Supplementary Materials for: “Scaling probabilistic models of genetic variation to millions of humans”

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

[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

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

SUPPLEMENTARY NOTE

Simulation and computing details

The goal of our empirical study is to assess the accuracy and scalability of the stochastic variational inference algorithm of Figure 5 and compare it to leading scalable methods in the research literature. In this section, we present the details for this study. 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 [1] and the ADMIXTURE algorithm [2]. We fit these algorithms to the largest real-world genotyping data publicly available — the HGDP [3; 4], the TGP [5], and the HO [6] data sets. We also studied fits to massive simulated data sets. Our simulated 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 simulated data sets, we studied the accuracy of these algorithms in retrieving the underlying 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 [7].

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 [7]. We approximate the predictive distribution of a heldout SNP using variational posterior estimates of θ and β.

On simulated data sets, we measured the accuracy in recovering the simulation parameter population proportions. We computed the Kullback-Leibler divergence [8] of the variational posterior estimate for θ to the true population proportions θ_i^* for each individual i. We then compared the median KL divergence across all individuals.

Choosing the number of ancestral populations. For our results on the real data sets (see Table 1), we fixed the number of populations K to the optimal values based on validation log likelihoods. Our sensitivity analysis (see Figure 1) revealed that $K = 8$ had the optimal validation likelihood on the TGP data, $K = 10$ was optimal for the HGDP data set, and $K = 14$ was optimal for the HO data set. For our results on simulated data sets (see Table 2), we set K to the number of ancestral populations used in the simulation scenario: $K = 6$ for Scenario A and $K = 10$ for Scenario B.

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 binary PLINK format [9] and outputs the expected population proportions. An option in the software tool computes the expected allele frequency parameters, given the population proportions and a list of SNP locations. Our software is available at *http://github.com/StoreyLab/terastructure*.

Computing hardware. All parts of this empirical study were run on a single multicore machine with two Intel Xeon E5-2680v2 processors, with 10 cores each and running at 2.8 GHz.

Stochastic Variational Inference for the PSD Model

We now derive the details of our algorithm. We first describe the main idea behind variational inference, present the TeraStructure algorithm, and then derive its details.

The variational objective for the PSD model. The goal of inference is to compute the posterior distribution of the per-individual population proportions and the per-population allele frequencies. The posterior is proportional to the joint,

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

It is difficult to compute because of the normalizing constant, which is 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 [10; 11]. A variational inference algorithm approximates the posterior $p(\beta, \theta \mid x)$ with a proxy distribution over the latent variables $q(\beta, \theta \mid \nu)$. This distribution is called the *variational distribution* and it is parameterized by free parameters ν, called *variational parameters*. Variational inference fits these parameters to be close to the exact posterior where closeness is measured with KL divergence. Thus variational inference turns posterior approximation into an optimization,

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

Unlike the posterior, the variational distribution $q(\cdot \mid \nu)$ does not explicitly depend on the data. Its dependence on the data emerges from the optimization; we optimize new variational parameters for each data set that we want to analyze.

Though intuitive, this KL is not computable because it also requires computing the marginal probability of the data. Variational inference optimizes an alternative objective,

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

This objective is equal to the negative KL up to an unknown additive constant; the maximum ν^* of Equation 3 is equal to the minimum of the KL divergence in Equation 2.

Note both terms of Equation 3 are expectations with respect to $q(\beta, \theta \mid \nu)$, so the variational objective is a function of the variational parameters ν. The first term of the variational objective encourages $q(\beta, \theta \mid \nu)$ to place its mass on configurations of the population proportions and allele frequencies that best explain the data; the second term, which is the negative entropy of the variational distribution, encourages it to be diffuse.

We have not yet fully specified the form of the variational family $q(\beta, \theta \mid \nu)$. We must set its factorization and parameterization to fully specify the variational objective. The key idea behind variational inference is to choose a variational family that facilitates optimizing the variational objective. As for most applications of variational inference we will use the *mean-field* distribution, where each variable is independent and governed by its own parametric distribution,

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

The variational parameters are $\nu = \{\hat{\beta}_{1:K,1:L}, \hat{\theta}_{1:N}\}$. The optimization algorithm initializes these parameters and then iteratively fits them to maximize the variational objective.

More precisely, $\hat{\theta}_i$ is the variational parameter for the i-th individual's population proportions θ_i, and $\hat{\beta}_{k,\ell}$ is the variational parameter for the distribution of genotypes in population k at location ℓ. The forms of each factor are the same as their respective prior. Setting up the variational family in this way comes from the general theory around mean-field variational inference in exponential families [12; 13].

