UNIVERSITY OF MICHIGAN
COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
Department of Statistics

STATS 701 FINAL PROJECT

Exploiting the population structure: STRUCTURE and VLAD algorithm

Student: Trong Dat Do
ID Number: 31544397
Advisor: Prof. Jonathan Terhorst

Ann Arbor, Michigan - 2020

PREFACE

Exploiting the population structure is an important task in genetics. It can help to understand ancestor populations of the data and assign each individual as an admixture proportions of those populations. It has several applications such as identifying migration and "criptic" population structure [7]. In this project, we together review some perspectives of learning population structure in the literature in the last 20 years, where we start with the most popular algorithm and some of its variants. Each variant can overcome a weakness of the original algorithm and make it better and faster through time. Finally, we introduce a potentially very fast algorithm that can be applied in the context that can give similar result to the original model.

This report is written as the final project of STATS701 offered by Department of Statistics, University of Michigan, Ann Arbor. The author wants to thank to Professor Jonathan Terhorst for his valuable lessons in the last semester and his insightful suggestions for the project. Thank to all members of the class for interesting lessons that they have shared.

4 December 2020,

Trong Dat Do

Contents

1 Introduction

Having more than 30,000 citations in the last 20 years, it can be said that the STRUCTURE algorithm [7] is the most popular algorithm for biologists to investigate the population structure based on genotype data. The algorithm only requires a table of multilocus data (SNP or microsatellite, ...) to be input and can cluster samples into different populations (admixture). Because of its simplicity and the ability to produce interesting insights about biological data sets, it is widely applied in many studies of different organisms (human [10], flowers [5], chickens [9]). However, the algorithm has some weaknesses such as can not fully capture the Linkage Disequilibrium of the genotype data, need to initially choose the number of population and has a large running time because of using the Markov Chain Monte Carlo (MCMC) method. In this project, we will spend the first half to study about STRUCTURE algorithm and some of its variants to overcome those weaknesses. One is by using a hidden Markov Chain to model the Linkage Disequilibrium of the loci ([4]). The next model uses Hierarchical Dirichlet Process to model the priors therefore can be more flexible and does not require one to initially choose the number of population K ([3]). The last one using Variational Inference to approximate the posterior distribution of the model ([8]). Despite of sacrificing the exactness of the posterior and putting a very strong assumption of independence in the latent variables, it greatly speeds up the algorithm.

For the second half of the report, we will together see a geometry-based algorithm to solve the same problem. It is called the VLAD algorithm [13]. It is a nice touches from modern high-dimension statistics. Usually, our problem will have large L (number of loci), small N (number of sample), and even smaller K (number of population), we can hope to exploit some low-dimension structures in this high-dimension setting. The VLAD algorithm uses the same likelihood as STRUCTURE and tries to use geometry properties to cluster populations. It has been proved to have a consistency property and fast running time. Finally, we apply all algorithms to some data sets including simulated and real data, and comment about their performance.

The rest of this report is organized as follows. In Section 2, we will review the STRUCTURE algorithms, the original version, and two variants using HDP and VI. We talk about VLAD algorithm in Section 3. The applications of all algorithms will be presented in Section 4, to simulated data sets in different setting. We will give a conclusion in Section 5.

2 STRUCTURE algorithm and variants

Suppose that our data set is a table $(X_{il})_{i=1,\dots,N,l=1,\dots,L}$ of haploid samples, where N is the number of sample, L is the number of loci (usually L is much larger than N), and (X_{il}) is the genotype of sample n at the locus l. In this report, we only focus on single nucleotide polymorphisms (SNPs) so each X_{il} will only take value 0 or 1. (But all the

algorithm can be extended to microsatellite setting and for diploid data easily.)

The main purpose of STRUCTURE algorithm is to cluster samples into populations in an admixture way. To be more specific, suppose that we have K ancestor populations. We assume there exists a Hardy-Weinberg equilibrium (HWE) in each population and let θ_{kl} to be the frequency of nucleus a at locus l to take value 1 in the population k, where $l = \overline{1,L}, a = 1,2, k = \overline{1,K}$. For each individual n, we denote Q_{ik} to be the proportion that he inherits from the population k, so $\sum_{k=1}^{K} Q_{ik} = 1$. Our final goal is to learn $\theta = (\theta_{kl})$ (HWE probability) and $Q = (Q_{ik})$ (admixture probability of individuals) from the data $X = (X_{il})_{i=1,\dots,N=1,\dots,L}$.

Notation and abbreviation: We write i, l, k for the running indices of $1, \dots, N$; $1, \dots, L$; $1, \dots, K$, respectively, and sometimes omit N, L, K. We use Dir for Dirichlet distribution; DP for Dirichlet Process; Ber for Bernoulli distribution; Discrete(α), where α is a vector on simplex of $k-1$ dimensions, for Discrete distribution on the set $\{1, \dots, k\}$.

2.1 The original STRUCTURE algorithm

In the admixture model of the original algorithm's paper [7], the authors introduce latent variable Z_{il}^{a} to be the ancestor population of X_{il}, that is

$$X_{il}|(Z_{il} = k), (\theta_{kl}) \sim \text{Ber}(\theta_{kl}), \tag{1}$$

and the latent variables Z_{il} is distributed based on Q_i

$$Z_{il}|Q_i \overset{iid}{\sim} \text{Discrete}(Q_i). \tag{2}$$

The graphical representation of the model can be seen in Figure 2.1.

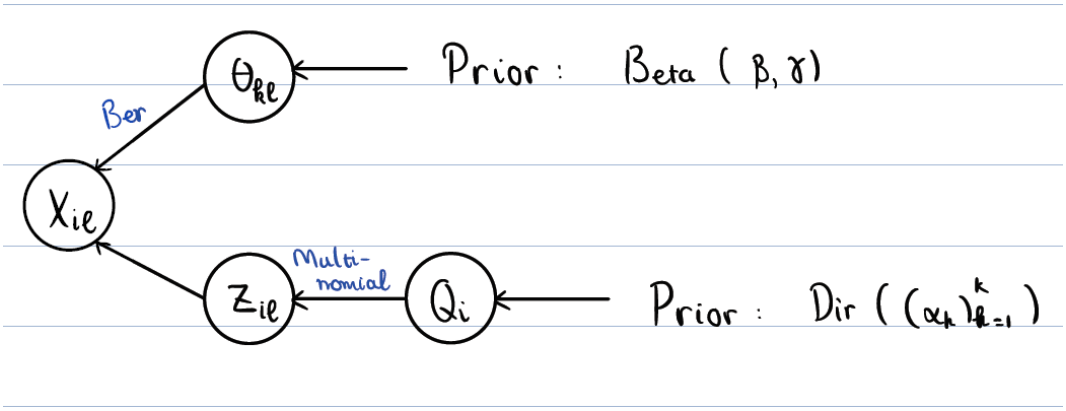

Figure 1: The graphical representation of STRUCTURE

After specifying the likelihood of the model. We put a prior on $(\theta_{l,k})$ and (Q_i) to complete it.

$$\theta_{kl} \overset{iid}{\sim} \text{Beta}(\beta, \gamma), \quad Q_i \overset{iid}{\sim} \text{Dir}((\alpha_k)_{k=1}^K). \tag{3}$$

Because $\theta_{l,k}$, frequency of nucleus a at locus l to take value 1 in the population k, takes value $\in [0,1]$, it is natural to put a $\text{Beta}(\beta, \gamma)$ prior on it. For each $Q_i = (Q_{ik})_{k=1}^K$, because its element sum up to 1, and the likelihood of $Z_{i,l}$ based on Q_i is counting, so putting a Dirichlet prior $\text{Dir}((\alpha_k)_{k=1}^K)$ on it is reasonable. The specified priors also enjoy the conjugacy property given each other, which makes the posterior inference feasible by using MCMC. The details of MCMC updating rule can be seen in Appendix A.

