Nonparametric Clustering with Variational Inference for Tumor Heterogeneity

Brown University, Math-CS Sc.B Honors Thesis

David Liu

Advisor: Professor Ben Raphael

Reader: Professor Erik Sudderth

May 1, 2017

Contents

1 Introduction

Cancer results from an evolutionary process where somatic mutations occur and accumulate in a population of cells. There are many types of mutations that can cause cancer. A mutation that causes genetic variation at a single genomic site is called a *single nucleotide variant (SNV)*. The different lineages which comprise a tumor are known as clones, and the phenomenon of clonal admixture is known as intratumor heterogeneity [1]. The relation of clones with each other is best visualized with a phylogenetic tree, since mutations accumulate within and across subpopulations.

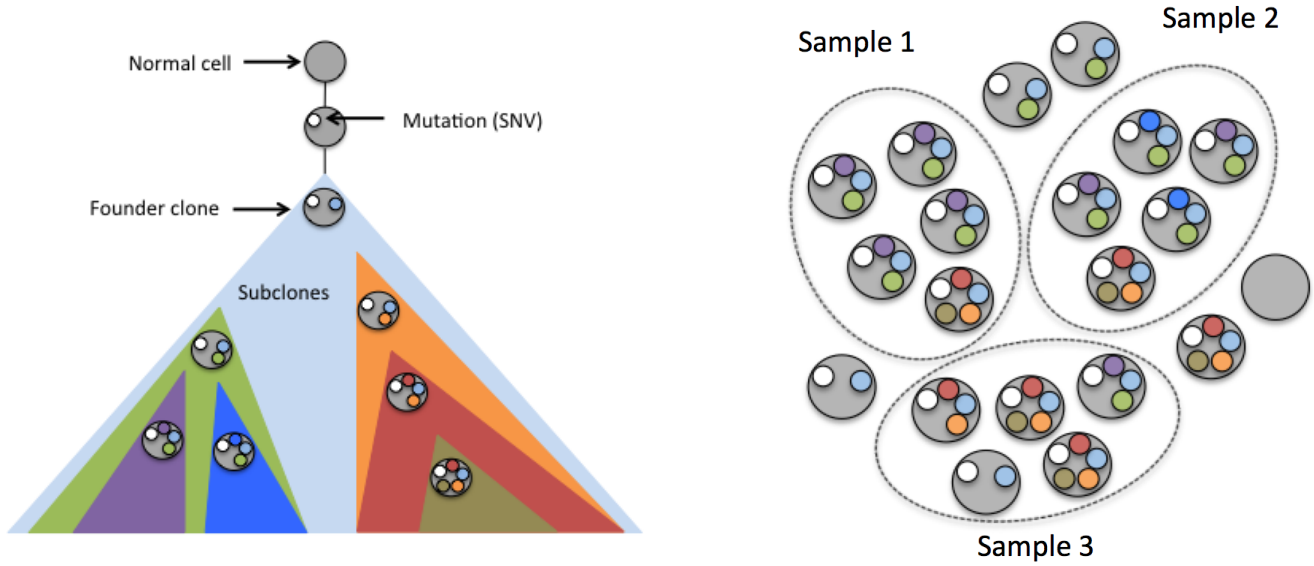

Figure 1: *Left: Example of a clonal tree caused by tumor heterogeneity. Right: Spatial samples from a tumor. Figure from* [2].

The sequencing and analysis of tumors has revealed extensive intratumor heterogeneity in cancers. This is a clinically relevant problem, as the genetic profile of a tumor can lead to treatment failure and drug resistance; increased tumor heterogeneity has been linked with more aggressive cancers [3], [4]. The ability to genetically profile a tumor would improve physicians' ability to tailor treatments according to the subpopulations and mutations present [5].

One of the most studied problems in tumor heterogeneity is tree inference, in which we estimate the evolutionary history and mixing proportions of clones [2]. To do so, we must know which mutations belong to each clone—to have a tree, we must know what the constituent nodes are. This problem is made more tractable by incorporating multiple samples from the tumor (eg. biopsies separated physically or temporally), which provides more information for inference, since the clonal membership of SNVs is invariant across samples. We view the problem of assigning mutations to clones as a general machine learning clustering problem.

2 General Model

2.1 Variant Allele Frequency

The data typically used for tree inference is called the *variant allele frequency (VAF)*, which is a measure of how much a mutation occurs in a population. The VAF is determined by sequencing a sample from a tumor, so that the reads represent the sample's mixture of clones [6]. By comparing to a control sample, if a read contains the mutated allele at a SNV, it is called a variant read; otherwise it is called a reference read. The VAF is defined for each SNV in each sample, as, in each sample, the number of variant reads at an SNV divided by the number of total reads at that SNV. Across multiple samples, each mutation that belongs to a clone should be observed to have about the same variant allele frequency, because the clone frequency is the same; thus a clustering should be true for each clone across all samples.

2.2 Clonal Membership and Generation of Reads

We can express the mathematical dependencies in our model in terms of the processes that generate them: SNVs (labels) are assigned to clones (clusters), and variant reads are generated according to clone-specific parameters. From here on, we will use SNVs to refer to labels, and we will use clusters to refer to clones.

Each SNV $n \in \{1, ..., N\}$ belongs to a cluster $k \in \{1, \dots, K\}$, $K \leq N$. Note that we do not know the true number of clusters in advance, since we do not know how the tumor mutated. These cluster memberships are described by the latent variables $\mathbf{z}_n$, a 1-of-K indicator vector that denotes the cluster assignment of SNV n to cluster k.