The variational factors for the population proportions $q(\theta_i \mid \hat{\theta}_i)$ are Dirichlet distributions. For each individual, the variational distribution of the proportions is

$$q(\theta_i \mid \hat{\theta}_i) = \frac{\Gamma(\sum_k \hat{\theta}_{i,k})}{\prod_k \Gamma(\hat{\theta}_{i,k})} \prod_{k=1}^{K} \theta_{i,k}^{\hat{\theta}_{i,k}-1}. \tag{5}$$

With fitted variational parameters, the approximate posterior expectation of the k population proportion for individual i is

$$\mathrm{E}_{q(\theta)}[\theta_{i,k}] = \frac{\hat{\theta}_{i,k}}{\sum_j \hat{\theta}_{i,j}}. \tag{6}$$

These are the quantities that we plot when visualizing posterior population proportions.

The variational factors for the allele frequencies $q(\beta_{k,\ell} \mid \hat{\beta}_{k,\ell})$ are beta distributions. Recall that the beta has two parameters. For each location ℓ and population k, the variational distribution of the allele frequency is

$$q(\beta_{k,\ell} \,|\, \hat{\beta}_{k,\ell}) = \frac{\Gamma(\hat{\beta}_{k,\ell,0} + \hat{\beta}_{k,\ell,1})}{\Gamma(\hat{\beta}_{k,\ell,0})\Gamma(\hat{\beta}_{k,\ell,1})} \beta_{k,\ell}^{\hat{\beta}_{k,\ell,0}-1}(1 - \beta_{k,\ell})^{\hat{\beta}_{k,\ell,1}-1}. \tag{7}$$

The approximate posterior expectation is

$$\mathrm{E}_{q(\beta)}[\beta_{k,\ell}] = \frac{\hat{\beta}_{k,\ell,0}}{\hat{\beta}_{k,\ell,0} + \hat{\beta}_{k,\ell,1}}. \tag{8}$$

We emphasize the variational family of Equation 4 gives each hidden variable its own 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 makes the variational family flexible. It represents different individuals with different population proportions and different populations with different arrays of allele frequencies.

With the variational objective function in Equation 3 and the variational family of Equation 4, we have turned the inference problem for the PSD model into an optimization problem. Given a data set of genotypes, our goal is to find the individualized variational parameters in a factorized distribution that minimize the KL divergence to the exact posterior. We implement stochastic variational inference in TeraStructure to solve this problem.

Stochastic variational inference for the PSD model. Traditional variational inference optimizes the objective by coordinate ascent. For example, the authors of [1] 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-sample-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.

We scale up the PSD model with stochastic optimization [14] applied to the variational objective [15; 16]. Stochastic optimization follows noisy realizations of the gradient to find the optimum (or local optimum) of an objective function. When the objective contains many terms — the expansion of Equation 3 contains a term for each observed allele — we can obtain easy to compute noisy gradients by repeatedly subsampling the data.

The computational structure of TeraStructure is as follows. The input is a massive set of genomic data. TeraStructure maintains the parameters of Equation 4, which are variational estimates of each individual's population proportions $\hat{\theta}_i$ and the allele frequencies of each ancestral population $\hat{\beta}_k$. It repeatedly cycles through the following steps:

1. Sample a SNP location ℓ, and collect the observations at that location from all individuals.

2. Estimate how each individual is represented by the ancestral populations, based *only* on the observations at location ℓ and the current estimates of the variational parameters.
3. Update the variational parameters. Use the estimates from Step 2 in the following updates:
   - Update the allele frequency parameters at the sampled location $\hat{\beta}_{1:K,\ell}$.
   - Update the population proportion parameters for all individuals, $\hat{\theta}_{1:N}$.

We repeat these steps until convergence, which we discuss below. This algorithm quickly optimizes the variational objective and finds good estimates of the latent population structure even before sampling all the data. We note that this general strategy can be applied to many models in computational biology and statistical genetics. If an inference problem can be turned into an optimization problem, then stochastic optimization can be used to efficiently solve it.

See Figure 5 for the full algorithm.