2.2 Modelling the Linkage Disequilibrium

In the algorithm above, the ancestor populations of loci $(Z_{i,l}^{(a)})_{l=1}^L$ are assumed to be i.i.d. given the mixing probability Q_i. This is biologically incorrect because of the recombination. The DNA information from parents pass to their children chunks by chunks. Therefore, there are some neighboring loci are more likely to belong to the same ancestor population than others. Falush et. al [4] suggest to model $Z_{i,l}^a$ as a Markov Chain

$$P(Z_{i,(l+1)}^a = k | Z_{i(l+1)}^a = k', r, q_i) = \begin{cases} \exp(-d_l r) + (1 - \exp(-d_l r)) q_{ik'}, & \text{if } k' = k, \\ (1 - \exp(-d_l r)) q_{ik}, & \text{otherwise,} \end{cases} \tag{4}$$

and for the initial locus

$$P(Z_{i1}^a = k | q_i) = q_{ik}, \tag{5}$$

where r is called the recombination rate and d_l is the genetics distance between the locus $l+1$ and l. We can see the intuition more clearly if we define a latent variable $(S_{i,l}^a)_{l=1\overline{1,L-1}}$, which can be described as linkage indicators, such that

$$S_{i,l}^a \overset{iid}{\sim} \text{Ber}(\exp(-r d_l)), \tag{6}$$

$$Z_{i,l+1}^a | S_{i,l+1}^a, Z_{i,l}^a, Q_i \begin{cases} = Z_{i,l}^a, & \text{if } S_{i,l+1}^a = 1, \\ \sim \text{Discrete}(Q_i), & \text{otherwise.} \end{cases} \tag{7}$$

The MCMC updating scheme can be written using the Forward Filtering Backward Sampling method, we also can update r by using Metropolis-Hasting sampling to find an appropriate recombination rate.

2.3 STRUCTURE using Hierachical Dirichlet Process (HDP)

Our of the problem when applying STRUCTURE as well as any other parametric clustering methods is to choose the number of clusters (K) in the beginning. For La-

tent Dirichlet Allocation (siblings algorithm of STRUCTURE using to cluster topics of documents), Teh et al. [12] introduce the Hierarchical Dirichlet Process to solve this issue. In 2019, [3] also propose a similar method for the STRUCTURE model. We will summary that method in this section. The graphical representation of the model is as follows.

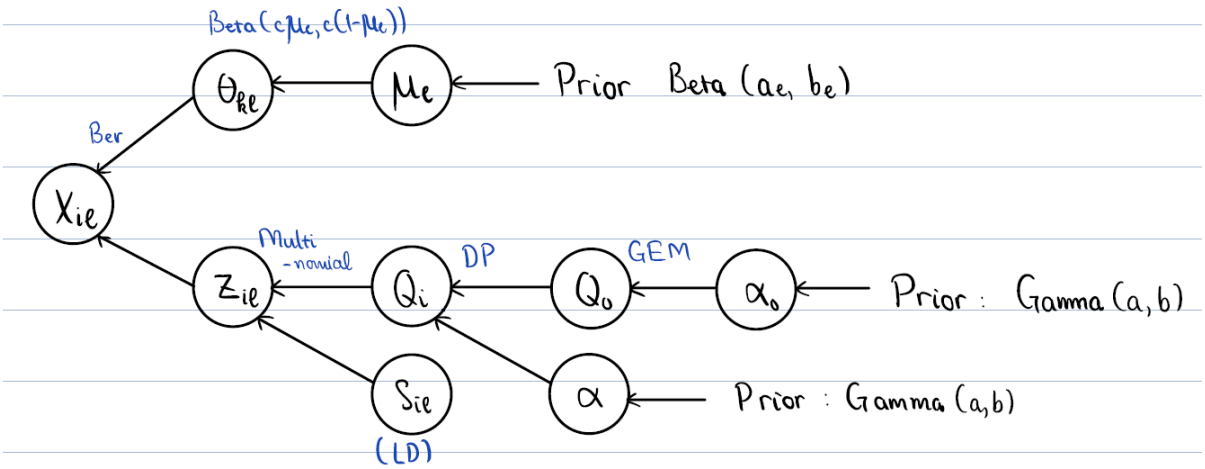

Figure 2: The graphical representation of HDPStructure

The main difference here is using Dirichlet Processes instead of Dirichlet distribution to model $(Q_i)_{i=1}^N$. To be more specific, they put the prior for mixing parameters as follows.

$$Q_i|Q_0 \sim \text{DP}(\alpha Q_0), \tag{8}$$

and $Q_0 = (q_{0k})_{k=1}^\infty$ is specified by the stick-breaking representation for Dirichlet Process ([11]) as follows.

$$v_{0k} \overset{iid}{\sim} \text{Beta}(1, \alpha_0), \qquad q_{0k} = v_{0k} \prod_{k'=1}^{k-1} (1 - v_{0k'}), \quad \forall\, k = 1, 2, \ldots \tag{9}$$

where α_0 is called concentration parameter. Larger α_0 corresponds to a larger number of populations with more uniform proportion. This distribution is also called GEM (Griffiths-Engen-McCloskey) in the Bayesian non-parametric literature. Given this representation, we also have the stick-breaking representation for $(Q_i) = (q_{ik})_{k=1}^\infty$.

$$v_{ik} \overset{iid}{\sim} \text{Beta}(\alpha v_{0k}, \alpha(1 - \sum_{k'=0}^{k} v_{0k'})), \quad q_{ik} = v_{ik} \prod_{k'=1}^{k-1} (1 - v_{ik'}), \quad \forall\, k = 1, 2, \ldots \tag{10}$$

Because there are infinity weights in the stick-breaking presentation, we can introduce a slice variable $C_i \sim \text{Uniform}([0, q_i^{min}])$, where $q_i^{min} = \min_l q_{iz_{i,l}}$ and cut down all the weight $< C_i$. After that, we can perform a Metropolis-Hasting step to make sure that the posterior is correct.

2.4 fastSTRUCTURE using Variational Inference (VI)

Because of using MCMC, STRUCTURE itself and two of its variant above have large running time. In 2014, [8] introduced Variational Inference for STRUCTURE. Instead of using MCMC to infer the posterior, we try to approximate the posterior by another distribution in a tractable family of distribution. By doing so, it becomes an optimization problem. We will describe the algorithm in this section.

