Scaling probabilistic models of genetic variation to millions of humans

Prem Gopalan[1], Wei Hao[2], David M. Blei[3*], and John D. Storey[2,4,5*]

[1] Department of Computer Science, Princeton University, Princeton NJ 08544 USA

[2] Lewis-Sigler Institute for Integrative Genomics, Princeton University, Princeton NJ 08544 USA

[3] Departments of Statistics and Computer Science, Columbia University, New York NY 10027 USA

[4] Center for Statistics and Machine Learning, Princeton University, Princeton NJ 08544 USA

[5] Department of Molecular Biology, Princeton University, Princeton NJ 08544 USA

* Address for correspondence: david.blei@columbia.edu or jstorey@princeton.edu

A major goal of population genetics is to quantitatively understand variation of genetic polymorphisms among individuals. Researchers have developed sophisticated statistical methods to capture the complex population structure that underlies observed genotypes in humans. The number of humans that have been densely genotyped across the genome has grown significantly in recent years. In aggregate about 1M individuals have been densely genotyped to date, and if we could analyze this data then we would have a nearly complete picture of human genetic variation. Existing state-of-the-art methods, however, cannot scale to data of this size. To this end, we have developed TeraStructure. TeraStructure is a new algorithm to fit Bayesian models of genetic variation in human populations on tera-sample-sized data sets (10^{12} observed genotypes, e.g., 1M individuals at 1M SNPs). It is a principled approach to approximate Bayesian inference that iterates between subsampling locations of the genome and updating an estimate of the latent population structure. On real and simulated data sets of up to 10K individuals, TeraStructure is twice as fast as existing methods and recovers the latent population structure with equal accuracy. On genomic data simulated at the tera-sample-size scales, TeraStructure continues to be accurate and is the only method that can complete its analysis.

Keywords: admixture, population structure, scalable data analysis, stochastic optimization, variational inference.

Software: TERASTRUCTURE is available for download at https://github.com/premgopalan/terastructure.

Funding: This research was supported in part by NIH grant R01 HG006448 and ONR grant N00014-12-1-0764.

INTRODUCTION

The quantitative characterization of genetic polymorphisms in human populations plays a key role in understanding evolution, migration, and trait variation. Genetic variation of humans is highly structured in that frequencies of genetic polymorphisms depend strongly on ancestry and evolutionary forces that vary among individuals. Therefore, to comprehensively understand human genetic variation, we must also understand the underlying structure of human populations.

Over the last fifteen years, scientists have successfully used genome-wide Bayesian models of genetic polymorphisms to infer the latent structure embedded in an observed population. The probabilistic model of Pritchard, Stephens and Donnelly [1], which we will refer to as the "PSD model", has become a standard tool both for exploring hypotheses about human genetic variation and for taking latent structure into account in downstream analyses. The basic idea behind the PSD model is that each individual's ancestry is composed of a mixture of ancestral populations, and an individual's genotype can thus be modeled as a random process that mixes the frequencies of genetic variants from among these ancestral populations.

The PSD model turns the problem of estimating ancestral population structure into one of posterior inference, i.e., estimating a conditional distribution. The assumed genomic structure—the population proportions for each individual and the allele frequencies for each population—are hidden random variables in the model; the collection of individuals at a collection of SNPs $x = \{x_{i,\ell}\}$ are observed random variables. The main computational problem for the PSD model is to estimate the posterior distribution of the hidden population structure given the data, $p(\beta, \theta \mid x)$. With this posterior, or posterior means of the hidden variables, population geneticists can explore the latent structure of their data and correct for ancestry in downstream analyses. Like many modern Bayesian models, this posterior is not tractable to compute: the original algorithm for using the PSD model [1] and subsequent innovations [2; 3] are all methods for approximating it.

Modern genetics, however, cannot take full advantage of the PSD model and related probabilistic models. The reason is that the existing solutions to the core computational problem—the problem of estimating the latent ancestral structure given a collection of observed genetic data—cannot handle the scale of modern datasets. They require repeatedly iterating through the entire data set to form its approximation. With massive data sets, this is not a practical methodology.

The sample sizes of genome-wide association studies now routinely involve tens of thousands of

people. Public and private initiatives have managed to measure genome-wide genetic variation on hundreds of thousands of individuals. For example, a recent study by the company 23andme used genotypes from 162,721 individuals [4]. Taken together, we now have dense genome-wide genotype data on the order of a million individuals. Fitting probabilistic models on these data would provide an unprecedented characterization of genetic variation and the structure of human populations. But, as we show in our study, this analysis is not possible with the current state of the art.

To this end, we have developed TeraStructure, an algorithm for analyzing data sets of up to 10^{12} genotypes. It is based on "variational inference" [5; 6], a general strategy for Bayesian computation that we can scale to massive data sets [7]. TeraStructure's computational flow iterates between subsampling observed single nucleotide polymorphism (SNP) genotypes, analyzing the subsample, and updating its estimate of the hidden ancestral populations.

RESULTS

TeraStructure provides a statistical estimate of the PSD model, capturing the heterogenous mixtures of ancestral populations that are inherent in a data set of observed human genomes. Formally, the PSD model assumes that there are K ancestral populations, each characterized by its minor allele frequencies β_k for each of the SNPs. Further, it assumes that each individual in the sample exhibits those populations with different proportions θ_i. Finally, it assumes that each SNP genotype ℓ in each individual i, denoted by $x_{i,\ell}$, is drawn from an ancestral population that itself is drawn from the individual-specific proportions. If we code each SNP genotype as a 0, 1, or 2 (to denote the three possible genotypes), then it models $x_{i,\ell} \sim \text{Binomial}(2, p_{i,\ell})$ where $p_{i,\ell} = \sum_k \theta_{i,k}\beta_{k,\ell}$.

TeraStructure has a significantly different computational structure, which is illustrated in Figure 1. At each iteration, it maintains an estimate of the population proportions for each person and the allele frequencies for each population.[1] It repeatedly iterates between the following steps: (a) sample a SNP from the data, $x_{\cdot,\ell}$, the measured genotypes at a single site in the genome across all people, (b) analyze how the current estimates of the ancestral populations explain the genotypes at that SNP, and (c) update the estimates of the latent structure—both the ancestral allele frequencies

[1] We describe and illustrate these quantities as though they are estimates. More technically, the algorithm stores parameterized approximate posteriors to them.

and per-individual population proportions.

It is the subsampling step of the inner loop that allows TeraStrucure to scale to massive genetic data. Rather than scan the entire population at each iteration, it iteratively subsamples a SNP, analyzes the subsample, and updates its estimate. On small data sets, this leads to faster estimates that are as good as those obtained by the slower procedures. More importantly, it lets us scale the PSD model up to sample sizes that are orders of magnitude greater than what the current state of the art can handle. We further emphasize that the technical approach behind TeraStructure—one that repeatedly subsamples from a massive data set and then updates an estimate of its hidden structure—can be adapted to many Bayesian models that are used in modern genetics research, such as HMMs, phylogenetic trees, and others.