Updating allele frequencies. After subsampling the individuals at a location ℓ, the first part of the algorithm in Figure 5 estimates the variational parameters for the allele frequencies at that location, $\hat{\beta}_{1:K,\ell}$. Recall these are parameters to posterior beta distributions. We derive updates for these parameters by considering the conditional distribution of each $\beta_{k,\ell}$ given the other latent variables, observations, and prior parameters a and b, also known as the *complete conditional* [15]. Consider the complete conditional for the allele frequency $\beta_{k,\ell}$. It is

$$\begin{aligned} p(\beta_{k,\ell}|\beta_{-k,\ell},\theta,x) &\propto p(\beta_{k,\ell}|a,b)\prod_{i=1}^{N} p(x_{i,\ell}|\theta_i,\beta_{1:K,\ell}) \\ &\propto \exp\Big\{(a-1)\log\beta_{k,\ell} + (b-1)\log(1-\beta_{k,\ell}) \\ &\qquad + \sum_n x_{n,\ell}\log\sum_k \theta_{n,k}\beta_{k,\ell} + \sum_n (2-x_{n,\ell})\log(1-\sum_k \theta_{n,k}\beta_{k,\ell})\Big\}, \end{aligned} \tag{9}$$

where $\beta_{-k,\ell}$ refers to the set of relevant variables except for the k-th. This complete conditional does not permit a closed form update because it is not an exponential family distribution. In order to reach an exponential family distribution, we apply the log sum bound (i.e., Jensen's inequality) to the two summations,

$$\begin{aligned} \log\left(\textstyle\sum_k \theta_{i,k}\beta_{k,\ell}\right) &\geq \textstyle\sum_k \phi_{i,\ell,k}\log\frac{\theta_{i,k}\beta_{k,\ell}}{\phi_{i,\ell,k}}, \\ \log(1-\textstyle\sum_k \theta_{i,k}\beta_{k,\ell}) &\geq \textstyle\sum_k \xi_{i,\ell,k}\log\frac{\theta_{i,k}(1-\beta_{k,\ell})}{\xi_{i,\ell,k}}. \end{aligned} \tag{10}$$

We have introduced auxiliary parameters $\phi_{i,\ell,k}$ and $\xi_{i,\ell,k}$, which are positive and sum to one (over the index k). The use of this approximation in variational inference is described in [17]. Using this bound retains a valid variational algorithm, because it maintains the bound on the KL divergence.

We substitute the bounds in Equation 10 into the conditional of Equation 9. This reveals a beta distribution,

$$p(\beta_{k,\ell}|\beta_{-k,\ell}, \theta, x) \propto \mathrm{Beta}\Big(a + \sum_{i=1}^{N} x_{i,\ell}\phi_{i,\ell,k}, b + \sum_{i=1}^{N}(2 - x_{i,\ell})\xi_{i,\ell,k}\Big). \tag{11}$$

The auxiliary parameters serve to tighten the log-sum bound and improve the Beta approximation to the complete conditional, and we update them by computing and normalizing the following quantities so they sum to one:

$$\begin{aligned} \phi_{i,\ell,k} &\propto \exp\Big\{\mathrm{E}[\log \theta_{i,k}] + \mathrm{E}[\log \beta_{k,\ell}]\Big\} \\ \xi_{i,\ell,k} &\propto \exp\Big\{\mathrm{E}[\log \theta_{i,k}] + \mathrm{E}[\log(1 - \beta_{k,\ell})]\Big\}. \end{aligned} \tag{12}$$

We update the variational parameters by computing the variational expectation of the parameters of the distribution in Equation 11,

$$\begin{aligned} \hat{\beta}_{k,\ell,0} &= a + \sum_{i=1}^{N} x_{i,\ell}\phi_{i,\ell,k} \\ \hat{\beta}_{k,\ell,1} &= b + \sum_{i=1}^{N}(2 - x_{i,\ell})\xi_{i,\ell,k}. \end{aligned} \tag{13}$$

The full procedure consists of iterating between the updates Equation 12 and Equation 13 until convergence.

Updating admixture proportions. In each iteration, once we have fitted the allele frequency parameters, we update the per-individual admixture proportion parameters $\hat{\theta}_i$. These are "global" variables in that their conditional distribution depends on more than the observations at the sampled location. Thus, we update them with a stochastic gradient step.

This step also rests on the complete conditional, which is a Dirichlet distribution. Assume that θ_i has a symmetric Dirichlet distribution as a prior with concentration parameter c. First, the traditional variational update (i.e., using all of the data at all locations) is a Dirichlet,

$$\hat{\theta}_{i,k} = c + \sum_{\ell=1}^{L}(x_{i,\ell}\phi_{i,\ell,k} + (2 - x_{i,\ell})\xi_{i,\ell,k}). \tag{14}$$

This uses the auxiliary variables from the previous step, but requires iterating over all locations ℓ.