Recall that the likelihood of the simple model is

$$Z_{il}|Q_i \sim \text{Discrete}(Q_i) \forall\, i, l, \tag{11}$$

$$p(X_{il} = 0|Z_{i,l} = k, (\theta)) = 1 - \theta_{lk}, \quad p(X_{il} = 0|Z_{i,l} = k, (\theta)) = \theta_{lk}, \tag{12}$$

and that our goal is to calculate the posterior distribution $p((Q_i), (\theta_{lk}), (Z_{il})|G)$, while the prior is given in (3). Instead of using MCMC to sample from the posterior, we now try to approximate $p(Q, \theta, Z|G)$ by a distribution $q^*(Q, \theta, Z)$

$$q^* = \arg\min_{q \in \mathcal{Q}} D_{KL}(q(Q, \theta, Z) \parallel p(Q, \theta, Z|G)) \tag{13}$$

where D_{KL} is the Kullback-Leibler (KL) divergence between two distribution. To make the optimization problem tractable, we need to specify the family $\mathcal{Q}$ of distributions of θ, Z, Q to be simple. Here we choose it to be the mean-field family, that is, they are independent

$$q(Q, \theta, Z) = \prod_{i,l} q(Z_{il}) \prod_i q(Q_i) \prod_{l,k} q(\theta_{lk}), \tag{14}$$

and for each factor, we choose the family of distributions to be

$$q(Z_{il}) = \text{Discrete}(\tilde{Z_{il}}) \tag{15}$$

$$q(Q_i) = \text{Dir}(\tilde{Q_i}) \tag{16}$$

$$q(P_{lk}) = \text{Beta}(\theta_{lk}^{\tilde{u}}, \theta_{lk}^{\tilde{v}}), \tag{17}$$

where $\tilde{Z_{il}}, \tilde{Q_i}, \theta_{lk}^{\tilde{u}}, \theta_{lk}^{\tilde{v}}$ are the parameters of the variational distribution. The chosen family makes the computation tractable and the parametric forms are also conjugate with the distributions in the likelihood. Now we need to find the parameters of the variational distribution to minimize the KL to the true posterior. Because we do not know the posterior, (13) seems to be infeasible. But we can work around it by doing some algebra. From now, when we write expectation, we understand it as taking the expectaion with respect to the variational distributions.

$$D_{KL}(q(Q, \theta, Z) \parallel p(Q, \theta, Z|G)) = E \log \frac{q(Q, \theta, Z)}{p(Q, \theta, Z|G)} \tag{18}$$

$$= E\left[\log \frac{q(Q, \theta, Z)}{p(Q, \theta, Z, G)} + \log p(G)\right] \tag{19}$$

$$= E\left[\log \frac{q(Q, \theta, Z)}{p(Q, \theta, Z, G)}\right] + \log p(G) \tag{20}$$

$$= -\mathcal{E} + \log p(G), \tag{21}$$

because the marginal distribution, $p(G)$ does not depend on Q, θ, Z, where $\mathcal{E}$ is called the "ELBO" (evidence lower bound, because it is less than $\log p(G)$), and defined by

$$\mathcal{E} = E\left[\log \frac{p(Q,\theta,Z,G)}{q(Q,\theta,Z)}\right] \tag{22}$$

$$= \sum_Z \int_{Q,\theta} [\log p(Q,\theta,Z,G) - \log q(Q,\theta,Z)] q(Q,\theta,Z) dQ d\theta \tag{23}$$

$$= \sum_Z \int_{Q,\theta} [(\log p(G|Z,\theta) + \log p(Z|Q) + \log p(\theta) + \log p(Q)) \tag{24}$$

$$- \log q(Q,\theta,Z)] q(Q,\theta,Z) dQ d\theta \tag{25}$$

$$= \sum_Z \int q(Z,P) \log p(G|Z,\theta) d\theta \tag{26}$$

$$+ \sum_Z \int q(Z,Q) \log p(Z|Q) dQ \tag{27}$$

$$- D_{KL}(q(Q) \parallel p(Q)) - D_{KL}(q(\theta) \parallel p(\theta)) - E[\log q(Z)]. \tag{28}$$

It is noticeable that in the paper, they have a typo in the last equation where the sign should be "-" like us instead of "+", and they also miss the term $E[\log q(Z^1, Z^2)]$ (as they consider diploid data). Using the results in Appendix D, we can write the $\mathcal{E}$ in terms of the variational parameters only. This function is concave w.r.t. each variational parameters, and its derivatives have closed form representations. Hence we can optimize it by iterative updates of the parameters. The rest of calculation is presented in Appendix B. It is claimed that fastSTRUCTURE is around 2 orders of magnitude faster than STRUCTURE, with the price of the accuracy of the posterior distribution.

3 VLAD algorithm

In this section, we describe the Dirichlet Simplex Nest and VLAD algorithm as developed by Mikhail et al. [13]. To make it simple to demonstrate, we consider haploid data (X_{il}) and will talk about diploid data later (!!).

We first define the so-called Dirichlet Simplex Nest generating model and show it is equivalent with our model above. Suppose that we have K vector $\theta_1, \ldots, \theta_K \in \mathbb{R}^L$, define $T = \text{Conv}(\theta_1, \ldots, \theta_K)$ its simplex. For each $i = 1, \ldots, N$, we generate a random vector $Q_i = (q_{ik})_{k=1}^K \sim \text{Dir}_K(\alpha)$ (Dirichlet distribution with parameter $(\alpha, \alpha, \ldots, \alpha)$ (K times)), and take $\mu_i := \sum_{k=1}^K q_{ik}\theta_k$. The data point X_i is generate by $X_i|\mu_i \sim F(\mu_i)$, where F is a probability kernel such that $E[X_i|\theta] = \mu_i$.

The main goal of VLAD algorithm is using geometry to learn all (θ_k) and (Q_i) from the data (X_i). The spirit of the algorithm is that, meanwhile X_i and $\mu_i \in \mathbb{R}^L$ are very high-dimensional data, but all μ_i live in a low-dimensional linear manifold (the convex hull of $\theta_1, \ldots, \theta_K$), and we can take advantage of that to learn.

To see the connection to the likelihood model that we have been working up to now, we recall it.

$$X_{il}|Z_{il} = k, (\theta) \sim \text{Ber}(\theta_{kl}), \quad Z_{il}|Q_i \sim \text{Discrete}(Q_i). \tag{29}$$

By integrating out the latent variable Z, we have

$$P(X_{il} = 1|Q_i, (\theta)) = \sum_{k=1}^{K} \theta_{kl} q_{ik} =: \mu_{il}. \tag{30}$$

In other words, $X_{il}|Q_i, (\theta) \sim \text{Ber}(\mu_{il})$, and therefore for data X_i of individual i,

$$X_i|Q_i, \Theta \sim \prod_{l=1}^{L} \text{Ber}(\mu_{il}) =: F(\mu_i), \quad Q_i \overset{iid}{\sim} \text{Dir}_K(\alpha) \tag{31}$$

where $\mu_i = (\mu_{il})_{l=1}^{L} = \Theta Q_i$ (we treat μ_i as a column vector $\in \mathbb{R}^L$, $Q_i \in \mathbb{R}^K$, and $\Theta \in \mathbb{R}^{L \times K}$ is the matrix presentation of (θ)). We now clearly see that our likelihood model is a Dirichlet Simplex Nest generating model. Let us now describe how to learn Θ and (Q_i) from this.

First, to motivate the geometry of the problem, we give a definition of Centroidal Voronoi Tessellation (CVT) (Du et al. 1999 [2]).

Definition 3.1 (Centroidal Voronoi Tessellation)**.** Let Ω an open set endowed with a distance d and a measure ρ. Ω is called a is called a Centroidal Voronoi Tessellation if there exists a collection of K points $c_1, \ldots, c_K \in \Omega$ and a partition $V_1, \ldots, V_k$ of Ω such that

$$V_k = \{x \in \Omega : d(x, c_k) < d(x, c_l) \, \forall \, l \neq k\}, \tag{32}$$

where we say V_k is the Voronoi cell of c_k, and

$$c_k = \frac{1}{\int \rho(x) dx} \int_{V_k} x \rho(x) dx, \tag{33}$$

i.e., c_k is the centroid of its own Voronoi cell.

The lemma below say that we can learn the information of (θ_k) if we know the CVT of the data.

Lemma 3.1. *Let $\Theta \in \mathbb{R}^{L \times K}$ to be the matrix form of (θ_k). Suppose it has full column rank. Consider the simplex T of (θ_k) endowed with the distance function $\|\cdot\|_{(\Theta\Theta^T)^\dagger}$ and the image measure of Dirichlet distribution*

$$P_\Theta(S) = P(\{q \in \Delta^{K-1} : \Theta q \in S\}), \tag{34}$$

for any $S \subset T$, where $q \sim \text{Dir}_K(\alpha)$ and $(\Theta\Theta^T)^\dagger$ is the pseudo inverse of $\Theta\Theta^T$. Then the centroids of the CVT of T fall on the line segment fall on the line connecting centroid of T to $\theta_1, \ldots, \theta_K$.