TeraStructure is built on variational inference, a method from the statistical machine learning literature that adapts ideas in statistical physics to solve a variety of approximate Bayesian inference problems. The main idea behind variational inference is as follows. We first parameterize individual distributions for each latent variable in the model, i.e., a distribution for each set of per-population allele frequencies $q(\beta_k)$ and a distribution for each individual's population proportions $q(\theta_i)$. We then fit these distributions so that their product is close to the true posterior, where closeness is measured by Kullback-Leibler divergence. (Kullback-Leibler is an information-theoretic quantity that asymmetrically measures the distance between two distributions.) Thus we do Bayesian inference by solving the following optimization problem,

$$q^*(\beta, \theta) = \arg\min_q \mathrm{KL}(\prod_k q(\beta_k) \prod_i q(\theta_i) || p(\beta_1, \ldots, \beta_K, \theta_1, \ldots, \theta_n \mid x)).$$

The key idea in TeraStructure is to solve this optimization problem with *stochastic variational inference* [7], an adaptation of the classical stochastic optimization algorithm [8] to variational inference. Specifically, we optimize the KL divergence by following noisy realizations of its derivatives, where the noise emerges from our subsampling the data at each iteration. The noisy derivatives are much cheaper to compute than the true derivatives, which require iterating over the entire data set. See Methods for the mathematical details that outline the variational objective function and how subsampling the data leads to noisy derivatives.

We applied TeraStructure to both real and simulated data sets to study and demonstrate its good performance. We compared it to ADMIXTURE [2] and FASTSTRUCTURE [3], the two algorithms for estimating the PSD model that work on modestly sized data. In our comparisons, we

timed all the algorithms under equivalent computational conditions. On simulated data, where the truth is known, we measured the quality of the resulting fits by computing the KL divergence between the estimated models and the truth. On the real data sets, where the truth is not known, we measured model fitness by predictive log likelihood of held-out measurements (Methods). The smaller the KL divergence and the larger the predictive likelihood, the better a method performs.

We first analyzed two real data sets: the Human Genome Diversity Panel (HGDP) data set [9; 10] and the 1000 Genomes Project (TGP) [11]. After preprocessing, HGDP consisted of 940 individuals at 642,951 SNPs for a total of 604 million observed genotypes and TGP consisted of 1,718 people at 1,854,622 SNPs for a total of 3.2 billion observed genotypes. In previous work, ADMIXTURE and FASTSTRUCTURE have been shown to perform reasonably well on data sets of this size [2; 3]. In applying all three algorithms to these data, we found that TeraStructure achieved the highest predictive log likelihood of held-out measurements by a modest margin (Table S1) and it also completed its estimation in a comparable period of time (Table 1).

We then studied the algorithms on synthetic data. We designed these data sets be similar to real genetic data sets, but at sizes that push the limits of what is available today (Methods). We simulated data sets consisting of 10,000 individuals, 100,000 individuals, and 1M individuals, each with 1M SNP genotypes per individual. On these data we know the true individual proportions, and we can visualize how well each algorithm reconstructs them (Figure 2). We found that ADMIXTURE and FASTSTRUCTURE were only able to analyze the 10,000-individual set, on which TeraStructure was both 2-3 times faster and more accurate (Tables 1 and S2). More importantly, TeraStructure was the only algorithm that was able to analyze the larger data sets of 100,000 individuals and 1M individuals, and again with high accuracy (Figure 2 and Table S2).

TeraStructure uses a convergence criterion to decide when to stop iterating (Methods). This lets us gauge how many SNPs were necessary to sample before the algorithm had learned the structure of the population. On the HGDP and TGP data, we found that TeraStructure needed to sample $\sim 90\%$ and $\sim 50\%$ of the SNPs, respectively, before converging (Table 1). On the tera-sample-sized data set of 1M individuals by 1M SNPs, TeraStructure sampled $\sim 50\%$ of the SNPs before converging.

When analyzing data with the PSD model, we must choose the number of ancestral populations K. For real data, TeraStructure addressed this model selection problem using a predictive approach [12]. We held out a set of genome SNP locations for each individual and computed the average predictive log likelihood under the model for varying numbers of ancestral populations.

The best choice of K is the one that assigns the highest probability to the held-out set. Our sensitivity analysis revealed that K = 8 had the highest validation likelihood on the TGP data, while K = 10 had the highest likelihood on the HGDP data (Figure S3). On the real data sets, we fixed the number of populations K for each data set to the K with the highest validation likelihood (Table S1); on simulated data sets, we set K to the number of ground truth ancestral populations (Table 1).

DISCUSSION

We have developed TeraStructure, a novel algorithm that repeatedly takes strategic subsamples of genotyping data to uncover the underlying structure of human populations. We have demonstrated the effectiveness of TeraStructure by applying it to large and globally sampled human SNP genotype data and comparing our predictive likelihood to existing algorithms. Further, we used a comprehensive simulation study to show that TeraStructure can accurately fit a standard probabilistic model of population genetic structure on data sets with a million individuals and 10^{12} observed genotypes. This is orders of magnitude beyond the capabilities of current state-of-the-art algorithms. We note that our results are from computation on a modest computing platform. On advanced computing architectures, TeraStructure can analyze even larger data sets, and holds promise of characterizing the structure of world-scale human populations.

Fitting probabilistic models of population structure such as the PSD model is an important part of analyzing genotyping data. Genomic studies are growing and it is vital that our statistical algorithms can scale to millions of individuals and trillions or more genotype observations. Such analyses are not possible with the current state-of-the-art algorithms as they require multiple iterations over the entire data. TeraStructure overcomes this limitation with a different computational structure—one that iterates between subsampling from a population, analyzing the sample, and updating an estimate of hidden structure. Using TeraStructure to analyze tera-sample-size data sets will provide the most comprehensive analyses to date of the global population genetics of humans.

METHODS

Real data sets

We used genotyping data from the Human Genome Diversity Project (HGDP) and 1000 Gnomes Project (TGP), which are the two largest publicly available datasets that sampled individuals globally. To help insure the quality of the data, we filtered the individuals for 95% genotyping completeness and we removed the SNPs with lower than 1% minor allele frequency. The HGDP dataset is the complete Stanford HGDP SNP Genotyping data. We filtered the individuals by removing those not in the "H952" set [13], which leaves us with only the individuals without first or second degree relatives in the data. The final dimensions are 642,951 SNPs by 940 individuals, and a total of 603 million observations (0.08% missing data). The TGP data set was 2012-01-31 Omni Platform Genotypes and is accessible from the NCBI ftp site. We removed related individuals using the sample information provided by the 1000 Genomes Project. The final dimensions are 1,854,622 SNPs by 1,718 individuals, and a total of 3.1 billion observations (0.3% missing data).

Simulated data sets

The goal of our study on synthetic data sets is to demonstrate scalability to tera-sized data sets—one million observed genotypes from one million individuals—while maintaining high accuracy in recovering ground truth per-individual population proportions θ_i and per-population allele frequencies β_k. To this end, we generated synthetic genotype data using the Pritchard-Stephens-Donnelly (PSD) model [1]. A specification of the the per-individual population proportions and the population allele frequencies is our "ground truth". To generate realistic synthetic data, we made the individual θ_i's visually similar to the proportions obtained from fitting our model to the TGP data set. We modeled allele frequencies $\beta_{1:K,l}$ from real data.

In our simulation, the process of drawing an individual i's proportions θ_i has two levels. At the first level, we drew S points in the K-simplex from a symmetric Dirichlet distribution, $q_s \sim \text{Dirichlet}(\alpha)$. Each of the S points represents a "region" of individuals, and each individual was assigned to one of the regions such that the regions are equally sized. Then, we drew the population proportions of each individual, $\theta_i \sim \text{Dirichlet}(\gamma q_{s,1}, \ldots, \gamma q_{s,K})$. Thus, each region has a fixed q_s and the proportion of individuals from that region are governed by the same scaled q_s param-

eter. The parameter q_s controls the sparsity of the θ_i, while the parameter γ controls how similar admixture proportions are within each group. For all simulations, we set $S = 50$, $\alpha = 0.2$, and $\gamma = 50$.