In an iteration of TeraStructure, we do not have access to all locations. TeraStructure uses a noisy variational update, where we scale up the contribution from the sampled location to mimic the full data. This gives an intermediate parameter,

$$\tilde{\theta}_{i,k} = c + L(x_{i,\ell}\phi_{i,\ell,k} + (2 - x_{i,\ell})\xi_{i,\ell,k}). \tag{15}$$

The update is a weighted combination of the previous estimate of $\hat{\theta}_i$ and the scaled variational update $\tilde{\theta}$,

$$\hat{\theta}^{t}_{i,k} = (1 - \rho_t)\hat{\theta}^{(t-1)}_{i,k} + \rho_t(c + L(x_{i,\ell}\phi_{i,\ell,k} + (2 - x_{i,\ell})\xi_{i,\ell,k})). \tag{16}$$

This is a valid stochastic gradient step; see [15] for further detail. This is much easier to compute than the update in Equation 14 because it does not iterate over all locations. Finally, we note that the relative weight of the scaled estimate decreases as a function of iteration (ρ_t in step 13 of Figure 5). The learning schedule for ρ_t guarantees that the theoretical conditions of [14] are satisfied, and that we reach a local optimum of the variational objective. By default, τ_0 and κ are both set to 1.

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 $(\phi_{1:N,\ell}, \xi_{1:N,\ell}, \hat{\beta}_{1:K,\ell})$ can be computed using the local optimization steps — steps 5 to 10 — in Figure 5. The local parameters need not be stored 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 $\hat{\beta}_{1:K,\ell}$ 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 $(\phi_{n,\ell}, \xi_{n,\ell})$ for any individual n that it owns. This corresponds to step 8 of the algorithm in Figure 5. Further, the sums required in step 9 of the algorithm in Figure 5 can also be computed in parallel. The "reduce" step consists of aggregating the per-thread sums in step 9, and estimating the new Beta parameters. Our reduce step is inexpensive. The global step in step 12 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 8 and 12, 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 $\theta_{i,k} \sim$

Gamma$(100, 0.01)$. Within each local step, we initialize $(\hat{\beta}_{k,l,0}, \hat{\beta}_{k,l,1})$ 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 genotypes, and evaluate the predictive accuracy on that set to assess convergence of the stochastic algorithm in Figure 5 [7]. The validation set is treated as missing during training.

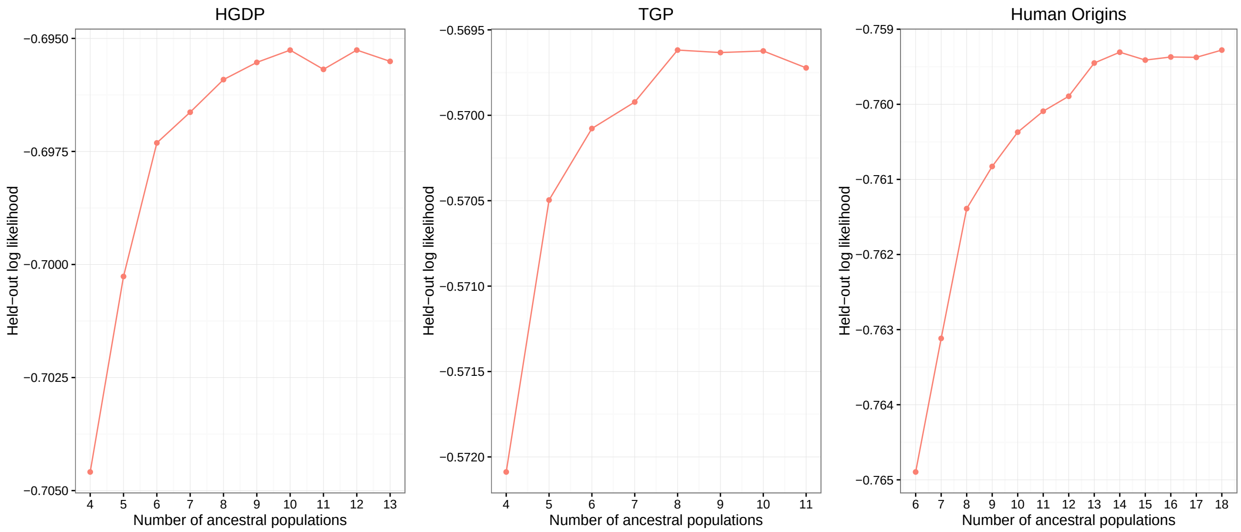
The validation set is chosen with computational efficiency in mind. We will periodically evaluate the held-out log likelihood on this set (the *validation log likelihood*) to determine convergence of the algorithm in Figure 5. By choosing individuals from a small fraction of total locations L, we ensure that this periodic computation is only required to recompute the optimal $\hat{\beta}_{1:K,\ell}$ for those locations.