From the lemma above, we can see that if we know the CVT of T, then we can learn (θ_k) by drawing segments from the centroid of T to its CVT centroids, and extend those segments by an appropriate scale. It also suggests that we can find the CVT centroids by doing a scaled $K-$means optimizations

$$\arg\min_{c_1,\dots,c_K} \left\{ \sum_{k=1}^{K} \sum_{x_i \in V_k} (x_i - c_k)^T (\Theta\Theta^T)^\dagger (x_i - c_k) \right\} \tag{35}$$

Unfortunately, we can not do this because $(\Theta\Theta^T)^\dagger$ is unknown. But we can exploit its value from the sample covariance of the data.

Let us first consider the noiseless problem $x_i = \mu_i$ for all i. In this case, the population covariance matrix take the form $\Sigma = BSB^T$, where S is the covariance matrix of $\mathrm{Dir}_K(\alpha)$. It can be calculated that $S = \dfrac{1}{K(K\alpha+1)} P$, where $P = I_K - \dfrac{1}{K}\mathbf{1}_K\mathbf{1}_K^T$ is the centering matrix. For each (x, q), $x = \Theta q$,

$$\bar{x} := x - E[x] = \Theta q - \frac{1}{K}\Theta\mathbf{1} = \Theta q - \frac{1}{K}\Theta\mathbf{1}(\mathbf{1}^T q) = \Theta P q =: \Theta\bar{q}. \tag{36}$$

It suggests that

Lemma 3.2. *The centroids of the CVT of simplex T under the $\|\cdot\|_{(\Theta\Theta^T)^\dagger}$ distance are given by $\{c_0 + c_k^* | k = 1, \dots, K\}$, where $(c_1^*, \dots, c_K^*)$ solves*

$$\min_{c_1,\dots c_K, v_1,\dots,v_K} \sum_{k=1}^{K} \int_{x \in \Theta V_k} (\bar{x} - c_k)^T \Sigma^\dagger (\bar{x} - c_k) \rho(x) dx. \tag{37}$$

and $c_0 = \int x\rho(x)dx$ is the centroid of T.

Now we can introduce the Voronoi Latent Admixture (VLAD) algorithm for data. We change the population covariance matrix Σ to the sample covariance matrix Σ_N. Moreover, instead of finding (c_k) in a high-dimensional setting, we can utilize SVD to find it faster. Let $\bar{X} \in \mathbb{R}^{N \times L}$ be the centered data, and its SVD $\bar{X} = U\lambda V^T$. Then $\Sigma_N = \dfrac{1}{n} W\Lambda^2 W^T$ and

$$(\bar{x}_i - c_k)^T \Sigma_N^\dagger (\bar{x}_i - c_k) = n(u_i - \eta_k)^T \Lambda W^T W \Lambda^{-1} W^T W \Lambda (u_i - \eta_k) = n \left\| u_i - \eta_k \right\|^2, \tag{38}$$

where $\bar{x}_i = W\Lambda u_i$ and $c_k = W\Lambda\eta_k$. Thus, we only need to perform $K-$means on K dimensional space. This greatly speeds up the algorithm. After learning (θ_k), Q_i can be learned by projecting the data point x_i on the simplex T and calculate the barycentric coordinate of the projected points.

Algorithm 1: VLAD algorithm

Input: data $x_1, \ldots, x_N$; number K, extension parameter γ
Output: simplex vertex $\theta_1, \ldots, \theta_k$
1: $\hat{c_0} \leftarrow \frac{1}{n}\sum_i x_i$ (find the center) ;
2: $\bar{x_i} \leftarrow x_i - \hat{c_0}, i = 1, \ldots, n$ (centering) ;
3: Compute top $K-1$ singular vector of centered data $\bar{X} = U\Lambda W^T$;
4: $\eta_1, \ldots, \eta_k \leftarrow$ K-means$(u_1, \ldots, u_n)$, where u_i is the $i-$th row of $U \in \mathbb{R}^{N\times(K-1)}$;
5: $\hat{c_k} \leftarrow W\Lambda\eta_k + \hat{c_0}$;
6: $\hat{\theta_k} \leftarrow \hat{c_0} + \gamma(\hat{c_k} - \hat{c_0})$

Now we need to calculate the extension parameter γ. Because it has not been known to have a closed form, we can estimate it by Monte Carlo method knowing α

Algorithm 2: Estimating γ based on α

1: Generate $q_1, \ldots, q_m \sim \mathrm{Dir}_K(\alpha)$;
2: $v_1, \ldots, v_K \leftarrow$ K-means$(q_1, \ldots, q_m)$;
3: $\gamma \leftarrow \sqrt{K^2-K}(\sum_{k=1}^K \left\| v_k - \frac{1}{K}\mathbf{1}_K \right\|_2)^{-1}$;

Finally, to estimate α, we propose a method of moment approach.

$$\hat{\alpha} = \arg\min_{\alpha>0} \left\| \hat{\Theta}(\gamma(\alpha))S(\alpha)\hat{\Theta}(\gamma(\alpha))^T - \hat{\Sigma} \right\|, \tag{39}$$

where $\hat{\Sigma}$ is the sample covariance matrix $\frac{1}{n}\bar{X}^T\bar{X}$, and $S(\alpha)$ is the covariance matrix of $\mathrm{Dir}_K(\alpha)$.

That is all the algorithm in the noiseless case. In general, we observe $x_i \sim F(\mu_i)$ instead of μ_i itself. Because $E[x_i|\mu_i] = \mu_i$, it is not needed to change the algorithm 1 and 2, we only need to recalculate the covariance matrix of DSN generating model.

Lemma 3.3. *The population covariance matrix of X w.r.t. the generating model* (31) *is*

$$C(\alpha) := Diag(\bar{\theta}) - Diag(\bar{\theta} * \bar{\theta}) + \Theta^T S(\theta)\Theta - Diag(Diag(\Theta^T S(\alpha)\Theta)). \tag{40}$$

where $\bar{\theta} = \frac{1}{K}\sum_{k=1}^K \theta_k$*, and* θ_k*'s are calculated based on* α *as in two algorithms above, and* $\bar{\theta} * \bar{\theta}$ *means elementwise product of* $\bar{\theta}$ *with itself (square all elements). We use* $Diag(a)$ *for* $a \in \mathbb{R}^L$ *for the diagonal matrix with diagonal a, and* $Diag(A)$ *for A being a matrix to be its diagonal.*

Proof. For $X = \Theta^T q_i$, where $q \sim \mathrm{Dir}_K(\alpha)$, we have

$$cov(X) = E[cov(X|q)] + cov(E[X|q]) \tag{41}$$

It can be seen that $cov(E[X|q]) = E\mu = \Theta^T S(\alpha)\Theta$, and

$$\begin{aligned} E[cov(X|q)] &= EDiag((\mu_l(1-\mu_l))_{l=1}^L) && (42)\\ &= EDiag((\mu_l)_{l=1}^L) - EDiag((\mu_l^2)_{l=1}^L) && (43)\\ &= Diag(\bar{\theta}) - Diag((E[(\mu_l^2)])_{l=1}^L) && (44)\\ &= Diag(\bar{\theta}) - Diag((cov(\mu_l) + E\mu_l^2)_{l=1}^L) && (45)\\ &= Diag(\bar{\theta}) - Diag(Diag(cov(\mu))) - Diag(\bar{\theta} * \bar{\theta}), && (46) \end{aligned}$$

□

Therefore in our model, we can estimate α by doing a grid search

$$\hat{\alpha} = \arg\min_{\alpha>0} \left\| C(\alpha) - \hat{\Sigma} \right\|. \tag{47}$$

It is noticed that we do not need to perform VLAD many times to do a grid search above, we just need to save $(\hat{c_k})$ and calculate $(\hat{\theta_k})(\gamma(\alpha))$ as a linear function of $(\hat{c_k})$ and $\gamma(\alpha)$. Therefore the running time of this step is negligible.