Each $\beta_{1:K,l}$ at a SNP location l, consists of K independent draws from a Beta distribution with parameters following that of the Balding-Nichols Model [14], i.e. $\beta_{k,l} \sim \text{Beta}(\frac{1-F_l}{F_l}p_l, \frac{1-F_l}{F_l}(1-p_l))$ where p_l is the marginal allele frequency and F_l is the Wright's F_{ST} at location l. The paired parameters p_l and F_l were estimated from the HGDP data set described earlier. For each pair, we chose a random complete SNP from the HGDP data and set the allele frequency p_l to the observed frequency. The Wright's F_{ST} F_l was set to the Weir & Cockerham $\mathrm{F_{ST}}$ estimate [15] with 5 discrete subpopulations, following analysis of the HGDP study in [16]. We simulated data with 1,000,000 SNPs and three different scales of individuals: 10,000, 100,000 and 1,000,000. With 1 million individuals and 1 million SNPS, the number of observations is tera-sample-sized, i.e., 10^{12} observations.

The PSD Model

We present the model and algorithm for unphased genotype data, though it easily generalizes to phased data. (Most massive population genetics data sets are unphased.) In unphased data, each observation $x_{i,l} \in \{0, 1, 2\}$ denotes the observed genotype for individual i at SNP location l. The data are coded for how many major alleles are present: $x_{il} = 0$ indicates two minor alleles; $x_{i,l} = 2$ indicates two major alleles; and $x_{i,l} = 1$ indicates one major and one minor allele. In this last case we do not code which allele came from the mother and which from the father. This is what it means for the data to be unphased.

The PSD model captures the heterogenous patterns of ancestral populations that are inherent in observed human genomes. It posits K ancestral populations, each characterized by its allele frequencies across sites, and assumes that each person's genome exhibits these populations with different proportions. Given a set of observed genomes, the goal of the algorithm is to estimate (i) the proportion of each ancestral population present in a given individual, (ii) the ancestral population allele frequencies for each SNP, (iii) the effective allele frequency for each individual/SNP combination. Given observed data, we uncover its population structure by estimating the conditional distribution of the allele frequencies and the per-individual population proportions.

Formally, each population k is characterized by an array of per-location distributions over major and minor alleles $\beta_{k,l} \in (0, 1)$. Each individual i is characterized by its per-population proportions $\theta_{i,k} > 0$, where $\sum_j \theta_{i,j} = 1$. The observation for individual i at location l is assumed drawn from a binomial. Its parameter is a mixture of the population parameters for that location $\beta_{1:K,l}$, where the mixture proportions are defined by the individual θ_i. Thus, across individuals, the basic population distributions are shared at each location but they are exhibited with different individualized proportions.

Placing priors on the hidden variables, the data are assumed drawn from the following model:

$$\begin{aligned} \beta_{k,l} &\sim \text{Beta}(a, b) \\ \theta_i &\sim \text{Dirichlet}(c) \\ x_{i,l} &\sim \text{Binomial}\left(2, \textstyle\sum_k \theta_{i,k}\beta_{k,l}\right). \end{aligned}$$

This is the model for unphased data in [1].

Scalable Computation for the PSD Model

How do we use the PSD model? We are given a set of measured genotypes from N individuals at L locations $x = x_{1:N,1:L}$. Given this data, we compute the posterior distribution of the basic population parameters $\beta = \beta_{1:K,1:L}$ and individual population proportions $\theta = \theta_{1:N,1:K}$. From the posterior we can compute estimates of the latent population structure.

For example, Figure S2 illustrates the posterior expected population proportions, computed from our algorithm, for the 1718 individuals of the 1000-Genomes data set. Figure S2 illustrates these posterior estimates at three values of the latent number of populations K, at $K = 7$, $K = 8$ and $K = 9$. This data set contains over 3 billion observations. Though the model is not aware of the country-of-origin for each individual, our algorithm uncovered population structure consistent with the major geographical regions. Some of the groups of individuals identify a specific region (e.g., red for Africa) while others represent admixture between regions (e.g., green for Europeans and Central/South Americans).

Specifically, we develop a stochastic variational inference algorithm [7] for the PSD model, a strategy whose computational structure is intrinsically efficient. At each iteration, we first subsample a

set of observed genotypes from the data set, a step which involves sampling a location and including the observations for all individuals at that location. We then analyze only those observations at the subsampled location. Finally, we update our estimates of the population-wide hidden structure based on the analysis of the subsample. In each iteration we obtain a new subsample corresponding to a new location and repeat the process.

This is in contrast to previous algorithms for approximate inference in the PSD model, like the MCMC algorithm of [1] or the variational inference algorithm of [3]. These algorithms form an approximate posterior through repeated iterations over the entire data set; such methods are slow for massive data sets. Our method subsamples a SNP location at each iteration, and provides a valid approximation of the admixture posterior that scales to population-size genomic data.

Variational Inference for the PSD Model

The admixture posterior is proportional to the joint distribution

$$p(\theta, \beta \mid x) = \frac{p(\theta)p(\beta)p(x \mid \theta, \beta)}{p(x)}. \tag{1}$$

This distribution is difficult to compute because of the normalizing constant, the marginal probability of the observed genotypes. The central computational problem for the PSD model is how to approximate the posterior.

Variational inference is a class of methods for approximate posterior inference that adapts earlier ideas in statistical physics [17] to probabilistic models [5; 6]. Broadly, we define a family of distributions over the hidden variables $q(\cdot)$ indexed by a set of free parameters ν. We then fit ν to find the member of the family that is close to the posterior, where closeness is measured with KL divergence,

$$\nu^* = \arg\min_{\nu} \mathrm{KL}(q(\beta, \theta \mid \nu) || p(\beta, \theta \mid x)). \tag{2}$$

The objective function of Equation 2 is not computable. (It is not computable for the same reason that exact Bayesian inference is intractable—it requires computing the marginal probability of the data.) Thus variational inference optimizes an alternative objective that is equal to the negative KL up to an unknown additive constant,

$$\mathcal{L}(\nu) = \mathrm{E}_q[\log p(\beta, \theta, x)] - \mathrm{E}_q[\log q(\beta, \theta \mid \nu)]. \tag{3}$$

This objective is a function of the variational parameters ν because each term is an expectation with respect to $q(\cdot)$. Further, though the additive constant is unknown, maximizing Equation 3 with respect to ν is equivalent to minimizing the KL divergence in Equation 2. Intuitively, the first term encourages that $q(\cdot)$ place mass on configurations of the latent variables that best explain the data; the second term, which is the entropy of the variational distribution, encourages that $q(\cdot)$ be diffuse.

To finish specifying the objective, we must set the form of $q(\cdot)$. A key idea behind variational inference is that the form of the variational distribution is set to make the problem tractable, that is, for the objective of Equation 2 to be computable (as well as its gradients). As for most applications of variational inference, we choose $q(\cdot)$ to be the *mean-field family*, the family where each variable is independent and governed by its own parametric distribution,

$$q(\beta, \theta) = \left(\prod_{k=1}^{K} \prod_{l=1}^{L} q(\beta_{k,l} \mid \hat{\beta}_{k,l}) \right) \prod_{i=1}^{N} q(\theta_i \mid \hat{\theta}_i). \tag{4}$$

Our notation is that $\hat{\theta}_i$ is the variational parameter for the ith individual's population proportions θ_i and $\hat{\beta}_{k,l}$ is the variational parameter for the distribution of genotypes in population k at location l. Further, we set the form of each factor to be the same form as the prior. Thus $q(\theta_{i,l} \mid \hat{\theta}_{i,l})$ are Dirichlet distributions and $q(\beta_{k,l} \mid \hat{\beta}_{k,l})$ are Beta distributions. These decisions come from the general theory around mean-field variational inference in exponential families [18; 19]. (See also Equation 7 and Equation 9.)

