellen Karasik, Bryan Gillard, Kimberley Ramsey, Susan Sullivan, Jason Bridge, Harold Magazine, John Syron, Johnelle Fleming, Laura Siminoff, Heather Traino, Maghboeba Mosavel, Laura Barker, Scott Jewell, Dan Rohrer, Dan Maxim, Dana Filkins, Philip Harbach, Eddie Cortadillo, Bree Berghuis, Lisa Turner, Eric Hudson, Kristin Feenstra, Leslie Sobin, James Robb, Phillip Branton, Greg Korzeniewski, Charles Shive, David Tabor, Liqun Qi, Kevin Groch, Sreenath Nampally, Steve Buia, Angela Zimmerman, Anna Smith, Robin Burges, Karna Robinson, Kim Valentino, Deborah Bradbury, Mark Cosentino, Norma Diaz-Mayoral, Mary Kennedy, Theresa Engel, Penelope Williams, Kenyon Erickson, Kristin Ardlie, Wendy Winckler, Gad Getz, David DeLuca, Daniel MacArthur, Manolis Kellis, Alexander Thomson, Taylor Young, Ellen Gelfand, Molly Donovan, Yan Meng, George Grant, Deborah Mash, Yvonne Marcus, Margaret Basile, Jun Liu, Jun Zhu, Zhidong Tu, Nancy J Cox, Dan L Nicolae, Eric R Gamazon, Hae Kyung Im, Anuar Konkashbaev, Jonathan Pritchard, Matthew Stevens, Timothèe Flutre, Xiaoquan Wen, Emmanouil T Dermitzakis, Tuuli Lappalainen, Roderic Guigo, Jean Monlong, Michael Sammeth, Daphne Koller, Alexis Battle, Sara Mostafavi, Mark McCarthy, Manual Rivas, Julian Maller, Ivan Rusyn, Andrew Nobel, Fred Wright, Andrey Shabalin, Mike Feolo, Nataliya Sharopova, Anne Sturcke, Justin Paschal, James M Anderson, Elizabeth L Wilder, Leslie K Derr, Eric D Green, Jeffery P Struewing, Gary Temple, Simona Volpi, Joy T Boyer, Elizabeth J Thomson, Mark S Guyer, Cathy Ng, Assya Abdallah, Deborah Colantuoni, Thomas R

Insel, Susan E Koester, A Roger Little, Patrick K Bender, Thomas Lehner, Yin Yao, Carolyn C Compton, Jimmie B Vaught, Sherilyn Sawyer, Nicole C Lockhart, Joanne Demchok, and Helen F Moore. The genotype-tissue expression (gtex) project. *Nature Genetics*, 45(6):580–585, 2013.

[135] T R Golub, D K Slonim, P Tamayo, C Huard, M Gaasenbeek, J P Mesirov, H Coller, M L Loh, J R Downing, M A Caligiuri, C D Bloomfield, and E S Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, Oct 1999.

[136] Peter Langfelder, Rui Luo, Michael C. Oldham, and Steve Horvath. Is my network module preserved and reproducible? *PLOS Computational Biology*, 7(1):1–29, 01 2011.

[137] Andrew T. McKenzie, Igor Katsyv, Won-Min Song, Minghui Wang, and Bin Zhang. Dgca: A comprehensive r package for differential gene correlation analysis. *BMC Systems Biology*, 10(1):106, 2016.

[138] Steve Horvath. Dna methylation age of human tissues and cell types. *Genome Biol*, 14(10):R115, 2013.

[139] Lin Cheng, Bao Sun, Yan Xiong, Lei Hu, Lichen Gao, Ji Li, Hongfu Xie, Xiaoping Chen, Wei Zhang, and Hong-Hao Zhou. Wgcna-based dna methylation profiling analysis on allopurinol-induced severe cutaneous adverse reactions: A dna methylation signature for predisposing drug hypersensitivity. *J Pers Med*, 12(4), Mar 2022.

[140] Jessie Nicodemus-Johnson, Rachel A Myers, Noburu J Sakabe, Debora R Sobreira, Douglas K Hogarth, Edward T Naureckas, Anne I Sperling, Julian Solway, Steven R White, Marcelo A Nobrega, Dan L Nicolae, Yoav Gilad, and Carole Ober. Dna methylation in lung cells is associated with asthma endotypes and genetic risk. *JCI Insight*, 1(20):e90151, Dec 2016.

[141] Kristel R. van Eijk, Simone de Jong, Marco PM Boks, Terry Langeveld, Fabrice Colas, Jan H. Veldink, Carolien GF de Kovel, Esther Janson, Eric Strengman, Peter Langfelder, RenéS. Kahn, Leonard H. van den Berg, Steve Horvath, and Roel A. Ophoff. Genetic analysis of dna methylation and gene expression levels in whole blood of healthy human subjects. *BMC Genomics*, 13(1):636, 2012.