Now suppose that for each sample $m \in \{1, \dots M\}$, we have total reads d_{mn} drawn from a Poisson with expected value equal to the coverage [7]. Let SNV n belong to cluster k. Then an integer number of variant reads v_{mn} are generated according to distribution $V(v_{mn}; d_{mn}, \boldsymbol{\phi}_{mk})$, where V is any probability distribution that depends on v_{mn} and has parameters $d_{mn}, \boldsymbol{\phi}_{mk}$. By vectorizing, it is clear that the clustering for a SNV n is fixed across samples m.

$$\boldsymbol{d_n} = \begin{bmatrix} d_{1n} \\ d_{2n} \\ \vdots \\ d_{Mn} \end{bmatrix}, \quad \boldsymbol{v_n} = \begin{bmatrix} v_{1n} \\ v_{2n} \\ \vdots \\ v_{Mn} \end{bmatrix} \tag{1}$$

$$\mathbf{v_n} \doteq \begin{bmatrix} v_{1n} \\ v_{2n} \\ \vdots \\ v_{Mn} \end{bmatrix} \sim \begin{bmatrix} V(v_{1n}; d_{1n}, \boldsymbol{\phi}_{1k}) \\ V(v_{2n}; d_{2n}, \boldsymbol{\phi}_{2k}) \\ \vdots \\ V(v_{Mn}; d_{Mn}, \boldsymbol{\phi}_{Mk}) \end{bmatrix} \doteq \boldsymbol{V(\phi_k)} \tag{2}$$

Let $\mathbf{x}_n$ be general notation for $\{\boldsymbol{d}_n, \boldsymbol{v}_n\}$, where the use of the total or variant reads will be clear from context. Generalizing notation further, let $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, $\boldsymbol{\phi} = \{\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_n\}$, and $\mathbf{z} = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$. Then we wish to discover the underlying $\mathbf{z}, \boldsymbol{\phi}$ for the model given some observations $\mathbf{x}$.

Problem (SNV clustering problem)**.** *Suppose that for SNVs* $n \in \{1, ..., N\}$ *in samples* $m \in \{1, \dots M\}$*. Further suppose that there exists clones (clusters)* $k \in \{1, \dots, K\}$ *and a true clustering* $\mathbf{z}$*. Given total reads* $\boldsymbol{d}_1, \dots, \boldsymbol{d}_n$ *and variant reads* $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$*, we seek to infer* $\boldsymbol{z}$ *and* $\boldsymbol{\phi}$*.*

3 Existing methods

There are existing clustering methods in bioinformatics such as SciClone and PyClone [8, 9]. These methods follow the model described above; two popular choices for V are the binomial and beta distributions. The binomial distribution is a natural choice due to the binary nature of read data. It is also attractive because the number of reads which belong to a cluster naturally weighs the cluster's mixing proportion, which is not necessarily true for the beta model. The beta distribution is also a natural choice for V because variant allele frequencies are in the range $(0, 1)$. That is, V is beta with data $f_{mn} = \frac{v_{mn}}{d_{mn}}$.

However, these methods differ in the prior for the cluster assignments and inference. For example, SciClone uses a fixed number of clusters through a Dirichlet prior and an ad-hoc heuristic for K, with inference through variational inference. PyClone uses a Dirichlet Process prior, and MCMC for inference. Thus PyClone has the same model, but uses a slower inference technique that may have poor convergence properties. On the other hand, SciClone has a different model, but the same inference technique.

State of the art clustering methods use the Dirichlet process to select the number of clusters, since as a nonparametric model, it is more rigorous when the true number of clusters is unknown. MCMC, while accurate in the long run, may be slow and have poor convergence properties, while variational inference is a faster technique that potentially trades off some accuracy for speed and scalability [10]. In this thesis, I attempt to augment existing approaches by proposing and implementing a method to cluster mutations using variational inference for a binomial mixture model with Dirichlet process prior, which is suited for the multi-sample clone mixing problem.

Inference Method \ Model Selection	**Dirichlet Prior + Heuristic (Fixed K)**	**Dirichlet Process Prior (countably infinite K)**
MCMC	(Many older methods)	PyClone
VI	SciClone	*This thesis*

Table 1: A comparison of methods used to solve the clonal mixture problem.

4 Binomial Mixture Model with DP prior

Figure 2 on the next page shows a graphical model representation of the model.

Suppose that each clone in each sample emits variant reads according to a binomial distribution. Thus, for cluster k, some SNV n which belongs to this cluster, and reads d_{mn}, we have variant reads distributed according to $\text{Bin}(v_{mn}; d_{mn}, \boldsymbol{\phi}_{mk})$. Call ϕ_{mk} the cluster frequency for sample m in cluster k. By the independence of reads across samples, the joint probability of reads for an SNV is the product across all samples. Using our vectorized notation,

$$\Pr(\mathbf{x_n}|\boldsymbol{\phi}_k) = \prod_{m=1}^{M} \text{Bin}(v_{mn}; d_{mn}, \boldsymbol{\phi}_k) \tag{3}$$

Let the cluster memberships $\mathbf{z}_n$ and weights π_k be generated by a Dirichlet process prior. The reader is referred to [11] for more mathematical detail on the DP. By truncating the DP at $K = N$, $K \in \{1, \ldots, N\}$ with probability 1.

The likelihood of $\mathbf{x}_n$ depends on the latent variables in a straightforward way from (3):

$$\begin{aligned}\Pr(\mathbf{x_n}|\mathbf{z}, \boldsymbol{\phi}_k) &= \prod_{k=1}^{K} \Pr(\mathbf{x_n}|\phi_{\mathbf{k}})^{\mathbf{z}_{nk}} \\ &= \prod_{k=1}^{K}\prod_{m=1}^{M} \text{Bin}(v_{mn}; d_{mn}, \boldsymbol{\phi}_{mk})^{\mathbf{z}_{nk}}\end{aligned} \tag{4}$$

Then the joint likelihood of the observed data and cluster memberships, follows from (4):

$$\Pr(\mathbf{x_n}, \mathbf{z}|\boldsymbol{\pi}, \boldsymbol{\phi}) = \prod_{k=1}^{K} \pi_k \prod_{m=1}^{M} (\text{Bin}(v_{mn}; d_{mn}, \phi_{mk}))^{\mathbf{z}_{nk}} \tag{5}$$

As described in [12], the Dirichlet Process can be described constructively with a stick-breaking process as follows, for some base measure H and concentration parameter α:

$$\pi_i(\mathbf{v}) = v_i \prod_{j=1}^{i-1}(1 - v_j) \tag{6}$$

$$DP = \sum_{i=1}^{\infty} \pi_i(\mathbf{v})\delta_{\phi_i} \tag{7}$$

1. Draw $v_k|\alpha \sim \text{Beta}(1, \alpha), \quad k = \{1, 2, \ldots\}$
2. Draw $\boldsymbol{\phi}_k|H \sim H^M, \quad k = \{1, 2, \ldots\}$
3. For the nth data point:
   (a) Draw $z_n|\{\pi_1, \pi_2, \ldots\} \sim \text{Cat}(\pi(\mathbf{v}))$.
   (b) Draw $\mathbf{x}_n|z_n \sim p(\mathbf{x}_n|\boldsymbol{\phi}_k)$.