We emphasize that in the variational family each hidden variable is endowed with its own variational distribution. While the model assumes each individual's proportions come from the same shared prior, the variational family provides a different parameter for each. This gives the variational family the flexibility it needs to represent different individuals with different population proportions. For example, to create Figure S2 we plotted the variational expectation of each individuals population parameters distribution $\mathrm{E}[\theta_i \mid \hat{\theta}_i]$.

With these components—the objective of Equation 3 and the variational family of Equation 4—we have turned the inference problem for the PSD model into an optimization problem.

Stochastic Variational Inference for the PSD Model

Traditional variational inference iterates over all the variational parameters. For example, the authors of [3] approximate the admixture posterior by updating each variational parameter in turn while holding the others fixed. This *batch* strategy is more efficient than MCMC but cannot scale to tera-sized data sets, where the number of individuals N is in the hundreds of thousands or millions and the number of locations L is in the millions.

To solve this problem, we use stochastic optimization [8] applied to the variational objective [20; 7]. Our algorithm follows easy-to-compute noisy estimates of the gradient to more quickly make progress in the variational objective. The noise and computability of the gradient stem from repeated subsampling from the data.

Our algorithm is called TERASTRUCTURE. It maintains a variational estimate of each individuals population proportions $\hat{\theta}_i$ and the allele frequencies of each basic population $\hat{\beta}_k$. It repeatedly cycles through the following steps:

1. Sample an observation from SNP location l from all individuals.
2. Estimate how each individual expresses the basic populations, only using the measured location l. Use these estimates to update the ancestral allele frequencies parameter $\hat{\beta}_{1:K,l}$ local to l.
3. Use these estimates to update the population proportions parameter of all individuals, $\hat{\theta}_{1:N}$.

The stochastic algorithm above can quickly make progress. After one iteration, which involved processing observations at only one of the L possible locations, we have an estimate of the population proportions of all individuals. Given the estimate of the population proportions, an estimate of the ancestral allele frequencies can be computed for any location. In comparison, the batch algorithm of [3] needs to iterate over the entire data set at least once, to make any progress.

Global and local parameters.

Before we develop our algorithm, we use the conditional dependencies in our graphical model to divide our variational parameters into *local* and *global* parameters [7].

In each iteration we subsample genotype measurements for all individuals at a SNP location l. Our sampled observations are $x_{1:N,l}$. Under the PSD model, given individual proportions $\theta_{1:N}$, the sample $x_{1:N,l}$ and the ancestral allele frequencies $\beta_{1:K,l}$ are conditionally independent of all other observations and allele frequencies $\beta_{1:K,-l}$. Thus, the allele frequencies $\beta_{1:K,l}$ are local to the observations $x_{1:N,l}$. The population proportions $\theta_{1:N}$, with the local variables, govern the distribution of observations at any sampled SNP location. Therefore, the $\theta_{1:N}$ are global variables. Following [7], we extend this notion of global and local sets to the variational parameters. Given observations $x_{l,1:N}$ at the location l, the $\hat{\theta}_{1:N}$ are the global variational parameters; the $\hat{\beta}_{1:K,l}$ are the local variational parameters.

In stochastic variational inference [7], we iteratively update local and global parameters. In each iteration, we first subsample a SNP location l and compute optimal local parameters for the sample, given the current settings of the global parameters. We then update the global parameters using a stochastic natural gradient [21] of the variational objective computed from the subsampled data and the local parameters.

We will now develop our algorithm by first obtaining closed form updates for our local and global variational parameters. For the local parameters, we will derive optimal coordinate updates; for the global parameters, we will derive the stochastic natural gradient update.

Computing the optimal local parameters. Given the global parameters $\hat{\theta}_{1:N}$, we can optimize local parameters $\hat{\beta}_{1:K,l}$ in closed form under certain assumptions. These assumptions involve the *complete conditionals* of the hidden variables in the model, and the variational family. A complete conditional is the conditional distribution of a latent variable given the observations and the other latent variables in the model [18]. If the complete conditional of a variable is in the same family as its prior, and the corresponding variational distribution is in the same family, then we can optimize its variational parameter by setting it to the expected natural parameter (under q) of the complete conditional.

If the complete conditional of each latent variable is in the same exponential family as its prior distribution, then the model is *conditionally conjugate*.

The complete conditionals for the $\beta_{k,l}$ at a sampled location l are

$$p(\beta_{k,l}|\beta_{-k,l},\theta,x) \quad \propto \quad p(\beta_{k,l}|a,b)\prod_{i=1}^{N} p(x_{i,l}|\theta_i,\beta_{1:K,l})$$

$$\propto \quad \exp\Big\{(a-1)\log\beta_{k,l} + (b-1)\log(1-\beta_{k,l}) \\ + \sum_n x_{n,l}\log\sum_k \theta_{n,k}\beta_{k,l} + \sum_n (c - x_{n,l})\log(1 - \sum_k \theta_{n,k}\beta_{k,l})\Big\}. \tag{5}$$

The complete conditional in Equation 5 is not in the exponential family because the expectation of the second and third log-of-summation terms, with respect to the variational family q, are intractable. Therefore, the PSD model is not conditionally conjugate.

To overcome the nonconjugacy in the model, we introduce multinomial approximations using the zeroth order delta method for moments [22; 23]. These approximations provide a lower bound to these intractable terms in the variational objective of Equation 3. In particular, we introduce auxiliary K-multinomial distributions $q(\phi_{il})$ and $q(\xi_{il})$,

$$\begin{aligned} \log\left(\textstyle\sum_k \theta_{i,k}\beta_{k,l}\right) &\geq \textstyle\sum_k \phi_{il,k}\log\frac{\theta_{i,k}\beta_{k,l}}{\phi_{il,k}}, \\ \log(1 - \textstyle\sum_k \theta_{i,k}\beta_{k,l}) &\geq \textstyle\sum_k \xi_{il,k}\log\frac{\theta_{i,k}(1-\beta_{k,l})}{\xi_{il,k}}. \end{aligned} \tag{6}$$

These distributions $q(\phi_{il})$ and $q(\xi_{il})$ approximate only the conditionals of the allele frequencies local to the sampled location l and the individual i; the parameters to these distributions are local.

Substituting the lower bounds from Equation 6 in Equation 5, the complete conditional is

$$p(\beta_{k,l}|\beta_{-k,l},\theta,x) \propto \text{Beta}\Big(a + \sum_{i=1}^N y_{i,l}\phi_{il,k}, b + \sum_{i=1}^N (c - y_{i,l})\xi_{il,k}\Big). \tag{7}$$

Our approximation has effectively placed the complete conditional of allele frequency $\beta_{k,l}$ in the exponential family. By choosing the variational distribution $q(\beta_{k,l} \mid \hat{\beta}_{k,l})$ from Equation 4 to be the Beta distribution, the same family as the prior distribution, we satisfy the conditions for a closed form coordinate update for the local parameters $\hat{\beta}_{1:K,l}$. The optimal $\hat{\beta}_{k,l}$ is the expected natural parameter (under q) of the complete conditional in Equation 7 [18].

Another perspective on the approximations in Equation 6 is they lead to a computationally efficient lower bound on the objective of Equation 3.