The demonstration above applies to haploid data. For diploid data, i.e. $(X_{il}^1, X_{il}^2)_{i,l}$. Similar to what in the literature, we model two nucleus sequences as two i.i.d. variables

$$(X_{il}^1)_{l=1}^L, (X_{il}^2)_{l=1}^L \overset{iid}{\sim} F(\mu_i), \tag{48}$$

where F is the vector of independent Bernoulli random variables and $\mu_i = \sum q_{ik}\theta_k$ as described above. We can run the algorithm for $2N$ variables now. After getting the estimations for $(\theta_k)_k$, we can project the mean of X_i^1 and X_i^2 to the convex hull of $(\theta_k)_k$ and compute q_i as the barycenter coordinate.

4 Applications

In this section, we consider the applications of methods that we described above to some simulation data set. The method that we use to generate data is similar to which in [12] and is described below. For $N = 200$ is the number of individuals and $L = 500$ the number of loci (dimension of the data).

1. Set a global admixture proportion Q_0 to be a Dirichlet distribution.
2. For each $i = 1, \ldots, N$, set the admixture proportion of each person $Q_i \overset{iid}{\sim} Q_0$.
3. Choose 5 recombination hot spots by sampling without replacement from $1, \ldots, L$. Then sample $S_{il} \sim \text{Ber}(0.99)$ if l is a recombination hot spot and $S_{il} \sim \text{Ber}(1-\lambda)$ for $\lambda \sim (0.01, 0.5)$ otherwise.

4. For $k = 1, \ldots, K, l = 1, \ldots, L$, sample latent haplotype for $k-$component at locus l by sampling $h_{kl} \overset{iid}{\sim} \text{Ber}(0.25)$.

5. For each i, draw $z_{i1} \sim G_i$, then for each $l = 2, \ldots, L$, set $z_{il} = z_{i(l-1)}$ if $s_{i(l-1)} = 1$, and sample from G_i otherwise.

6. Choose noise level set $\eta = .1$ and set $X_{il} = h_{z_{il}l}$ with probability $1 - \eta$ and sample from $\text{Ber}(0.5)$ with probability η.

In the following, "exact-fitted" refers to models with the correct K (number of ancestor population) and "over-fitted" refers to models with K great than the true number of ancestor populations. "Balanced" means when the global admixture proportion is distributed by a Dirichlet distribution $\text{Dir}((\alpha_k)_{k=1}^K)$ for $\alpha_1 = \cdots = \alpha_K$. In each scenario, we test with 3 algorithms: the original STRUCTURE, the fastSTRUCTURE (VI) and the VLAD algorithm. For the STRUCTURE, we run 1000 MCMC iterations as burn-in and collect data from the next 1000 runs. For the fastSTRUCUTRE, we stop when changes of $q(_{ik})$ less than a threshold.

4.1 Exact-fitted models and balanced data

We simulate the data with the base admixture model $Q_0 = \text{Dir}([.5, .5, .5])$. In the following picture, each individual is presented by a thin vertical stick and different colors on the stick mean admixture proportion of that person w.r.t. different ancestor population. (Figure 4.1.)

It can be seen that all the methods do a good job on recovering the admixture proportion of individuals. Table 1 summary the accuracy (measuring by root mean square error, after switch labels appropriately) and running time

Algorithm	RMSE	running time
STRUCTURE	0.05677	1 hour 20 minutes
fastSTRUCTURE	0.05128	1 minute 18 seconds
VLAD	0.04998	8 seconds

Table 1: Summary table for balanced data

It can be seen that VLAD is around 10 times faster than fastSTRUCTURE, and gives a better approximation to the truth than the others. This may due to the fact that the true generating model is similar to the assumption of VLAD, where we need Q_0 to be a symmetric Dirichlet distribution.

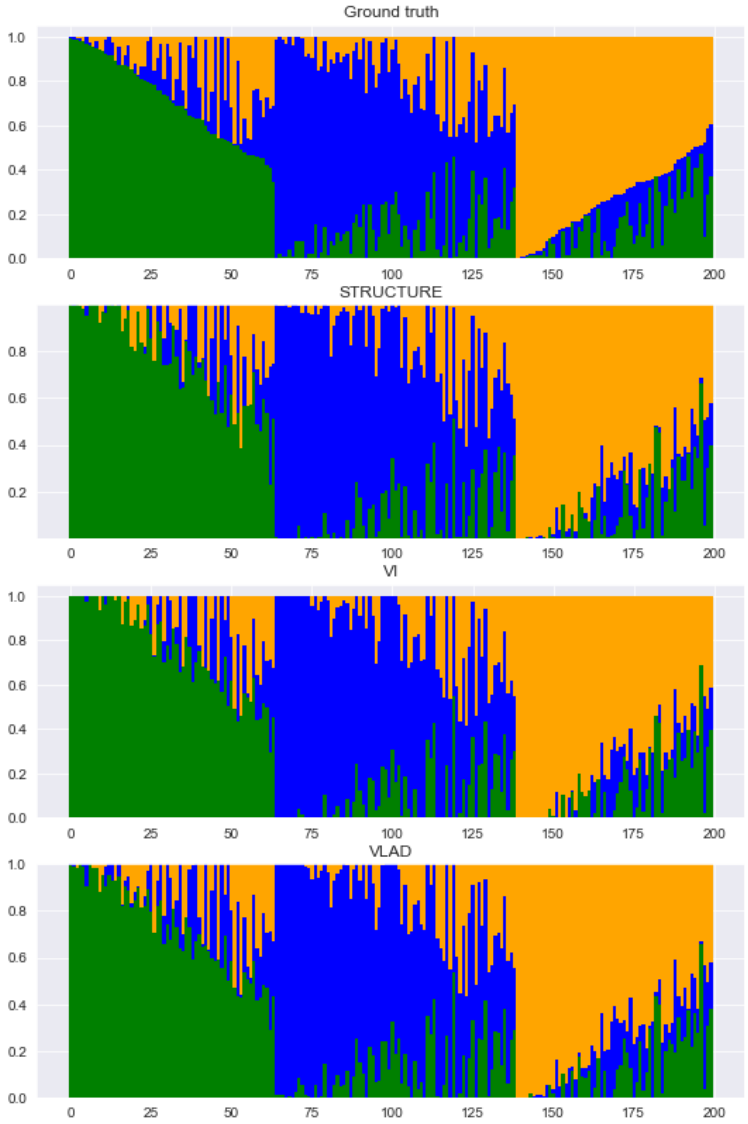

Figure 3: Results from algorithms with balanced data

4.2 Exact-fitted models and unbalanced data

In this test, we simulate data from $Q_0 = [0.6, 0.35, 0.15]$. It can be seen that all algorithms still do well. VLAD seems to be robust even when the true Q_0 does not satisfy its assumption. This observation is also said in [13]. It is also interesting that although VI assumes that all the distribution of Q, Z, θ are independent, it still gives a very good result. (Figure 4.2.)