The full cluster assignment posterior

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{z},\mathbf{x})}{p(\mathbf{x})} = \frac{p(\mathbf{z},\mathbf{x})}{\int p(\mathbf{z},\mathbf{x})\, d\mathbf{z}} \tag{8}$$

involves a Dirichlet Process and is thus analytically intractable. We must use some sort of computational technique, such as variational inference, to perform inference on this posterior.

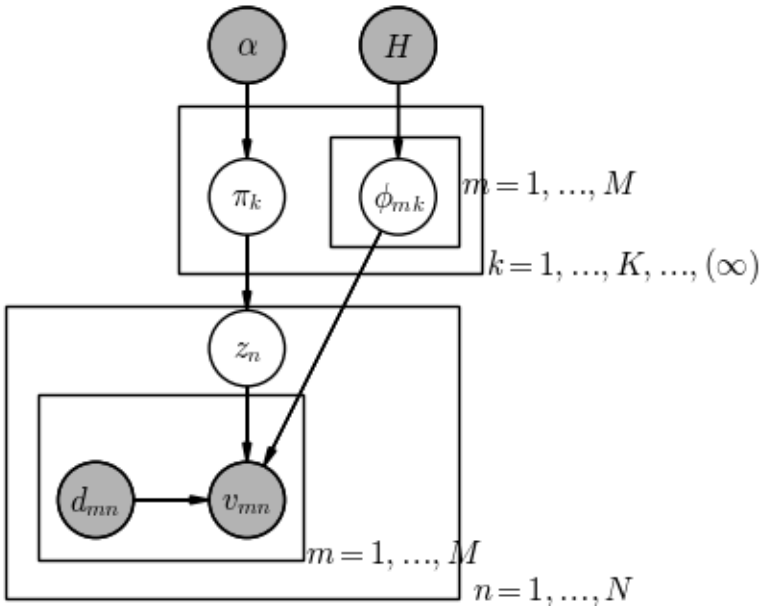

Figure 2: Graphical model for the VAFs.

$n = 1, \ldots, N$: SNVs
$m = 1, \ldots, M$: samples
$k = 1, \ldots, K$: clusters

α = Hyperparameter for the stick-breaking process
$H \sim U(0,1) \sim \text{Beta}(1,1)$ = Base distribution for cluster frequencies (ϕ_{mk})
π_k = Cluster weights, generated from the stick-breaking process
$z_n \in \{1, \ldots, K, \ldots\} \sim \text{Cat}_\infty(\pi_1, \ldots, \pi_K, \ldots)$ = Cluster membership for SNV n
ϕ_{mk} = Cluster frequency
$v_{mn} \sim \text{Binom}(v_{mn}; d_{mn}, \phi_{mk})$ = Observed variant reads for sample m, SNV n, belonging to cluster k.
$d_{mn} \sim \text{Pois}(\text{Coverage})$ = Observed total reads for sample m, SNV n

5 Overview of Variational Inference

Variational inference (VI) is an alternative to MCMC-based inference methods. At a high level, variational inference factors a posterior using the mean-field approximation, which approximates the posterior in a higher-dimensional space using simpler independent functions. Then a simple coordinate ascent can be performed in order to infer the model parameters. The following is a general treatment of VI, where latent variables refer to cluster memberships.

5.1 The ELBO

The following is from [12]. Let $\mathbf{z}$ denote the latent variables, and $\mathbf{x}$ denote the data. We seek to approximate the posterior $p(\mathbf{z}|\mathbf{x})$ from a family of distributions $\mathcal{D}$ by solving the following optimization problem:

$$q^*(z) = \arg\min_{q(\mathbf{z})\in\mathcal{D}} \textsc{kl}\left((q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}))\right). \tag{9}$$

where KL is the KL-divergence, which measures the "distance" between two distributions.

However, (9) requires us to compute the log evidence (which is intractable over the space of all $\mathbf{z}$) since

$$\textsc{kl}\left(q(\mathbf{z})\|p(\mathbf{z}\,|\,\mathbf{x})\right) = \mathrm{E}\left[\log q(\mathbf{z})\right] - \mathrm{E}\left[\log p(\mathbf{z},\mathbf{x})\right] + \log p(\mathbf{x}). \tag{10}$$

Instead, we optimize an objective function which is not dependent on $\log p(\mathbf{x})$. We call this the evidence lower bound (ELBO), which is equal to the negative KL-divergence plus the log evidence.

$$\textsc{elbo}(q) = \mathrm{E}[\log p(\mathbf{z},\mathbf{x})] - \mathrm{E}[\log q(\mathbf{z})]. \tag{11}$$

and thus we see that $\log p(\mathbf{x})$ is a constant with respect to q. The ELBO gets its name from the fact that it is a lower bound for the log evidence.

5.2 The mean-field variational family

The standard technique is to select a simple family of distributions for $\mathcal{D}$, the mean-field variational family [12]. In this family, the latent variables $\mathbf{z}$ are mutually independent so that the joint distribution factorizes:

$$q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j). \tag{12}$$

where q_j is a bounded variation dependent only on z_j. The structure of the model will dictate the optimal form of q_j.