Computing stochastic gradient updates for the global parameters. We now turn to the stochastic optimization of the population proportions parameter $\hat{\theta}_n$ using the subsampled observations $x_{1:N,l}$ at location l. We compute noisy estimates of the natural gradient [21] of the variational

objective with respect to $\hat{\theta}_n$, and we follow these estimates with a decreasing step-size. Following [7], we can compute the natural gradient of Equation 3 with respect to the global variational parameter $\hat{\theta}_i$ by first computing the coordinate update for $\hat{\theta}_i$ and then subtracting its current setting.

To compute the coordinate update for $\hat{\theta}_i$, we write down the complete conditional of the population proportions θ_i:

$$\begin{aligned} p(\theta_i|\beta, x) &\propto p(\theta_i|c) \prod_{l=1}^{L} \prod_{k=1}^{K} p(\beta_{k,l}|a, b) \prod_{l=1}^{L} p(x_{i,l}|\theta_i, \beta_l) \\ &\propto \exp\Big\{ \sum_k (c-1) \log \theta_{i,k} \\ &\qquad + \sum_l x_{i,l} \log \sum_k \theta_{i,k}\beta_{k,l} + \sum_l (c - x_{i,l}) \log(1 - \sum_k \theta_{i,k}\beta_{k,l}) \Big\}. \end{aligned} \tag{8}$$

Similar to the complete conditionals of the local variables in Equation 5, the complete conditional in Equation 8 is not in the exponential family. We use the multinomial approximations in Equation 6 to bring the complete conditional into the exponential family, and in the same family as the prior distribution over the population proportions:

$$p(\theta_i|\beta, x) \propto \text{Dirichlet}\Big(\alpha_k + \sum_{l=1}^{L} (x_{i,l}\phi_{il,k} + (c - x_{i,l})\xi_{il,k})\Big). \tag{9}$$

Following [7], the stochastic natural gradient of the variational objective with respect to the global parameter $\hat{\theta}_i$, using L replicates of $x_{i,l}$ is

$$\frac{\partial \mathcal{L}(\nu)}{\partial \hat{\theta}_{i,k}} = \alpha + L(x_{i,l}\phi_{il,k} + (c - x_{i,l})\xi_{il,k}) - \hat{\theta}_{i,k}. \tag{10}$$

Notice we have used the expected natural parameter from the complete conditional in Equation 9 in Equation 10. We arrive at this form of the natural gradient by premultipling the gradient by the inverse Fisher information, and replacing the summation over all SNP locations in Equation 10 with a summation over L replications from the sampled location. Equation 10 is a noisy natural gradient of a lower bound on the variational objective of Equation 3.

To optimize the variational objective with respect to the population proportions $\hat{\theta}_i$, we use the natural gradients in Equation 10 in a Robbins-Monro algorithm [7]. At each iteration we update the global variational parameters with a noisy gradient computed from the SNP observations at location l. The step-size at iteration t is ρ_t, and is set using the schedule

$$\rho_t = (t + \tau)^{-\kappa}. \tag{11}$$

This satisfies the Robbins-Monro conditions on the step-size, and guarantees convergence to a local optimum of the variational objective [8].

The stochastic algorithm. The full algorithm is shown in Figure S1. For each iteration, we first subsample a SNP location l and compute optimal local parameters $(\phi_{1:N,l}, \xi_{1:N,l}, \hat{\beta}_{1:K,l})$ for the sample, given the current settings of the global parameters $\hat{\theta}_{1:N}$. We then update the global parameter $\hat{\theta}_i$ of each individual i using the stochastic natural gradient of the variational objective, with respect to $\hat{\theta}_i$, computed from the subsampled data and local parameters.

Memory efficient computation. During training, the stochastic variational inference algorithm is only required to keep the variational population proportions $\hat{\theta}_{i,k}$ for all individuals $i \in 1, \cdots, N$ in memory. For a given location, the optimal local parameters $(\phi_{1:N,l}, \xi_{1:N,l}, \hat{\beta}_{1:K,l})$ can be computed using the local optimization steps—steps 6 to 9—in Figure S1. The local parameters need not be kept around beyond the corresponding sampling step. This drastically cuts the memory needed. The memory requirement is therefore $O(NK)$ where N is the number of individuals and K is the number of latent ancestral populations. Further, this results in a small fitted model state: the fitted $\hat{\theta}_{1:N}$. Given the $\hat{\theta}_{1:N}$, the allele frequencies $\hat{\beta}_{1:K,l}$ can be optimized for any given location l, using the local step.

Linear scaling in the number of threads. We can compute the local steps and the global steps in parallel across T threads. First, we "map" the individuals into T disjoint sets, and each thread is responsible for computation on one of these sets of individuals. Notice that each thread can independently compute the local parameters $(\phi_{n,l}, \xi_{n,l})$ for any individual n that it owns. This corresponds to step 6 of the algorithm in Figure S1. Further, the sums required in step 7 of the algorithm in Figure S1 can also be computed in parallel. The "reduce" step consists of aggregating the per-thread sums in step 7, and estimating the new Beta parameters. This is an $O(T + K)$ operation, where T is the number of threads and K is the number of ancestral populations. Since T and K are small constants, our reduce step is inexpensive. The global step in step 9 can also be computed in parallel.

Given T threads, the computational complexity of the stochastic algorithm is $O(\frac{NK}{T})$. The algorithm is dominated by the parallel computation in steps 6 and 9, which scale linearly in the number

of threads T. By increasing T, we scale our algorithm linearly in the number of threads.

Initializing variational parameters. We initialize the population proportions randomly using $\theta_{ik} \sim \text{Gamma}(100, 0.01)$. Within each local step, we initialize $(\hat{\beta}_{kl,0}, \hat{\beta}_{kl,1})$ at location l to the prior parameters (a, b). We use the same initialization procedure on all data sets.

Assessing convergence using a validation set. We hold out a *validation set* of genotype observations, and evaluate the predictive accuracy on that set to assess convergence of the stochastic algorithm in Figure S1 [12]. These observations are treated as missing during training.

The validation set is chosen with computational efficiency in mind. We will periodically evaluate the heldout log likelihood on this set (the *validation log likelihood*) to determine convergence of the algorithm in Figure S1. By choosing individuals from a small fraction of total locations L, we ensure that this periodic computation is only required to recompute the optimal $\hat{\beta}_{1:K,l}$ for those locations.

The TERASTRUCTURE algorithm stops when the change in validation log likelihood is less than 0.0001%. We measure this change over $100,000$ iterations.

For the validation set, we uniformly sample at random 0.5% of the L locations, and at each location we uniformly sample at random and keep aside observed genotypes for r individuals. The number of per-location held out individuals r is set to $N/100$ for large N ($N > 2000$) and otherwise to $N/10$. This allows for a reasonably small fraction of individuals to be held out from each location. Further, r is limited to a maximum of 1000 individuals for any N.

Choosing the number of ancestral populations. In our experiments on the real data sets (see Table S1), we fixed the number of populations K to the optimal values based on validation log likelihoods. Our sensitivity analysis (see Figure S3) revealed that $K = 8$ had the optimal validation likelihood on the TGP data, while $K = 10$ was the optimal for the HGDP data set. In our experiments on simulated data sets (see Table S2), we set K to the number of ground truth ancestral populations: $K = 6$.

Experimental setup for the study

The goal of our empirical study is to assess the accuracy and scalability of the stochastic variational inference algorithm of Figure S1 and compare to leading scalable methods in the research literature. In this section, we present the details of our experimental setup. We refer the reader to the main article for the results.