We kept the convergence criterion fixed across all data sets in this study. 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 data sets much smaller or much larger in scale than what we considered in this study, we expect the number of iterations in this interval to decrease or increase, respectively. This is a tuneable option in the software.

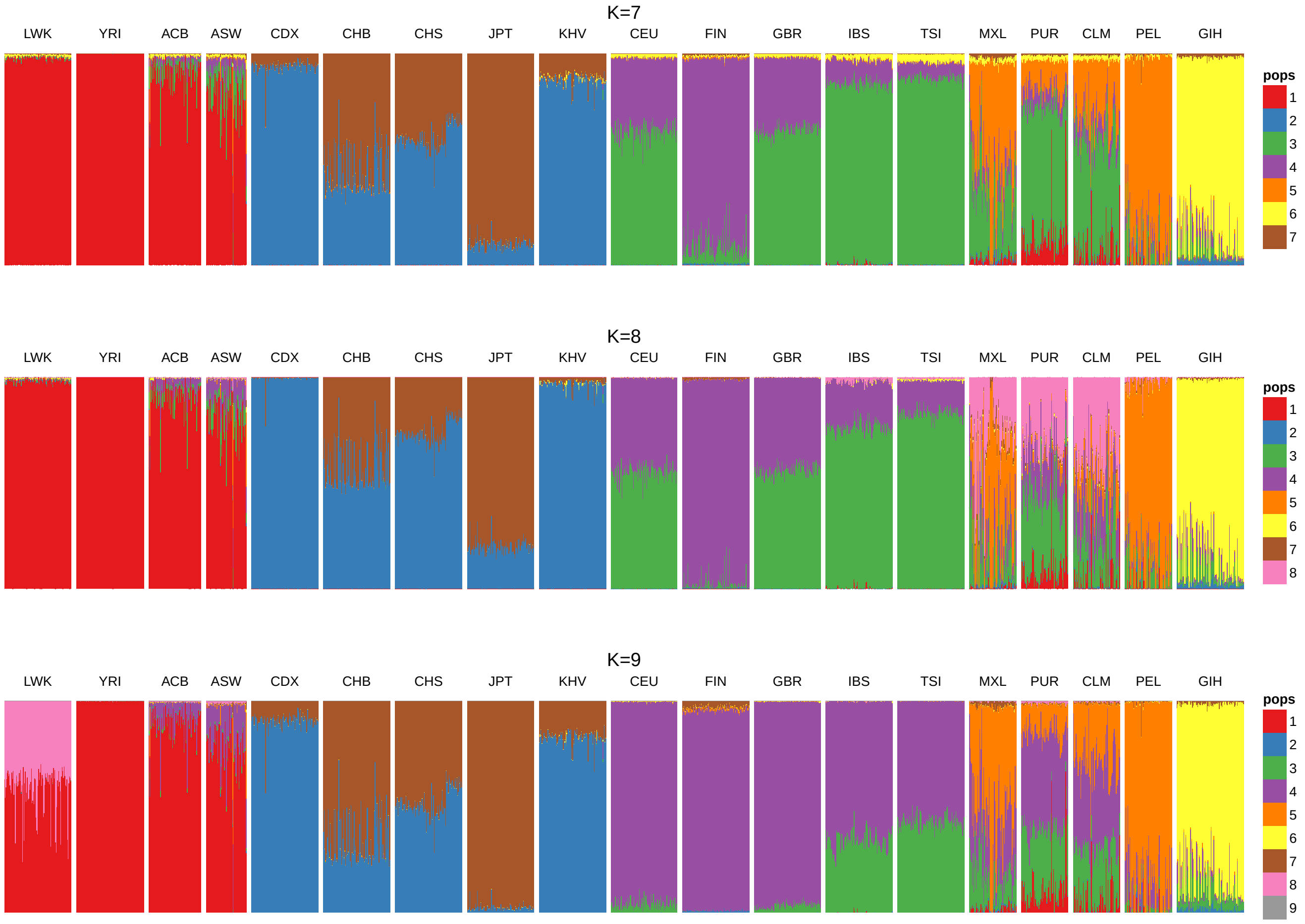
For the validation set, we uniformly sample at random 0.5% of the L locations, and at each location we uniformly sample at random and keep aside observed genotypes for r individuals. The number of per-location held out individuals r is set to $N/100$ for large N $(N > 2000)$ and otherwise to $N/10$. 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.