[142] Milena Gasco, Shukri Shami, and Tim Crook. The p53 pathway in breast cancer. *Breast Cancer Res*, 4(2):70–76, 2002.

[143] P. Yang, C. W. Du, M. Kwan, S. X. Liang, and G. J. Zhang. The impact of p53 in predicting clinical outcome of breast cancer patients with visceral metastasis. *Scientific Reports*, 3(1):2246, 2013.

[144] Xuanwen Bao, Natasa Anastasov, Yanfang Wang, and Michael Rosemann. A novel epigenetic signature for overall survival prediction in patients with breast cancer. *Journal of Translational Medicine*, 17(1):380, 2019.

[145] Søren Kristiansen, Lars M. Jørgensen, Per Guldberg, and György Sölétormos. Aberrantly methylated dna as a biomarker in breast cancer. *The International Journal of Biological Markers*, 28(2):141–150, 2022/08/10 2013.

[146] M S Breen, A X Maihofer, S J Glatt, D S Tylee, S D Chandler, M T Tsuang, V B Risbrough, D G Baker, D T O'Connor, C M Nievergelt, and C H Woelk. Gene networks specific for innate immunity define post-traumatic stress disorder. *Mol Psychiatry*, 20(12):1538–1545, Dec 2015.

[147] Steve Horvath and Jun Dong. Geometric interpretation of gene coexpression network analysis. *PLoS Comput Biol*, 4(8):e1000117, Aug 2008.

[148] Abhishek K. Sarkar, Po-Yuan Tung, John D. Blischak, Jonathan E. Burnett, Yang I. Li, Matthew Stephens, and Yoav Gilad. Discovery and characterization of variance qtls in human induced pluripotent stem cells. *PLOS Genetics*, 15(4):1–16, 04 2019.

[149] Robert W Corty and William Valdar. vqtl: An R Package for Mean-Variance QTL Mapping. *G3 Genes|Genomes|Genetics*, 8(12):3757–3766, 12 2018.

[150] Jingwen Gan, Yige Cao, Libo Jiang, and Rongling Wu. Mapping covariation quantitative trait loci that control organ growth and whole-plant biomass. *Front Plant Sci*, 10:719, 2019.

[151] Daniel J. Kliebenstein, Marilyn AL West, Hans van Leeuwen, Olivier Loudet, RW Doerge, and Dina A. St Clair. Identification of qtls controlling gene expression networks defined a priori. *BMC Bioinformatics*, 7(1):308, 2006.

[152] L Kruglyak and E S Lander. A nonparametric approach for mapping quantitative trait loci. *Genetics*, 139(3):1421–1428, Mar 1995.

[153] Eric E. Schadt, Stephanie A. Monks, Thomas A. Drake, Aldons J. Lusis, Nam Che, Veronica Colinayo, Thomas G. Ruff, Stephen B. Milligan, John R. Lamb, Guy Cavet, Peter S. Linsley, Mao Mao, Roland B. Stoughton, and Stephen H. Friend. Genetics of gene expression surveyed in maize, mouse and man. *Nature*, 422(6929):297–302, 2003.

[154] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, and Jill P Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, 102(43):15545–15550, Oct 2005.

[155] Vamsi K Mootha, Cecilia M Lindgren, Karl-Fredrik Eriksson, Aravind Subramanian, Smita Sihag, Joseph Lehar, Pere Puigserver, Emma Carlsson, Martin Ridderstråle, Esa Laurila, Nicholas Houstis, Mark J Daly, Nick Patterson, Jill P Mesirov, Todd R Golub, Pablo Tamayo, Bruce Spiegelman, Eric S Lander, Joel N Hirschhorn, David Altshuler, and Leif C Groop. Pgc-1a-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, 34(3):267–273, 2003.

[156] Raymond B. Cattell. The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2):245–276, 04 1966.

[157] Pedro R. Peres-Neto, Donald A. Jackson, and Keith M. Somers. How many principal components? stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics & Data Analysis*, 49(4):974–997, 2005.

[158] Lars Kai Hansen, Jan Larsen, Finn Årup Nielsen, Stephen C. Strother, Egill Rostrup, Robert Savoy, Nicholas Lange, John Sidtis, Claus Svarer, and Olaf B. Paulson. Generalizable patterns in neuroimaging: How many principal components? *NeuroImage*, 9(5):534–544, 1999.

[159] Aman Agrawal, Alec M. Chiu, Minh Le, Eran Halperin, and Sriram Sankararaman. Scalable probabilistic pca for large-scale genetic variation data. *PLOS Genetics*, 16(5):1–19, 05 2020.