We compared our algorithm to the best existing algorithms for discovering population structure: the FASTSTRUCTURE algorithm [3] and the ADMIXTURE algorithm [2]. We fit these algorithms to the largest real-world genotyping data publicly available—the HGDP [9; 10] and the TGP [11] data sets. We also studied fits to massive synthetic data sets. Our synthetic data sets have up to $N = 1,000,000$ individuals and $L = 1,000,000$ SNP locations, for a total of 10^{12} genotype observations.

On the synthetic data sets, we studied the accuracy of these algorithms in retrieving the ground truth population structure, the run time of these algorithms, and the ability of these algorithms to scale to massive data sets. On the real data sets, we used the predictive approach to evaluating model fitness [12].

Metrics. On real data sets, we computed the predictive accuracy on a *test set* of observed genotypes by computing the held-out log likelihood under the PSD model. The test set is chosen to enable a fair comparison to other algorithms. We hold out genotypes for 0.5% of the N individuals from each location $l \in 1, \cdots, L$. A better predictive accuracy corresponds to a better fit to the data [12]. We approximate the predictive distribution of a heldout SNP using variational posterior estimates of θ and β.

On synthetic data sets, we measured the accuracy in recovering the ground truth population proportions. We computed the Kullback Leibler divergence [24] of the variational posterior estimate $\mathrm{E}[\theta_i|\hat{\theta}_i]$ to the true population proportions θ_i^* for each individual i. We then compared the median KL divergence across all individuals.

Hyperparameters. We set the Dirichlet parameter c to $\frac{1}{K}$ to enforce a sparse prior on the per-individual population proportions. We set the learning rate parameters, τ_0 to 1 and κ to 0.5, to

allow rapid learning in the early iterations. Finally, we set the hyperparameters a and b to 1 to enforce a uniform prior on the population parameters $\beta_{1:K,1:L}$. We used the same hyperparameter settings and initialization in all of our experiments.

Open-source software. Our software is implemented in C++ and has 5400 lines of code. It uses the POSIX Threading library for multi-threaded computation. It inputs genotype data in text or PLINK format [25] and outputs the population proportions $\mathrm{E}[\theta_i|\hat{\theta}_i]$. An option to the software tool computes the expected allele frequency Beta parameters local to a list of locations, given the global individual population proportions $\hat{\theta}$ and a list of SNP locations. Our software is available at *http://github.com/premgopalan/terastructure*.

Computing hardware. All experiments were run on a single multicore machine with two Intel Xeon E5-2680v2 processors, with 10 cores each and running at 2.8 GHz. The maximum RAM required for our experiments is 10 GB.

REFERENCES

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

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

[3] Raj, A., Stephens, M., and Pritchard, J. Variational inference of population structure in large SNP datasets. *Genetics* **197**, 573–589 (2014).

[4] Bryc, K., Durand, E., Macpherson, J. M., Reich, D., and Mountain, J. The genetic ancestry of african, latino, and european americans across the united states. *bioRxiv* **http://dx.doi.org/10.1101/009340** (2014).

[5] Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. Introduction to variational methods for graphical models. *Machine Learning* **37**, 183–233 (1999).

[6] Wainwright, M. and Jordan, M. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning* **1**(1–2), 1–305 (2008).

[7] Hoffman, M., Blei, D., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Machine Learning Research* **14**, 1303–1347 (2013).

[8] Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics* **22**(3), 400–407 (1951).

[9] Cann, H. M., de Toma, C., Cazes, L., Legrand, M. F., Morel, V., Piouffre, L., Bodmer, J., Bodmer, W. F., Bonne-Tamir, B., Cambon-Thomsen, A., Chen, Z., Chu, J., Carcassi, C., Contu, L., Du, R., Excoffier, L., Ferrara, G. B., Friedlaender, J. S., Groot, H., Gurwitz, D., Jenkins, T., Herrera, R. J., Huang, X., Kidd, J., Kidd, K. K., Langaney, A., Lin, A. A., Mehdi, S. Q., Parham, P., Piazza, A., Pistillo, M. P., Qian, Y., Shu, Q., Xu, J., Zhu, S., Weber, J. L., Greely, H. T., Feldman, M. W., Thomas, G., Dausset, J., and Cavalli-Sforza, L. L. A human genome diversity cell line panel. *Science* **296**(5566), 261–262, Apr (2002).

[10] Cavalli-Sforza, L. L. The Human Genome Diversity Project: past, present and future. *Nat. Rev. Genet.* **6**(4), 333–340, Apr (2005).

[11] 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. *Nature* **491**(7422), 56–65, Nov (2012).

[12] Geisser, S. and Eddy, W. A predictive approach to model selection. *Journal of the American Statistical Association* **74**, 153–160 (1979).

[13] Rosenberg, N. A. Standardized subsets of the hgdp-ceph human genome diversity cell line panel, accounting for atypical and duplicated samples and pairs of close relatives. *Annals of Human Genetics* **70**(6), 841–847 (2006).

[14] Balding, D. and Nichols, R. A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. *Genetica* **96**(1-2), 3–12 (1995).

[15] Weir, B. S. and Cockerham, C. C. Estimating f-statistics for the analysis of population structure. *Evolution* **38**, 1358–1370 (1984).

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

[17] Peterson, C. and Anderson, J. A mean field theory learning algorithm for neural networks. *Complex Systems* **1**(5), 995–1019 (1987).

[18] Ghahramani, Z. and Beal, M. Propagation algorithms for variational Bayesian learning. In *Neural Information Processing Systems*, 507–513, (2001).

[19] Bishop, C. *Pattern Recognition and Machine Learning*. Springer New York, (2006).

[20] Sato, M. Online model selection based on the variational Bayes. *Neural Computation* **13**(7), 1649–1681 (2001).

[21] Amari, S. Information geometry on hierarchy of probability distributions. *IEEE Transactions on Information Theory* **47**(5), 1701–1711 (2001).

[22] Bickel, P. and Doksum, K. *Mathematical Statistics: Basic Ideas and Selected Topics*, volume 1. Pearson Prentice Hall, Upper Saddle River, NJ, 2nd edition, (2007).

[23] Wang, C. and Blei, D. Variational inference in nonconjugate models. *Journal of Machine Learning Research* **14**, 1005–1031 (2013).

[24] Kullback, S. and Leibler, R. On information and sufficiency. *The Annals of Mathematical Statistics* **22**(1), 79–86 (1951).

[25] Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., Maller, J., Sklar, P., De Bakker, P. I., Daly, M. J., 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).

Figures and Tables

Figure 1: A schematic diagram of stochastic variational inference for the Pritchard-Stephens-Donnelly (PSD) model. The algorithm maintains an estimate of the latent population proportions for each individual. At each iteration it samples SNP measurements from the large database, infers the per-population frequencies for that SNP, and updates its idea of the population proportions. This is much more efficient than algorithms that must iterate across all SNPs at each iteration.