5.3 Coordinate ascent

The optimization is solved using a coordinate ascent algorithm, where via the mean-field assumption, the independence of the latent variables gives us something similar to orthogonality. Let $\mathbf{z}_{-j}$ denote the set of latent variables $\mathbf{z}_l$ such that $l \neq j$. Consider the complete conditional probability of z_j, which is a function

of the other latent variables and the data, $p(z_j \mid \mathbf{z}_{-j}, \mathbf{x})$. Since the expectation in the ELBO is with respect to $q(\mathbf{z})$, which we have assumed factorizes, then we can dissect out the dependence [13] with respect to $\mathbf{z}_j$ by using (11) and (12):

$$\begin{aligned} \text{ELBO}(q) &= \int \prod q_i(\mathbf{z}_i) \left(\log p(\mathbf{z}, \mathbf{x}) - \sum_i \log q_i(\mathbf{z}_i) \right) d\mathbf{z} \\ &\propto \int q_j(z_j) \mathrm{E}_{-j} \left[\log p(\mathbf{x}, \mathbf{z}) \right] \, d\mathbf{z}_j - \int q_j(z_j) \log q_j(\mathbf{z}_j) \, d\mathbf{z}_j \end{aligned} \tag{13}$$

Now suppose that we fix z_{-j} and maximize the ELBO. Then the ELBO is maximized when $\log q_j(\mathbf{z}_j) \propto \mathrm{E}_{-j} \left[\log p(\mathbf{x}, \mathbf{z}) \right]$, by the positivity of the KL-divergence. Thus the optimal $q^*(\mathbf{z}_j)$ occurs when

$$q_j^*(\mathbf{z}_j) \propto \exp \left(\mathrm{E}_{-j} \left[\log p(\mathbf{x}, \mathbf{z}) \right] \right) \tag{14}$$

(14) underlies the coordinate-ascent variational inference algorithm. By iterating through each variational factor, fixing the others, and performing coordinate ascent (similar to Gibbs sampling), then we eventually reach a local optimum of the ELBO.

5.4 Exponential family distributions yield a general formula

If the posterior is in the exponential family, then the computation of coordinate ascent and ELBO can be generalized. Recall that a distribution is in exponential form if it can parameterized by

$$f_X(x \mid \theta) = h(x) \exp \left(\theta^T \cdot T(x) - A(\theta) \right) \tag{15}$$

where $T(x)$ is the sufficient statistic vector, θ is the natural parameter vector, and $A(\theta)$ is the cumulant. The details are in [14], but the intuition is that because the optimal variational updates (14) are proportional to exp(E[log(.)]) then writing the distribution in exponential form reveals dependencies that hold for all exponential family members.

6 Variational Inference on DP Binomial Model

Now we apply the variational inference framework to the model we developed.

6.1 The model and its ELBO

We write the ELBO as a function of the data and latent variables:

$$\begin{aligned}\text{ELBO}\left(q(\mathbf{x}, \mathbf{z}|\gamma, \alpha_0, \beta_0)\right) =& E_q[\log p(\mathbf{v}|\gamma)] + E_q[\log p(\boldsymbol{\phi}|\alpha_0, \beta_0)] + \\ & \sum_{n=1}^{N} \left(E_q[\log p(z_n|\mathbf{v})] + E_q[\log p(x_n|z_n)]\right) \\ & - E_q[\log q(\mathbf{z}, \mathbf{v}, \boldsymbol{\phi})]\end{aligned} \tag{16}$$

where λ represents the hyperparameter governing the stick-breaking process, and α_0, β_0 are the hyperparameters governing the base beta distribution. By the mean-field assumption, the joint distribution for the last term in the ELBO factors as follows:

$$q(\mathbf{z}, \mathbf{v}, \boldsymbol{\phi}) = \underbrace{\prod_{k=1}^{K} q(\boldsymbol{\phi}_k)}_{\substack{\text{Observation: likelihoods} \\ \text{Product of betas} \\ 2MK \text{ variational parameters} \\ \{\alpha_{mk}, \beta_{mk}\}_{m=1,k=1}^{M,K}}} \times \underbrace{\prod_{k=1}^{K} q(\mathbf{v}_k)}_{\substack{\text{Allocation: cluster proportions} \\ \text{Product of betas} \\ 2K \text{ variational parameters} \\ \{\eta_{k0}, \eta_{k1}\}_{k=1}^{K}}} \times \underbrace{\prod_{n=1}^{N} q(z_n)}_{\substack{\text{Allocation: cluster responsibilities} \\ \text{Product of categoricals} \\ NK \text{ variational parameters} \\ \{\hat{r}_{nk}\}_{n=1,k=1}^{N,K}}}$$

Note that the cluster proportions and data assignments are dictated by the Dirichlet Process—we call this the allocation model. On the other hand, the observation parameters vary depending on the structure of the generative model—we call this the observation model (Hughes 2015). Here we derive the form of the $q(\boldsymbol{\phi}_k)$ is a function of $\boldsymbol{\phi}_{\boldsymbol{k}}$, as the observation model is specific to our multi-sample binomial model.

We note that for an individual allelic site in a sample, the data likelihood is binomial. With a beta prior, we know that the resulting posterior for $q(\boldsymbol{\phi}_{mk})$ is conjugate to the binomial, and thus $q(\boldsymbol{\phi}_{mk}) \sim \text{Beta}(\boldsymbol{\phi}_k|\alpha_{mk}, \beta_{mk})$ where α_{mk}, β_{mk} are variational parameters. Because reads across samples at a site are assumed to be independent, then we have

$$\begin{aligned}q(\boldsymbol{\phi}_k) &= \prod_{m=1}^{M} q(\boldsymbol{\phi}_{mk}) \\ &= \prod_{m=1}^{M} \text{Beta}(\boldsymbol{\phi}_k|\alpha_{mk}, \beta_{mk})\end{aligned}$$

6.2 Coordinate ascent algorithm

The derivations of the coordinate ascent algorithm are in Appendices A and B. The derivations for the ELBO are in Appendix C.