[160] Florian Privé, Keurcien Luu, Bjarni J Vilhjálmsson, and Michael G B Blum. Performing Highly Efficient Genome Scans for Local Adaptation with R Package pcadapt Version 4. *Molecular Biology and Evolution*, 37(7):2153–2154, 04 2020.

[161] Hugues Aschard, Bjarni J Vilhjálmsson, Nicolas Greliche, Pierre-Emmanuel Morange, David-Alexandre Trégouët, and Peter Kraft. Maximizing the power of principal-component analysis of correlated phenotypes in genome-wide association studies. *Am J Hum Genet*, 94(5):662–676, May 2014.

[162] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 8/10/2022 1998.

[163] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[164] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018.

[165] Akiko Nagai, Makoto Hirata, Yoichiro Kamatani, Kaori Muto, Koichi Matsuda, Yutaka Kiyohara, Toshiharu Ninomiya, Akiko Tamakoshi, Zentaro Yamagata, Taisei Mushiroda, Yoshinori Murakami, Koichiro Yuji, Yoichi Furukawa, Hitoshi Zembutsu, Toshihiro Tanaka, Yozo Ohnishi, Yusuke Nakamura, and Michiaki Kubo. Overview of the biobank japan project: Study design and profile. *J Epidemiol*, 27(3S):S2–S8, Mar 2017.

[166] Goddard P Kichaev G Gusev A Pasaniuc B Mancuso N, Shi H. Integrating gene expression with summary association statistics to identify genes associated with 30 complex traits. *Am J Hum Genet*, 100(3):473–487, 2017.

[167] Mallen C Zhang W Doherty M Kuo CF, Grainge MJ. Rising burden of gout in the uk but continuing suboptimal management: a nationwide population study. *Annals of the Rheumatic Diseases*, 74:661–667, 2015.

[168] Richard A. Sturm, David L. Duffy, Zhen Zhen Zhao, Fabio P.N. Leite, Mitchell S. Stark, Nicholas K. Hayward, Nicholas G. Martin, and Grant W. Montgomery. A single snp in an evolutionary conserved region within intron 86 of the herc2 gene determines human blue-brown eye color. *AJHG*, 82(2):424–31, 2008.

[169] Iain Mathieson, Iosif Lazaridis, Nadin Rohland, Swapan Mallick, Nick Patterson, Songül Alpaslan Roodenberg, Eadaoin Harney, Kristin Stewardson, Daniel Fernandes, Mario Novak, et al. Genome-wide patterns of selection in 230 ancient eurasians. *Nature*, 528(7583):499–503, 2015.

[170] Janssens AC Rivadeneira F Lao O van Duijn K Vermeulen M Arp P Jhamai MM van Ijcken WF den Dunnen JT Heath S Zelenika D Despriet DD Klaver CC Vingerling JR de Jong PT Hofman A Aulchenko YS Uitterlinden AG Oostra BA van Duijn CM Kayser M, Liu F. Three genome-wide association studies and a linkage analysis identify herc2 as a human iris color gene. *AJHG*, 82(2):411–23, 2008.

[171] Lin Q. Chen D. et al. Reed, J.A. Ski pathways inducing progression of human melanoma. *Cancer and Metastasis Reviews*, 24(2):265–272, 2005.

[172] W Chen, SS Lam, H Srinath, CA Schiffer, WE Royer, and K Jr Lin. Competition between ski and creb-binding protein for binding to smad proteins in transforming growth factor-β signaling", journal="journal of biological chemistry. 282(15):11365–11376, 2007.

[173] Zhang J Zhang J Li X Xie M, Wu X. Ski regulates smads and taz signaling to suppress lung cancer progression. *Journal of Biological Chemistry*, 56(10):2178–2189, 2017.

[174] P De Camilli, A Thomas, R Cofiell, F Folli, B Lichte, G Piccolo, H M Meinck, M Austoni, G Fassetta, and G Bottazzo. The synaptic vesicle-associated protein amphiphysin is the 128-kd autoantigen of stiff-man syndrome with breast cancer. *Journal of Experimental Medicine*, 178(6):2219–2223, 1993.

[175] Nicolas Duforet-Frebourg, Keurcien Luu, Guillaume Laval, Eric Bazin, and Michael G.B. Blum. Detecting genomic signatures of natural selection with principal component analysis: Application to the 1000 genomes data. *Mol Biol Evol*, 33(4):1082–1093, 2016.

[176] Michael G. B. Blum Keurcien Luu, Eric Bazin. pcadapt: an r package to perform genome scans for selection based on principal component analysis. *Molecular Ecology Resources*, 17(1):67–77, 2017.

[177] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2014.

[178] CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983.

[179] Forrest W Young, Yoshio Takane, and Jan de Leeuw. The principal components of mixed measurement level multivariate data: An alternating least squares method with optimal scaling features. *Psychometrika*, 43(2):279–281, 1978.

[180] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.