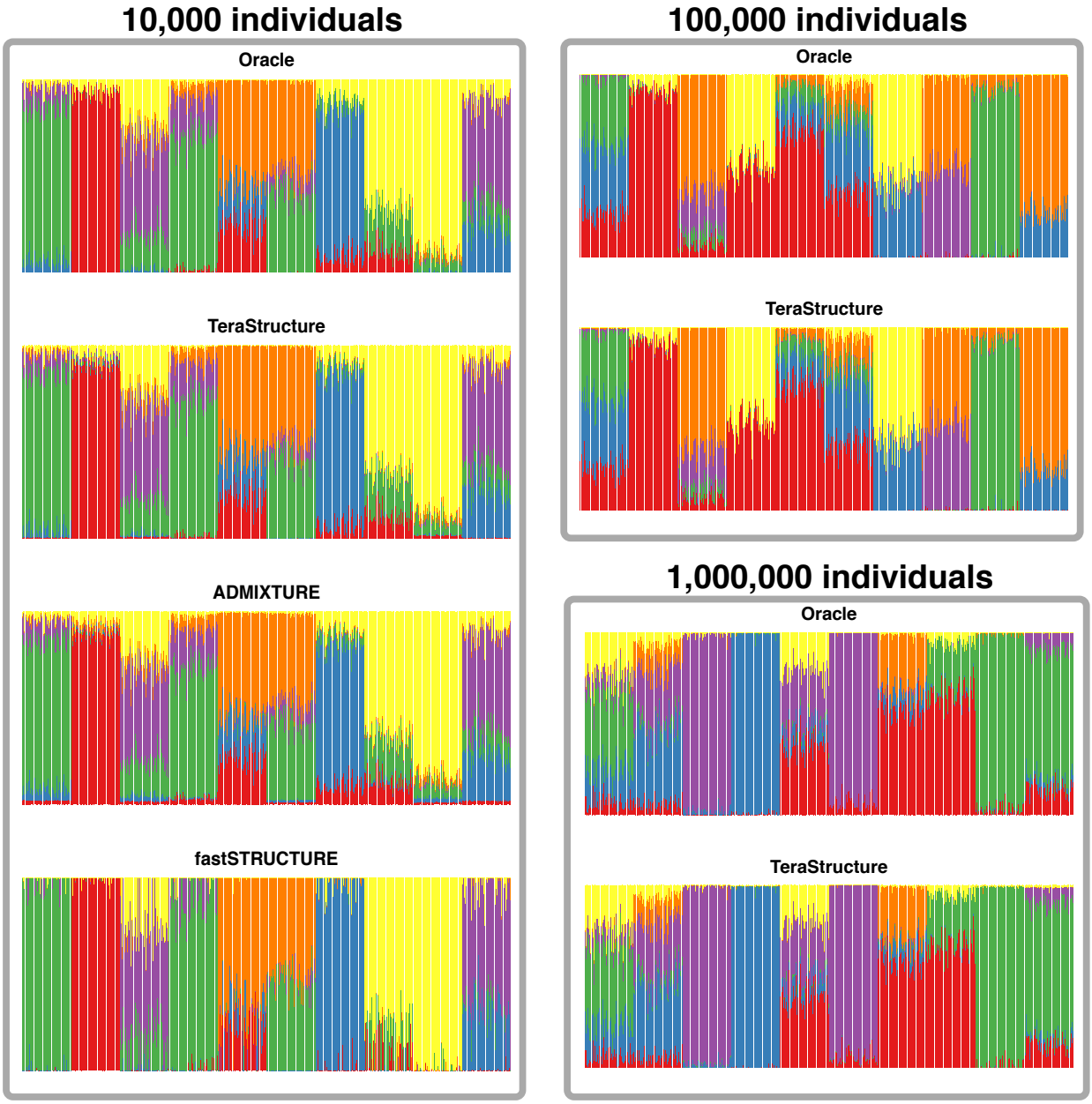

Figure 2: TERASTRUCTURE recovers the ground truth per-individual population proportions on the synthetic data sets with high accuracy. Each panel shows a visualization of the ground truth θ_i^* and the inferred $\mathrm{E}[\theta_i|\hat{\theta}_i]$ for all individuals in a data set. The current state-of-the-art algorithms cannot complete their analyses of 100,000 and 1,000,000 individuals. TeraStructure is able to analyze data of this size and gives highly accurate estimates..

Data set	N	L	S	Time (hours)		
				TERASTRUCTURE	ADMIXTURE	FASTSTRUCTURE
HGDP	940	644,258	0.9	< 1	< 1	12
TGP	1718	1,854,622	0.5	3	3	21
syn10K	10,000	1,000,000	1.0	9	28	216
syn100K	100,000	1,000,000	0.7	158	–	–
syn1M	1,000,000	1,000,000	0.5	509	–	–

Table 1: The running time of all algorithms on both real and synthetic data. TERASTRUCTURE is the only algorithm that can scale beyond $N = 10,000$ individuals to the synthetic data sets with $N = 100,000$ individuals and $N = 1,000,000$ individuals. S is the fraction of SNP locations subsampled, with repetition, during training; L is the number of SNP locations. $S * L$ also equals the number of training iterations of the outer loop in the algorithm of Figure S1 prior to convergence, since we subsample one SNP location in each iteration. The TERASTRUCTURE and ADMIXTURE algorithms were run with ten parallel threads, while FASTSTRUCTURE, which does not have a threading option, was run with a single thread. Even under the best-case assumption of ten times speedup due to parallel computation, the TeraStructure algorithm is twice as fast as both ADMIXTURE and FASTSTRUCTURE algorithms on the data set with $N = 10,000$ individuals. On the real data sets, TERASTRUCTURE is faster than the other algorithms. In contrast to other methods, TERASTRUCTURE iterated over the SNP locations at most once on all data sets.

Supplementary Figures and Tables

1. For all users $i \in 1, \cdots, N$, initialize the population proportions θ_i randomly
2. **repeat**
3. Sample a SNP location l and all observations $x_{1:N,l}$ at that location
4. For $k \in 1, \cdots, K$, initialize $(\hat{\beta}_{kl,0}, \hat{\beta}_{kl,1})$ at SNP location l to (a, b),
5. **Local step: repeat**
6. For $k \in 1, \cdots, K$ and $i \in 1, \cdots, N$ set

$$\begin{aligned} \phi_{il,k} &\propto \exp\Big\{\mathrm{E}[\log \theta_{i,k}] + \mathrm{E}[\log \beta_{k,l}]\Big\} \\ \xi_{il,k} &\propto \exp\Big\{\mathrm{E}[\log \theta_{i,k}] + \mathrm{E}[\log(1-\beta_{k,l})]\Big\} \end{aligned}$$

7. For $k \in 1, \cdots, K$ set the Beta parameters at SNP location l

$$\begin{aligned} \hat{\beta}_{kl,0} &= a + \textstyle\sum_{i=1}^{N} x_{i,l}.\phi_{il,k} \\ \hat{\beta}_{kl,1} &= b + \textstyle\sum_{i=1}^{N} (c - x_{i,l})\xi_{il,k} \end{aligned} \tag{12}$$

8. **until** local parameters $\phi_{1:N,l}$, $\xi_{1:N,l}$ and $\hat{\beta}_{1:K,l}$ converge
9. **Global step:** For $i \in 1, \cdots, N$ set the population proportions

$$\hat{\theta}^{t}_{i,k} = (1-\rho_t)\hat{\theta}^{(t-1)}_{i,k} + \rho_t L(\alpha_k + x_{i,l}\phi_{il,k} + (c - x_{i,l})\xi_{il,k}) \tag{13}$$

10. Set the step-size $\rho_t = (\tau_0 + t)^{-\kappa}$ for iteration t
11. **until** convergence criteria are met

Figure S1: TERASTRUCTURE Algorithm – Stochastic variational inference for the PSD model.