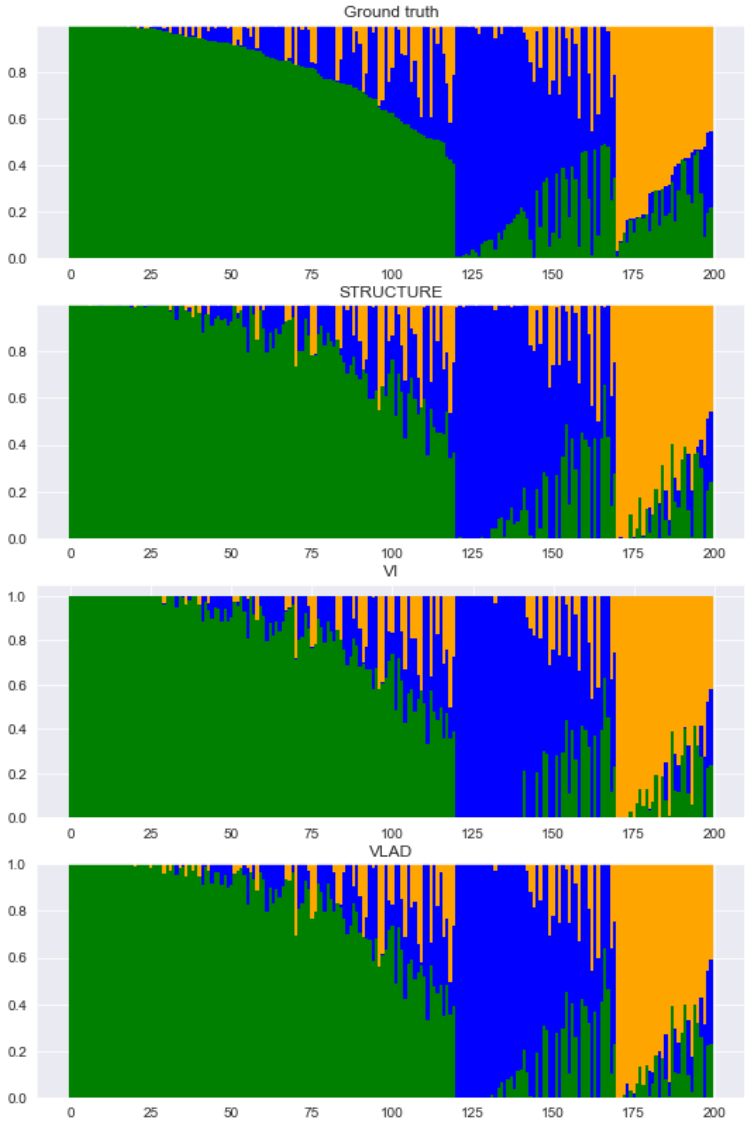

Figure 4: Results from algorithms with unbalanced data

Algorithm	RMSE	running time
STRUCTURE	0.04988	1 hour 20 minutes
fastSTRUCTURE	0.060580	2 minute 10 seconds
VLAD	0.05074	9 seconds

Table 2: Summary table for unbalanced data

4.3 Over-fitted models

We assume that we do not know the true K, and set $K = 5$ to all models. As usually, people often over-fit the number of cluster. The results can be seen in Figure 4.3.

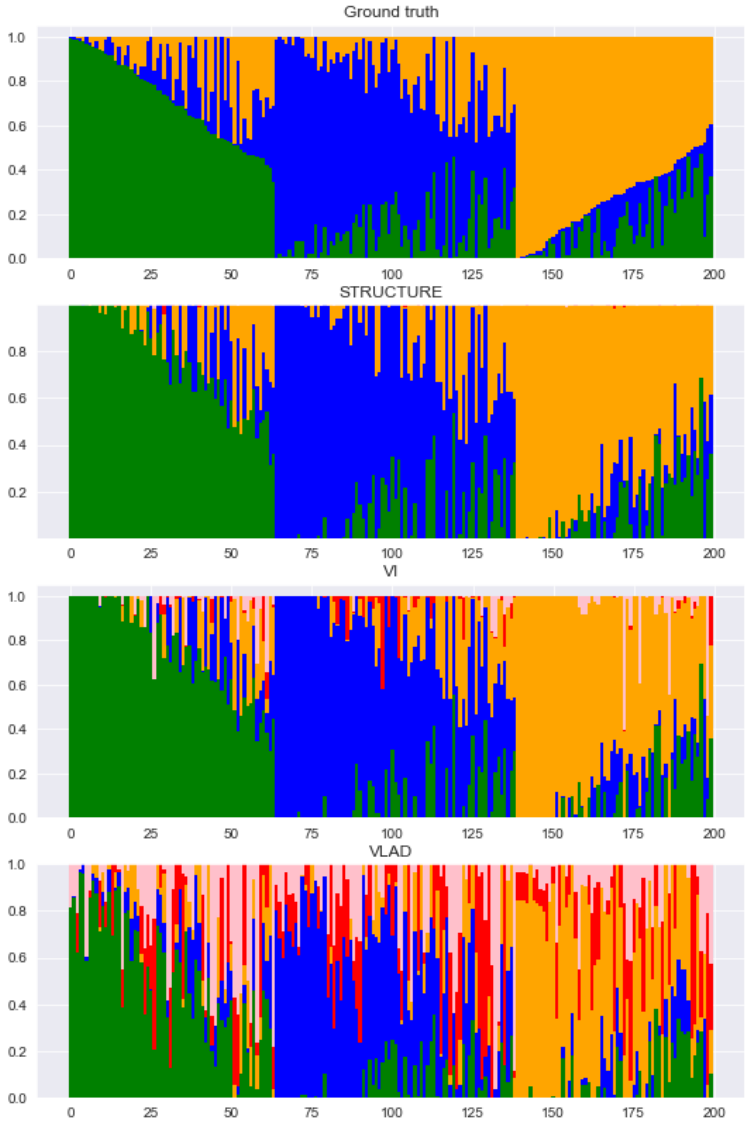

Figure 5: Results of over-fitted algorithms

It can be seen that STRUCTURE is really robust, the fastSTRUCTURE clusters some individuals to have admixture proportion from population 4 and 5, and the VLAD seems to treat all the population equally, although it is able capture some parts of the truth.

We can see that different K can lead to different clustering results. But it is known in literature that there are a lot of ways to choose K ([7], [6], etc). We can also choose K from VLAD itself as there is one part of VLAD using the K-means algorithm. As we exploit its inertia and use the "elbow" method, we can select an appropriate K.

5 Conclusion and Discussion

The project report has reflected some parts of the unsupervised learning algorithms in genetics, although there are a lot more that the author wished to cover. In detailed, it has

1. Reviewed the STRUCTURE model, and some of its variants: including Linkage Disequilibrium modeling, Hierarchical Dirichlet Process, and using Variational Inference to help speeding up the algorithm;
2. Introduced the application of VLAD algorithm in this context;
3. Coded the original STRUCTURE, the fastSTRUCTURE and the VLAD algorithms in jupyter notebook files that can be found on GitHub;
4. Applied those algorithm to simulated data,

and the author wishes to cover

1. Code the Hierarchical Dirichlet Process version of STRUCTURE (I will), learn the slice sampling techniques, use Forward Filtering Backward Sampling for the LD;
2. Apply those methods to real data and give benchmarks;
3. Demonstrate ADMIXTRURE [1] and compare it with VLAD
4. Code the SMARTPCA: a nice touch of high-dimension statistics to test the homogeneity of data [6].

Despite of its incompleteness, through the project, we have seen different perspectives to the same problem: Exploiting the population structure. The application part in Section 4 also gives a intuition view about applying those methods to data and comment about their performances. In the future, we will study those points above that we have not done in this project and try to improve the VLAD algorithm so it can fit data generated from a broader class of distribution (not only symmetric Dirichlet distribution).