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 hyperparameters and initialization in all of our analyses.

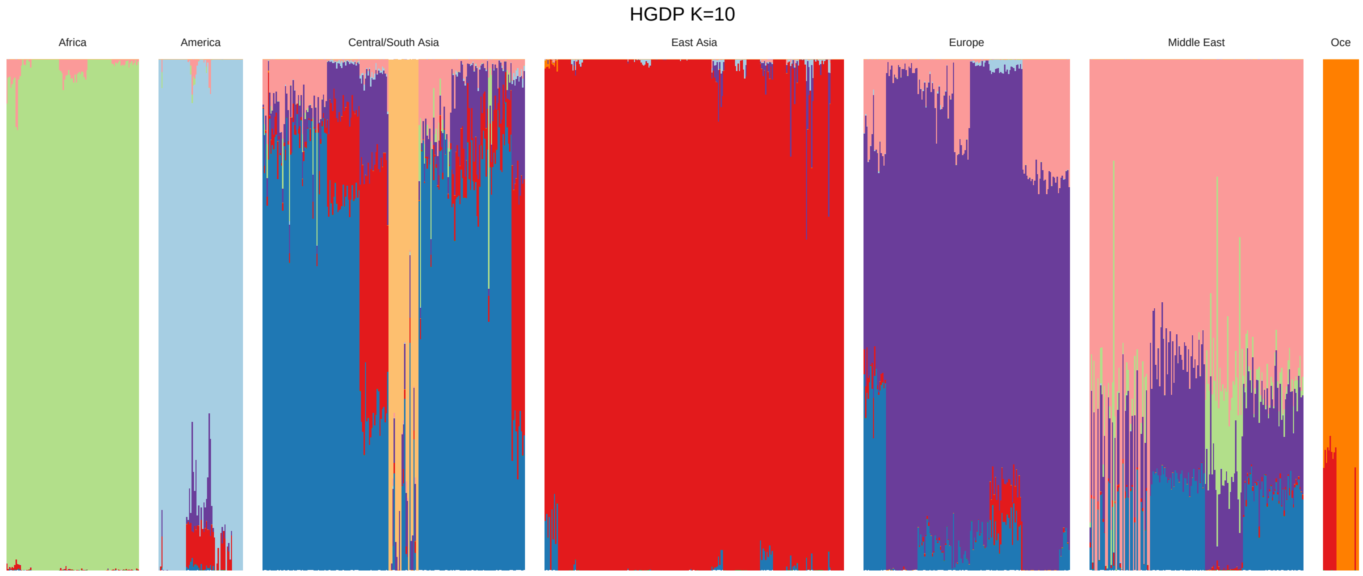
SUPPLEMENTARY FIGURES AND TABLES



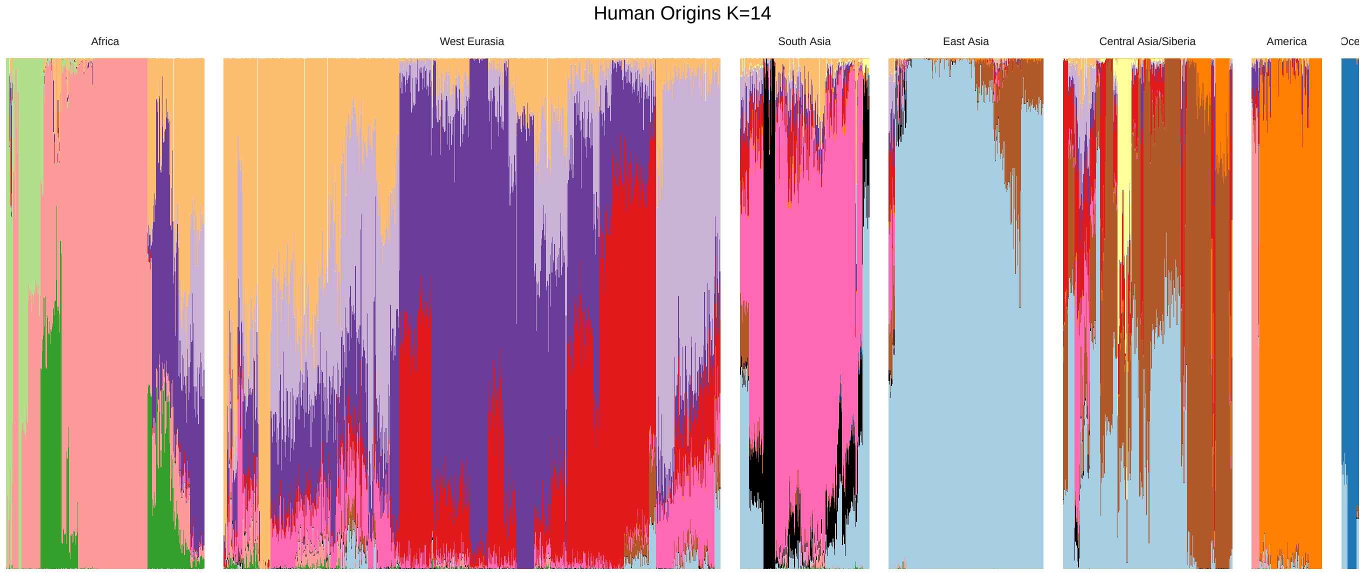
Supplementary Figure 1: Predictive log likelihood as a function of the number of ancestral populations on the Human Genome Diversity Panel (HGDP), 1000 Genomes Project (TGP), and Human Origins (HO) data sets. The HGDP data peaks at 10 populations, the TGP data peaks at 8 populations, and the HO data peaks at 14 populations.