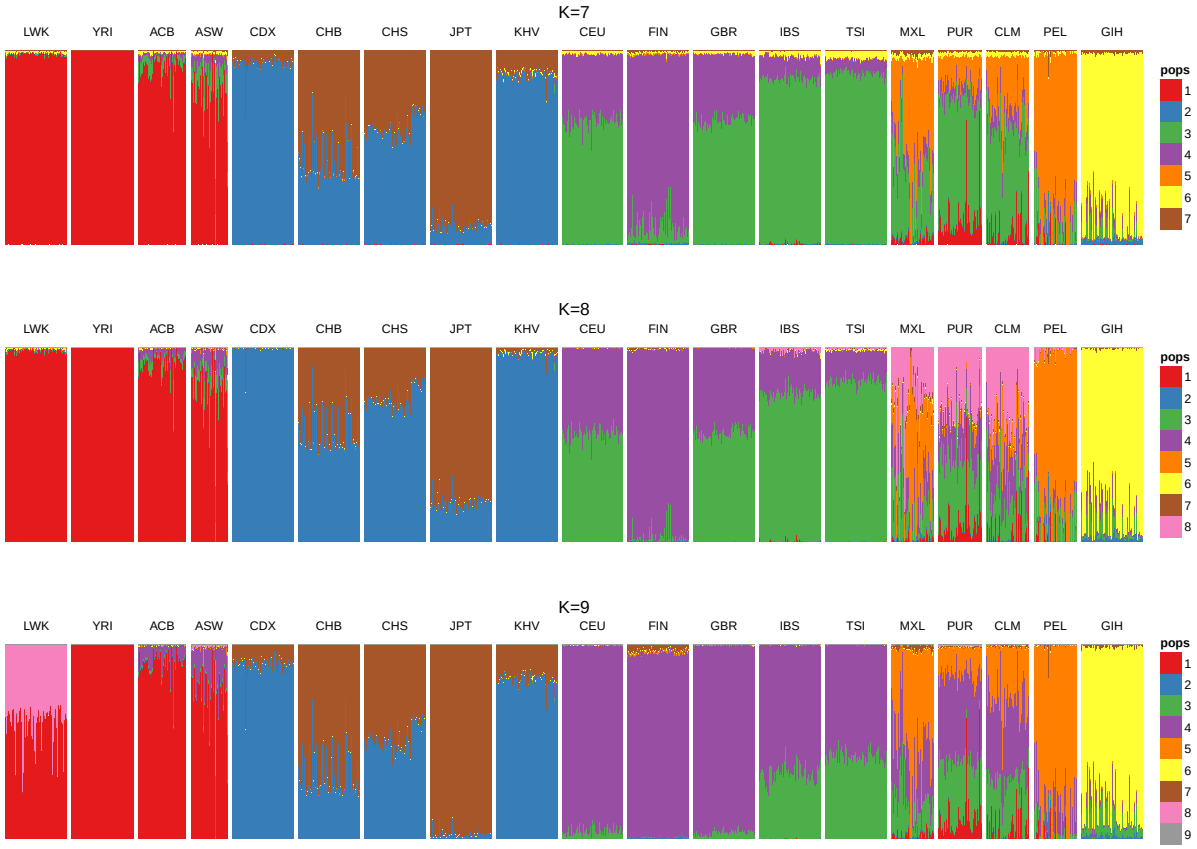

Figure S2: Population structure inferred from the TGP data set using the TERASTRUCTURE algorithm at three settings for the number of populations K. The visualization of the θ's in the Figure shows patterns consistent with the major geographical regions. Some of the clusters identify a specific region (e.g. red for Africa) while others represent admixture between regions (e.g. green for Europeans and Central/South Americans). The presence of clusters that are shared between different regions demonstrates the more continuous nature of the structure. The new cluster from $K = 7$ to $K = 8$ matches structure differentiating between American groups. For $K = 9$, the new cluster is unpopulated.

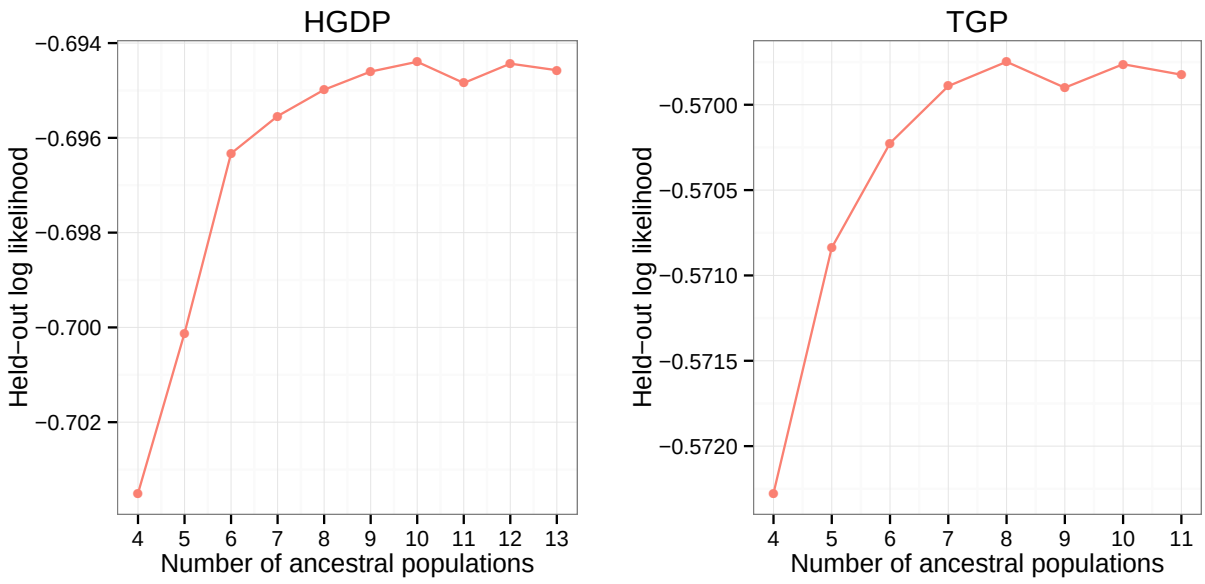

Figure S3: Predictive log likelihood as a function of the number of ancestral populations on the Human Genome Diversity Panel (HGDP) and 1000 Genomes Project (TGP) data sets. The HGDP data peaks at 10 population, and the TGP data peaks at 8 populations.

Data set	N	Mean predictive log likelihood		
		TERASTRUCTURE	ADMIXTURE	FASTSTRUCTURE
HGDP	940	-0.71	-0.71	-0.71
TGP	1718	-0.60	-0.60	-0.61

Table S1: The predictive accuracy of TERASTRUCTURE is comparable to the ADMIXTURE [2] and the FASTSTRUCTURE [3] algorithms, implying a similar model fit. The mean test log likelihood under the model fits is shown. We generated 5 test sets at random and computed the mean over these heldout sets. N is the number of individuals in the data set. The number of ancestral populations is set to $K = 10$ for HGDP and $K = 8$ for TGP.

Data set	Replication	N	L	Median per-individual KL divergence		
				TERASTRUCTURE	ADMIXTURE	FASTSTRUCTURE
syn10K	1	10,000	1,000,000	**0.016**	0.020	6.68
syn10K	2	10,000	1,000,000	**0.009**	0.019	5.15
syn10K	3	10,000	1,000,000	**0.020**	0.022	4.49
syn100K	1	100,000	1,000,000	**0.006**	–	–
syn100K	2	100,000	1,000,000	**0.013**	–	–
syn100K	3	100,000	1,000,000	**0.009**	–	–
syn1M	1	1,000,000	1,000,000	**0.015**	–	–

Table S2: The accuracy of the algorithms on synthetic data. TERASTRUCTURE is the only algorithm that was able to complete its analysis on the synthetic data sets with $N = 100,000$ individuals and $N = 1,000,000$ individuals. On these massive data sets, TERASTRUCTURE found a highly accurate fit to the data (see also Figure 2). On smaller synthetic data, TERASTRUCTURE finds a fit to the data that is closer to the ground truth than either of the other methods. The number of ancestral populations is set to the number of ground truth ancestral populations: K=6.