The initial responsibilities of the cluster were chosen by setting $\hat{r}_{nk} = 1$ if n was set to be in cluster k by the k-means++ algorithm with $\frac{N}{2}$ initial clusters [14], with the other $\hat{r}_{n\cdot}$ set to be $\frac{1}{k}$. These responsibilities were then normalized. For the other parameters, we assume that they are set to their prior or uniform values. Thus the truncation level of K was set to be N, with all $K > N$ having zero probability.

Since we assumed a uniform prior, $\alpha_0 = \beta_0 = 1$. The other parameters were chosen empirically. We let $\gamma_1 = \eta_1 = 1.0$ and $\gamma_0 = \eta_0 = 1.5$.

We declare the coordinate ascent procedure to be complete when the difference in ELBO between two iterations is less than some convergence threshold. Empirically, we chose the threshold to be equal to 0.01.

The following pseudocode follows the format of [14].

Algorithm 1: CAVI for the DP Binomial mixture model

Input: Data $\mathbf{x}_n$, where each x_i is an integer vector with M entries.
$\gamma_0, \gamma_1, \alpha_0, \beta_0$, hyperparameters

Output: Converged variational parameters $\{\alpha_{mk}, \beta_{mk}\}_{m=1,k=1}^{M,K}, \{\eta_{k0}, \eta_{k1}\}_{k=1}^{K}, \{\hat{r}_{nk}\}_{n=1,k=1}^{N,K}$

Initialize: $\alpha_0 = \beta_0 = \alpha_{mk} = \beta_{mk} = 1, \forall m, k$
$\gamma_1 = \eta_1 = 1.0, \gamma_0 = \eta_0 = 1.5$
$\hat{r}_{nk} \leftarrow \texttt{kmeans++}(\mathbf{x})$

while *the* ELBO *has not converged* **do**

▷ *Compute data-specific (local) parameters*

$\mathrm{E}_q[\log p(x_n|\alpha_{mk}, \beta_{mk})] \leftarrow \mathrm{E}_q[\log \left(\binom{d_{mn}+v_{mn}}{v_{mn}} (\boldsymbol{\phi}_k)^{v_{mn}} (1-\boldsymbol{\phi}_k)^{d_{mn}} \right)]$

$\hat{r}_{nk} \leftarrow \exp(S_k)$

▷ *Compute sufficient statistics*

$S_k = \sum_{n=1}^{N} \hat{r}_{nk} s(x_n) = \sum_{n=1}^{N} \hat{r}_{nk} \left[\begin{bmatrix} v_{1n} & d_{1n} \end{bmatrix} \quad \cdots \quad \begin{bmatrix} v_{Mn} & d_{Mn} \end{bmatrix} \right]$

$N_k = \sum_{n=1}^{N} \hat{r}_{nk}$

$N_k^{>} = \sum_{k+1}^{K} N_k$

▷ *Compute cluster-specific (global) parameters*

$\eta_{k1} \leftarrow 1 + \sum_n \hat{r}_{nk} = 1 + N_k$

$\eta_{k0} \leftarrow \gamma + \sum_n \sum_{j=k+1}^{K} \hat{r}_{nj} = N_k^{>}$

$\alpha_{mk} \leftarrow (\alpha_0 - 1) + S_{km}$

$\beta_{mk} \leftarrow (\beta_0 - 1) + S_{km}$

Compute $\text{ELBO}(q) = \mathbb{E}\left[\log p(\mathbf{z}, \mathbf{x})\right] - \mathbb{E}\left[\log q(\mathbf{z})\right]$

end

return *Converged variational parameters*

6.3 MAP estimates

For each cluster we pool reads by cluster membership:

$$v_{mk}^{\text{pooled}} = \sum_n (v_{mn})^{\mathbf{z}_n} \tag{17}$$

$$d_{mk}^{\text{pooled}} = \sum_n (d_{mn})^{\mathbf{z}_n} \tag{18}$$

and we can make MAP estimates by converting from the variational parameters back to the original parameters of the posterior:

$$\mathbf{z}_n^{\text{MAP}} = \arg\max_k \hat{r}_{nk} \tag{19}$$

$$\phi_{mk}^{\text{MAP}} = \frac{v_{mk}^{\text{pooled}} + \alpha_{mk} - 1}{d_{mk}^{\text{pooled}} + \alpha_{mk} + \beta_{mk} - 2} \tag{20}$$

6.4 Implementation

The VI coordinate ascent algorithm was implemented in Python. The code is available on Github.

7 Experiments and Results

The DP/VI coordinate ascent algorithm was benchmarked against SciClone (VI) and PyClone (DP/MCMC) (see Table 1). These methods were first run on simulated data with the following parameters:

Number of clusters (K)	10
Number of SNVs (N)	100
Number of samples (M)	4, 5, 6
Coverage	50, 100, 1000

Table 2: Parameters for the simulated datasets.

To evaluate the accuracy of cluster assignments ($\mathbf{z}^{\text{MAP}}$), we used the adjusted Rand Index (ARI), which is defined in [15]. The ARI takes as input two clusterings and returns a number in $[0, 1]$, where 0 indicates a totally random clustering, and 1 indicates that the two clusterings are the same. Thus, for a putative clustering $\mathcal{C}$ and the true clustering $\mathcal{K}$, we are interested in $\text{ARI}(\mathcal{C}, \mathcal{K})$.

To evaluate the accuracy of the cluster parameters ($\boldsymbol{\phi}^{\text{MAP}}$), we define the cluster frequency error (CFE), which is the expected error between the putative cluster parameters $\boldsymbol{\phi}_k^{\text{MAP}}$ and the true cluster parameters $\boldsymbol{\phi}_k$ over the putative clusters. Suppose that a clustering algorithm estimates C putative clusters. Then

$$\text{CFE}(\boldsymbol{\phi}^{\text{MAP}}) \triangleq \frac{1}{C} \sum_{c=1}^{C} \min_{k \in \{1,\ldots,K\}} \left\| \boldsymbol{\phi}_c^{\text{MAP}} - \boldsymbol{\phi}_k \right\|. \tag{21}$$

To graphically illustrate a clustering, plots for clusterings were generated, as shown in Figure 3. More examples of plots for data at varying coverages and dimensionality are available in Appendix D. The vertical lines of each color indicate the values of each true cluster parameter. The colored curves are $\text{Beta}(v_{mn}, d_{mn} - v_{mn})$ plots of each data point x_n, in order to show the weight of x_n in a cluster. The thick black curves are the estimated cluster posteriors.