Supplementary Figure 2: Population structure inferred from the TGP data set using the TeraStructure algorithm at three values 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.



Supplementary Figure 3: Population structure inferred from the HGDP data set using the TeraStructure algorithm for the value of K with the best predictive likelihood.



Supplementary Figure 4: Population structure inferred from the HO data set using the TeraStructure algorithm for the value of K with the best predictive likelihood.

```
Data: Observed genotype for $N$ individuals measured at $L$ locations
For all users $i \in 1, \cdots, N$, initialize the population proportions $\hat{\theta}_i$ randomly. Assume $K$ ancestral populations.
Repeat
    Sample a SNP location $l$ and all observations $x_{1:N,\ell}$ at that location.
    For $k \in 1, \cdots, K$, initialize $(\hat{\beta}_{k,\ell,0}, \hat{\beta}_{k,\ell,1})$ at SNP location $\ell$ to $(a, b)$,
    (Allele frequency parameters)
    Repeat:
        For $k \in 1, \cdots, K$ and $i \in 1, \cdots, N$ set
            $\phi_{i,\ell,k} \propto \exp\left\{\mathrm{E}[\log \theta_{i,k}] + \mathrm{E}[\log \beta_{k,\ell}]\right\}$
            $\xi_{i,\ell,k} \propto \exp\left\{\mathrm{E}[\log \theta_{i,k}] + \mathrm{E}[\log(1 - \beta_{k,\ell})]\right\}$
        For $k \in 1, \cdots, K$ set the Beta parameters at SNP location $l$
            $\hat{\beta}_{k,\ell,0} = a + \sum_{i=1}^{N} x_{i,\ell}\phi_{i,\ell,k}$
            $\hat{\beta}_{k,\ell,1} = b + \sum_{i=1}^{N} (2 - x_{i,\ell})\xi_{i,\ell,k}$
    until local parameters $\phi_{1:N,\ell}$, $\xi_{1:N,\ell}$ and $\hat{\beta}_{1:K,\ell}$ converge
    (Population proportions parameters)
    For $i \in \{1, \cdots, N\}$, $k \in \{1, \cdots, K\}$
        $\hat{\theta}^{t}_{i,k} = (1 - \rho_t)\hat{\theta}^{(t-1)}_{i,k} + \rho_t L(c + x_{i,\ell}\phi_{i,\ell,k} + (2 - x_{i,\ell})\xi_{i,\ell,k})$
    Set the step-size $\rho_t = (\tau_0 + t)^{-\kappa}$ for iteration $t$
until convergence criteria are met
```

Supplementary Figure 5: TeraStructure Algorithm – Stochastic variational inference for the PSD model. We use a minibatch size of one, i.e., each iteration subsamples one location.

Data set	N	Mean predictive log likelihood		
		TeraStructure	ADMIXTURE	fastSTRUCTURE
HGDP	940	-0.71	-0.71	-0.71
TGP	1,718	-0.60	-0.60	-0.61
HO	1,941	-1.15	-1.15	-1.21

Supplementary Table 1: The predictive accuracy of TeraStructure is comparable to the ADMIXTURE [2] and the fastSTRUCTURE [1] 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, $K = 8$ for TGP, and $K = 14$ for HO.