Appendices

A MCMC update for STRUCTURE

We will derive the MCMC update for the original STRUCTURE first. By Proposition D.1 and notice that Beta distribution is the Dirichlet distribution in one dimension, we have the update rule for θ

$$\theta_{kl}|(X_{il}^a),(Z_{il}^a) \sim \text{Beta}(\beta + n_{kl}^1, \gamma + n_{kl}^0), \tag{49}$$

where $n^1_{kl} = \#\{(a,i) : (Z^a_{il} = k)\&(X^a_{il} = 1)\}, n^0_{k,l} = \#\{(a,i) : (Z^a_{il} = k)\&(X^a_{il} = 0)\}$. Similarly for $q_i = (q_{ik})_{k=1}^K$,

$$q_i|(Z^a_{il}) \sim \text{Dir}(\alpha_1 + n_{i1}, \ldots, \alpha_K + n_{iK}), \tag{50}$$

where $n_{ik} = \#\{(l,a) : Z^a_{il} = k\}$.

To get the update rule for (Z^a_{il}), we use Bayesian rule

$$P(Z^a_{il} = k|X^a_{i,l} = 1, (\theta), (q)) = \frac{\theta_{kl} q_{ik}}{\sum_{k'} \theta_{k'l} q_{ik'}}, \tag{51}$$

and

$$P(Z^a_{il} = k|X^a_{i,l} = 0, (\theta), (q)) = \frac{(1-\theta_{kl}) q_{ik}}{\sum_{k'} (1-\theta_{k'l}) q_{ik'}}. \tag{52}$$

For the hyperparameter β, γ and $(\alpha_k)_{k=1}^K$, in the paper, they just keep β, γ as constants and might update $(\alpha_k)_{k=1}^K$ by using appropriate Metropolis-Hasting update rule.

B Derivation for VI algorithm's updating scheme

$$\mathcal{E} = \sum_{i,l} \Bigg\{ \sum_k EZ_{ilk} (\mathbb{1}[X_{il} = 0] E[\log(1-\theta_{kl})] + \mathbb{1}[X_{il} = 1] E[\log(\theta_{kl})] + E[\log Q_{ik}]) \tag{53}$$

$$- \sum_k E \log q(Z_{ilk}) \Bigg\} + \log \frac{B(\tilde{\theta^u_{kl}}, \tilde{\theta^v_{kl}})}{B(\beta, \theta)} + (\beta - \tilde{\theta^u_{kl}}) E[\log \theta_{kl}] + (\gamma - \tilde{\theta^v_{kl}}) E[\log(1-\theta_{kl})] \tag{54}$$

$$+ \sum_i \Bigg\{ \sum_k (\alpha_k - \tilde{Q_{ik}}) E[\log Q_{uk}] + \log \Gamma(\alpha_k) - \log \Gamma(\tilde{Q_{ik}}) \Bigg\} + \log \Gamma(\tilde{Q_{n0}}) - \log \Gamma(\alpha_0) \tag{55}$$

Using Proposition D.2, we have

$$E[\log \theta_{kl}] = \psi(\tilde{\theta^u_{kl}}) - \psi(\tilde{\theta^u_{kl}} + \tilde{\theta^v_{kl}}); \;\; E[\log(1-\theta_{kl})] = \psi(\tilde{\theta^v_{kl}}) - \psi(\tilde{\theta^u_{kl}} + \tilde{\theta^v_{kl}}), \tag{56}$$

$$E[\log Q_{ik}] = \psi(\tilde{Q_{ik}}) - \psi(\tilde{Q_{i0}}); \;\; E \log q(Z_{ilk}) = \tilde{Z_{ilk}} \log \tilde{Z_{ilk}} + (1 - \tilde{Z_{ilk}}) \log(1 - \tilde{Z_{ilk}}), \tag{57}$$

where $\psi(\cdot)$ is the digamma function, $\alpha_0 = \sum_k \alpha_k$, $Q_{i0} = \sum_k Q_{ik}$. Hence by optimizing a variable keeping other fixed, we have the update rule for $\tilde{Z}$

$$Z_{ilk} \propto \exp\left\{ \mathbb{1}[X_{il} = 0] \psi(\tilde{\theta^v_{kl}}) + \mathbb{1}[X_{il} = 1] \psi(\tilde{\theta^u_{kl}}) - \psi(\tilde{\theta^u_{kl}} + \tilde{\theta^v_{kl}}) + \psi(\tilde{Q_{ik}}) - \psi(\tilde{Q_{i0}}) \right\}; \tag{58}$$

for $\tilde{Q}$

$$\tilde{Q}_{ik} = \alpha_k + \sum_l \tilde{Z}_{ilk}; \tag{59}$$

for $(\tilde{\theta}^u, \tilde{\theta}^v)$

$$\tilde{\theta}^u_{kl} = \beta + \sum_i \mathbb{1}[X_{il} = 1]\tilde{Z}_{ilk}, \tag{60}$$

$$\tilde{\theta}^v_{kl} = \gamma + \sum_i \mathbb{1}[X_{il} = 0]\tilde{Z}_{ilk}. \tag{61}$$

C Proof of results from Section 3

Proof. (**Lemma 3.1**) Denote $c_1, \ldots, c_K, V_1, \ldots, V_K$ the CVT of the simplex Δ^{K-1} with metric l^2 and probability measure $\mathrm{Dir}_K(\alpha)$. Because of the symmetry, each extreme point of Δ^{K-1} lie on the ray connecting the centroid of Δ^{K-1} and one c_k, $k = 1 \ldots, K$. Now if we can prove that the CVT of T is $\Theta c_1, \ldots, \Theta c_K, \Theta V_1, \ldots, \Theta V_K$, then by the fact that linear transformation of a line is still a line, we will get what we need. Indeed, for any $x = \in \Theta V_k$, we can write $x = \Theta q$, $q \in V_k$, and

$$\begin{aligned} d_{(\Theta\Theta^T)^\dagger}(x, \Theta c_k) &= (\Theta q - \Theta c_k)^T (\Theta\Theta^T)^\dagger (\Theta q - \Theta c_k) && (62)\\ &= \|q - c_k\|_2^2 && (63)\\ &< \|q - c_j\|_2^2 && (64)\\ &= d_{(\Theta\Theta^T)^\dagger}(x, \Theta c_j), && (65) \end{aligned}$$

for all $j \neq k$, because $\Theta^T(\Theta\Theta^T)^\dagger\Theta = Id_K$ (can be seen using SVD decomposition of Θ and the fact that its rank equals K). Moreover, by change of variables formula for induced measure, we also have

$$\int_{\Theta V_k} x P_\Theta(dx) = \int_{V_k} \Theta q P_\Theta(d(\Theta q)) = \int_{V_k} \Theta q \,\mathrm{Dir}(dq) = \Theta c_k. \tag{66}$$