To compare the different methods, violin plots were generated to compare the accuracy measures described above, across the coverages and number of samples outlined in Table 2. These plots follow in Figures 4, 5, 6. PyClone was run with 10000 iterations and a burn-in length of 1000 samples; these were the parameters recommended by PyClone's documentation. SciClone was also run with its default parameters, and a convergence threshold of 0.01.

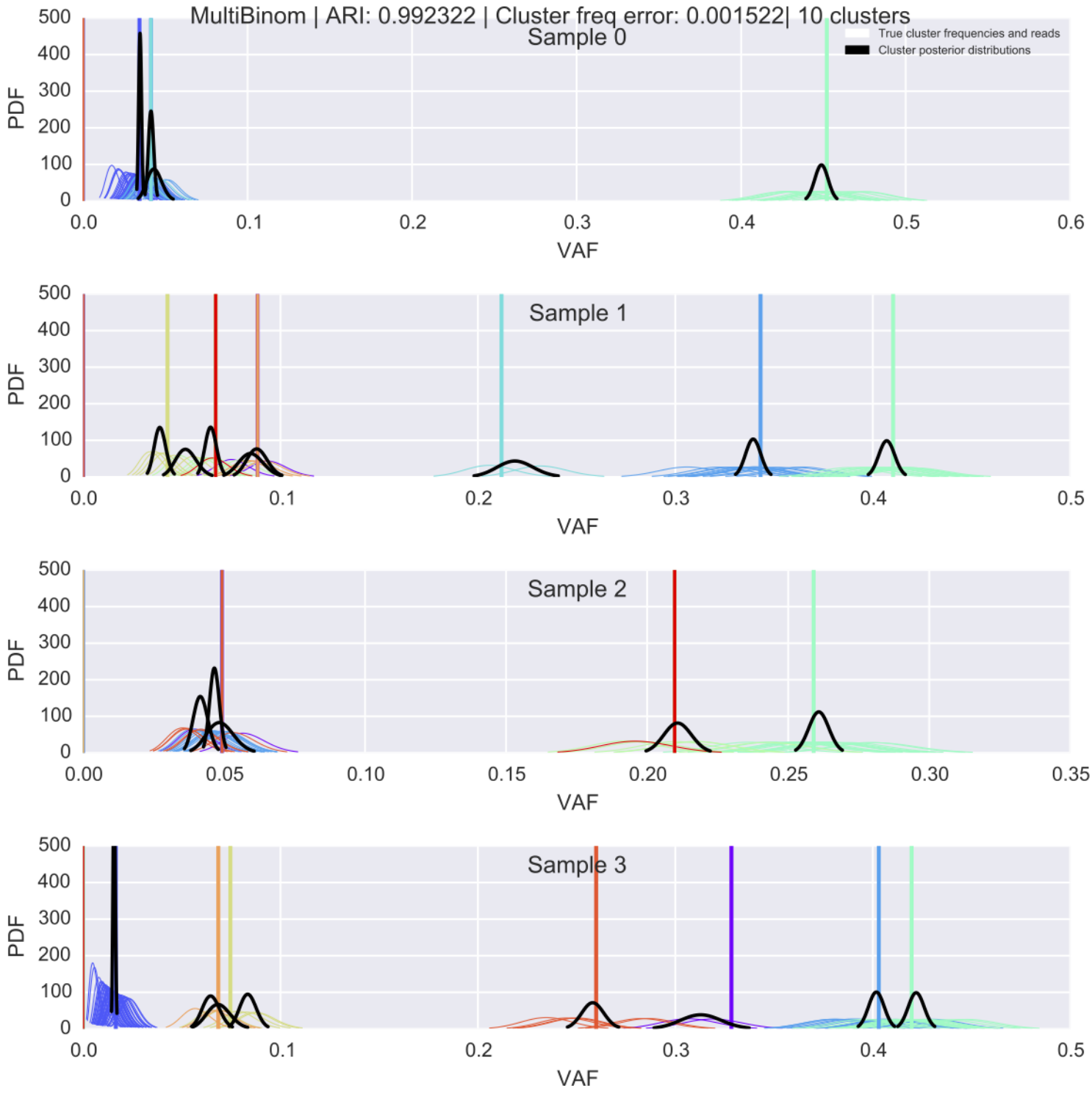

Figure 3: *Cluster posteriors (black) and true clusters (colored, vertical lines) with* $\mathbf{x}_n$ *(colored beta curves).*