Data set	Replication	N	L	Median per-individual KL divergence		
				TeraStructure	ADMIXTURE	fastSTRUCTURE
Scenario A (10K)	1	10,000	1,000,000	**0.016**	0.020	6.68
Scenario A (10K)	2	10,000	1,000,000	**0.009**	0.019	5.15
Scenario A (10K)	3	10,000	1,000,000	**0.020**	0.022	4.49
Scenario A (100K)	1	100,000	1,000,000	**0.006**	–	–
Scenario A (100K)	2	100,000	1,000,000	**0.013**	–	–
Scenario A (100K)	3	100,000	1,000,000	**0.009**	–	–
Scenario A (1M)	1	1,000,000	1,000,000	**0.015**	–	–
Scenario B	1	10,000	100,000	**0.21**	0.42	7.21
Scenario B	2	10,000	100,000	**0.27**	0.42	7.97
Scenario B	3	10,000	100,000	**0.26**	0.42	7.68
Scenario B	1	10,000	1,000,000	**0.16**	–	–
Scenario B	2	10,000	1,000,000	**0.23**	–	–
Scenario B	3	10,000	1,000,000	**0.25**	–	–

Supplementary Table 2: The accuracy of the algorithms on simulated data generated via Scenario A. TeraStructure is the only algorithm that was able to complete its analysis on the simulated 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 simulated data, TeraStructure finds a fit to the data that is closer to the simulation model than either of the other methods. The number of ancestral populations is set to the number of ancestral populations used in the simulation: K=6.

REFERENCES

[1] Raj, A., Stephens, M., and Pritchard, J. K. fastSTRUCTURE: variational inference of population structure in large SNP data sets. *Genetics* **197**(2), 573–589, Jun (2014).

[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] 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).

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

[5] 1000 Genomes Project Consortium. An integrated map of genetic variation from 1,092 human genomes. *Nature* **491**(7422), 56–65, Nov (2012).

[6] Lazaridis, I., Patterson, N., Mittnik, A., Renaud, G., Mallick, S., Kirsanow, K., Sudmant, P. H., Schraiber, J. G., Castellano, S., Lipson, M., Berger, B., Economou, C., Bollongino, R., Fu, Q., Bos, K. I., Nordenfelt, S., Li, H., de Filippo, C., Prufer, K., Sawyer, S., Posth, C., Haak, W., Hallgren, F., Fornander, E., Rohland, N., Delsate, D., Francken, M., Guinet, J. M., Wahl, J., Ayodo, G., Babiker, H. A., Bailliet, G., Balanovska, E., Balanovsky, O., Barrantes, R., Bedoya, G., Ben-Ami, H., Bene, J., Berrada, F., Bravi, C. M., Brisighelli, F., Busby, G. B., Cali, F., Churnosov, M., Cole, D. E., Corach, D., Damba, L., van Driem, G., Dryomov, S., Dugoujon, J. M., Fedorova, S. A., Gallego Romero, I., Gubina, M., Hammer, M., Henn, B. M., Hervig, T., Hodoglugil, U., Jha, A. R., Karachanak-Yankova, S., Khusainova, R., Khusnutdinova, E., Kittles, R., Kivisild, T., Klitz, W., Ku?inskas, V., Kushniarevich, A., Laredj, L., Litvinov, S., Loukidis, T., Mahley, R. W., Melegh, B., Metspalu, E., Molina, J., Mountain, J., Nakkalajarvi, K., Nesheva, D., Nyambo, T., Osipova, L., Parik, J., Platonov, F., Posukh, O., Romano, V., Rothhammer, F., Rudan, I., Ruizbakiev, R., Sahakyan, H., Sajantila, A., Salas, A., Starikovskaya, E. B., Tarekegn, A., Toncheva, D., Turdikulova, S., Uktveryte, I., Utevska, O., Vasquez, R., Villena, M., Voevoda, M., Winkler, C. A., Yepiskoposyan, L., Zalloua, P., Zemunik, T., Cooper, A., Capelli, C., Thomas, M. G., Ruiz-Linares, A., Tishkoff, S. A., Singh, L., Thangaraj, K.,

Villems, R., Comas, D., Sukernik, R., Metspalu, M., Meyer, M., Eichler, E. E., Burger, J., Slatkin, M., Paabo, S., Kelso, J., Reich, D., and Krause, J. Ancient human genomes suggest three ancestral populations for present-day Europeans. *Nature* **513**(7518), 409–413, Sep (2014).

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

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

[9] 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).

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

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

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

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

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

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

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

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