Hence by the definition of CVT, we are done. □

Proof. (**Lemma 3.2**) We need to prove that the CVT centroids of the simplex T (c_k) satisfies $c_k = d_k + c_0$, where d_k is defined from the optimization problem

$$\min_{d_1,\ldots d_K, v_1,\ldots,v_K} \sum_{k=1}^K \int_{x\in\Theta V_k} (\bar{x} - d_k)^T \Sigma^\dagger (\bar{x} - d_k)\rho(x)dx. \tag{67}$$

and d_k lives in the space $\Theta\Delta^{K-1} - c_0$. Because $c_0 = \int x P_\Theta(x) = \Theta\frac{1}{K}1_K$ so we can write $c_k = \Theta v_k$ and $d_k = \Theta(v_k - \frac{1}{K}1_K) = \Theta\overline{v_k}$ for some $v_k \in \Delta^{K=1}$. The objective function of

optimization problem above is proportion to

$$\sum_{k=1}^{K}\int_{V_k}(\bar{x}-B\bar{v}_k)^T(\Theta P\Theta^T)^\dagger(\bar{x}-B\bar{v}_k)P_\Theta(dx)=\sum_{k=1}^{K}\int_{V'_k}(B\bar{q}-B\bar{v}_k)^T(\Theta P\Theta^T)^\dagger(\bar{q}-B\bar{v}_k)P_\Theta(d(\Theta q)) \tag{68}$$

$$=\sum_{k=1}^{K}\int_{V'_k}(BP(q-v_k))^T(\Theta P\Theta^T)^\dagger(BP(q-v_k))P_\Theta(d(\Theta q)) \tag{69}$$

$$=\sum_{k=1}^{K}\int_{V'_k}(q-v_k)^TP\Theta^T(\Theta P\Theta^T)^\dagger\Theta P(q-v_k)\,\mathrm{Dir}(d(q)) \tag{70}$$

$$=\sum_{k=1}^{K}\int_{V'_k}(q-v_k)^TP(q-v_k)\,\mathrm{Dir}(d(q)) \tag{71}$$

$$=\sum_{k=1}^{K}\int_{V'_k}\|q-v_k\|^2\,\mathrm{Dir}(d(q)), \tag{72}$$

as $V_k = BV'_k$ and $P\Theta^T(\Theta P\Theta^T)^\dagger = P$, and because q, v_k live in a linear manifold parallel to the subspace that P project on, so $\|P(q-v_k)\| = \|(q-v_k)\|$. Hence it is equivalent to the fact that $v_1,\dots,v_K, V'_1,\dots,V'_K$ is the CVT of Δ^{K-1}, which implies $c_k = \Theta v_k$. Thus $c_k = c_0 + c_k^*$ as we wish. □

D Auxiliary results

Proposition D.1. *[Conjugacy of Dirichlet Distribution] Suppose that we have the following Bayesian model*

$$\theta \sim \mathrm{Dir}(a_1,\dots,a_K), \tag{73}$$

$$Z_1, Z_2, \dots, Z_L|\theta \overset{iid}{\sim} \mathrm{Discrete}(\theta), \tag{74}$$

then

$$\theta|Z_1, Z_2, \dots, Z_L \sim \mathrm{Dir}(a_1+n_1,\dots,a_K+n_K), \tag{75}$$

where $n_k = \#\{l : Z_l = k\}\ \forall\, k = 1,\dots,K$ (count number of data equals to k).

Proof. We have the posterior pdf

$$f(\theta|(Z_l)) \propto f(\theta)p((Z_l)|\theta) \tag{76}$$

$$\propto \prod_{k=1}^{K} \theta_k^{a_K-1} \prod_{k=1}^{K} \theta_k^{n_k} \tag{77}$$

$$\propto \prod_{k=1}^{K} \theta_k^{a_K+n_K-1}. \tag{78}$$

Hence

$$\theta|Z_1, Z_2, \ldots, Z_L \sim \text{Dir}(a_1 + n_1, \ldots, a_K + n_K), \tag{79}$$

□

Proposition D.2. *If $X \sim \text{Beta}(\beta, \gamma)$, then we have*

$$E[\log X] = \psi(\beta) - \psi(\beta + \gamma), \tag{80}$$

where ψ is the digamma function, which is defined by

$$\psi(x) = \frac{d}{dx} \log \Gamma(x). \tag{81}$$

Proof.

$$\begin{aligned}
E[\log X] &= \frac{\Gamma(\beta+\gamma)}{\Gamma(\beta)\Gamma(\gamma)} \int_0^1 \log x x^{\beta-1}(1-x)^{\gamma-1} dx \\
&= \frac{\Gamma(\beta+\gamma)}{\Gamma(\beta)\Gamma(\gamma)} \int_0^1 \frac{\partial}{\partial\beta} x^{\beta-1}(1-x)^{\gamma-1} dx \\
&= \frac{\Gamma(\beta+\gamma)}{\Gamma(\beta)\Gamma(\gamma)} \frac{\partial}{\partial\beta} \int_0^1 x^{\beta-1}(1-x)^{\gamma-1} dx \\
&= \frac{\Gamma(\beta+\gamma)}{\Gamma(\beta)\Gamma(\gamma)} \frac{\partial}{\partial\beta} \frac{\Gamma(\beta)\Gamma(\gamma)}{\Gamma(\beta+\gamma)} \\
&= \frac{\Gamma'(\beta)}{\Gamma(\beta)} - \frac{\Gamma'(\beta+\gamma)}{\Gamma(\beta+\gamma)} \\
&= \psi(\beta) - \psi(\beta+\gamma),
\end{aligned}$$

where we can exchange the integral with the derivative because both of integrands are absolutely integrable. □

References

[1] ALEXANDER, D. H., NOVEMBRE, J., AND LANGE, K. Fast model-based estimation of ancestry in unrelated individuals. *Genome research 19*, 9 (2009), 1655–1664.

[2] DU, Q., FABER, V., AND GUNZBURGER, M. Centroidal voronoi tessellations: Applications and algorithms. *SIAM review 41*, 4 (1999), 637–676.

[3] ELLIOTT, L. T., DE IORIO, M., FAVARO, S., ADHIKARI, K., AND TEH, Y. W. Modeling population structure under hierarchical dirichlet processes. *Bayesian Anal. 14*, 2 (06 2019), 313–339.

[4] FALUSH, D., STEPHENS, M., AND PRITCHARD, J. K. Inference of population structure using multilocus genotype data: Linked loci and correlated allele frequencies. *Genetics 164*, 4 (2003), 1567–1587.

[5] HARTER, A. V., GARDNER, K., FALUSH, D., LENTZ, D., BYE, R., AND RIESEBERG, L. Origin of extant domesticated sunflowers in eastern north america. *Nature 430* (2004), 201–205.

[6] PATTERSON, N., PRICE, A. L., AND REICH, D. Population structure and eigenanalysis. *PLOS Genetics 2*, 12 (12 2006), 1–20.

[7] PRITCHARD, J. K., STEPHENS, M., AND DONNELLY, P. Inference of population structure using multilocus genotype data. *Genetics 155*, 2 (2000), 945–959.

[8] RAJ, A., STEPHENS, M., AND PRITCHARD, J. K. faststructure: Variational inference of population structure in large snp data sets. *Genetics 197*, 2 (2014), 573–589.

[9] ROSENBERG, N. A., BURKE, T., ELO, K., FELDMAN, M. W., FREIDLIN, P. J., GROENEN, M. A. M., HILLEL, J., MÄKI-TANILA, A., TIXIER-BOICHARD, M., VIGNAL, A., WIMMERS, K., AND WEIGEND, S. Empirical evaluation of genetic clustering methods using multilocus genotypes from 20 chicken breeds. *Genetics 159*, 2 (2001), 699–713.

[10] 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*, 5602 (2002), 2381–2385.

[11] SETHURAMAN, J. A constructive definition of dirichlet priors. *Statistica sinica* (1994), 639–650.

[12] TEH, Y. W., JORDAN, M. I., BEAL, M. J., AND BLEI, D. M. Hierarchical dirichlet processes. *Journal of the American Statistical Association 101* (2004).

[13] YUROCHKIN, M., GUHA, A., SUN, Y., AND NGUYEN, X. Dirichlet simplex nest and geometric inference. In *Proceedings of the 36th International Conference on Machine Learning* (Long Beach, California, USA, 09–15 Jun 2019), K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97 of *Proceedings of Machine Learning Research*, PMLR, pp. 7262–7271.