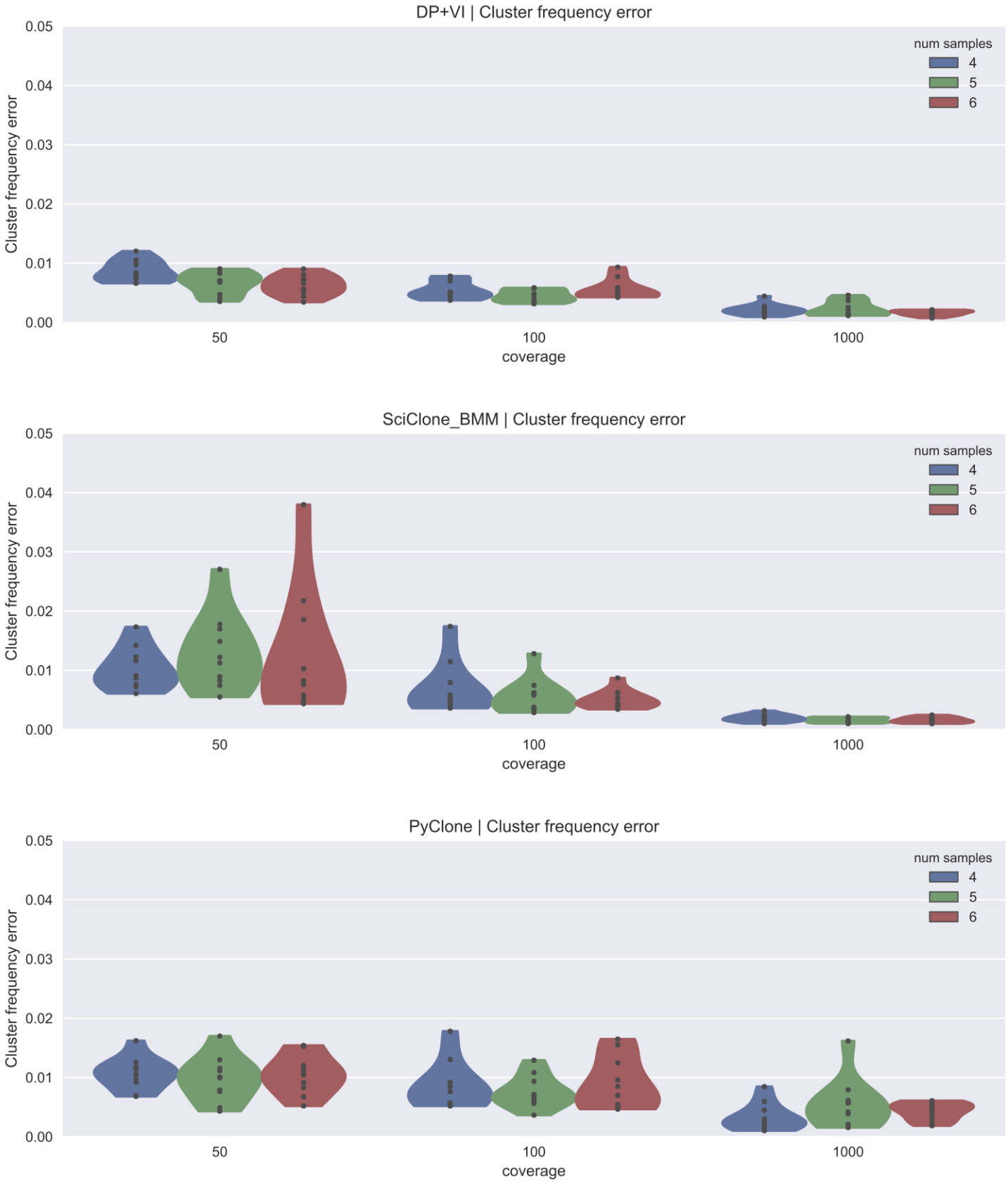

Figure 4: *Comparison of cluster frequency errors on simulated data.*

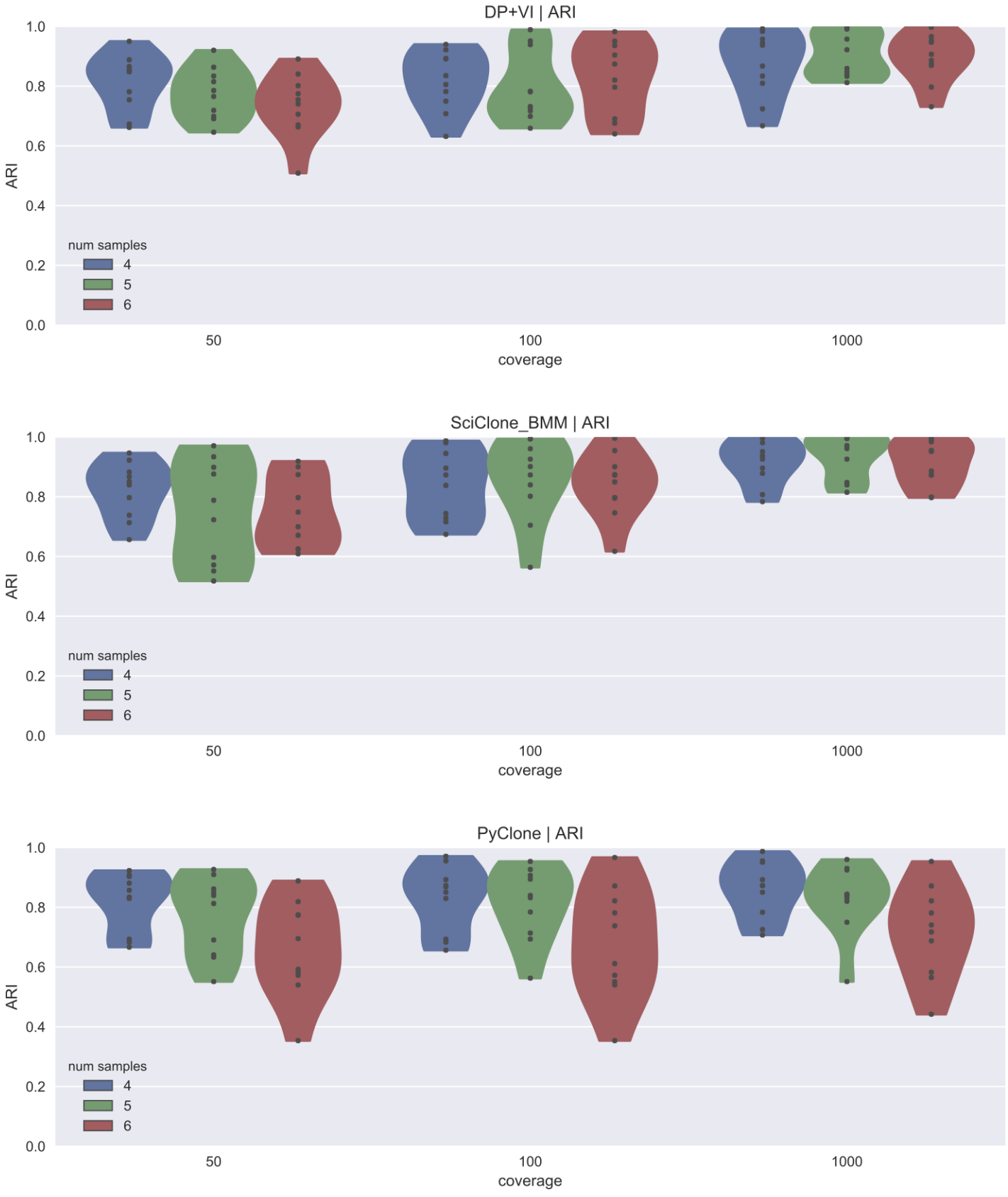

Figure 5: *Comparison of adjusted Rand Index on simulated data.*

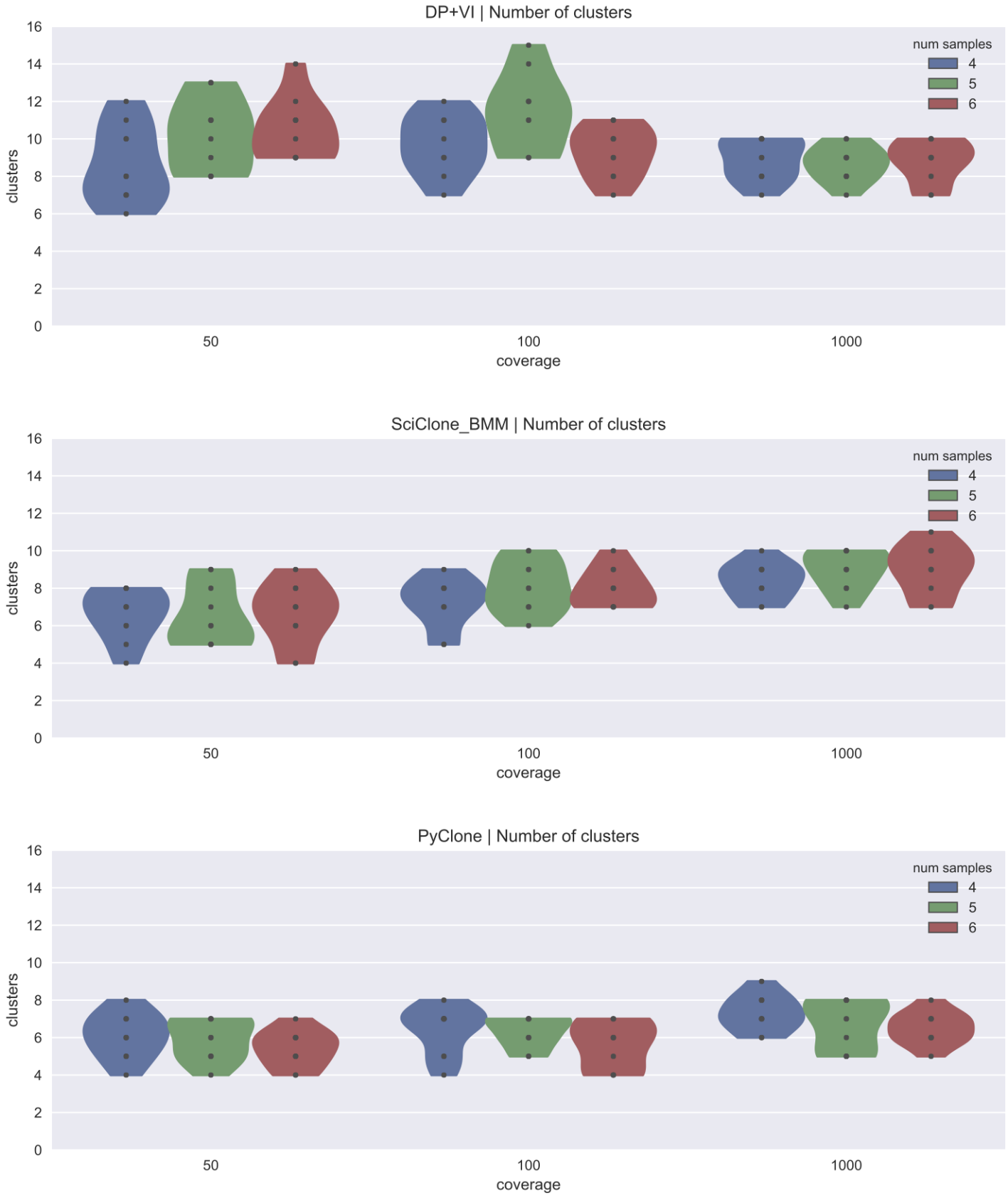

Figure 6: *Comparison of number of clusters on simulated data. There are 10 true clusters.*

Recall that PyClone is DP/MCMC, while SciClone has a Dirichlet prior and uses variational inference. As can be seen from the violin plots, the variational methods outperform MCMC, and the DP/VI method is comparable to SciClone.

The variational methods outperform the MCMC methods in CFE. In the variational methods, the cluster frequency error decreases with both coverage and sample size, whereas the decrease in CFE in PyClone is quite modest. Greater coverage seems to have a larger effect on reducing CFE than adding more samples does. The DP/VI method has a lower cluster frequency error at lower to medium coverage than SciClone.

In terms of ARI, the variational methods have a higher mean ARI that gets closer to 1 as coverage increases. It is somewhat unexpected that the ARI does not increase as the number of samples increases. In fact, in PyClone, the ARI decreases as the number of samples increases, which is likely due to the length of the MCMC run needing to increase to maintain performance as dimensionality increases. However, the DP/VI method seem to be about the same, with SciClone's mean ARI being tighter around 1 as coverage increases than the DP/VI method's. PyClone does not have any clustering with an ARI equal to 1, which means it failed to captures the correct clustering in all cases.

The number of clusters is also an important statistic to consider, since it is tells us how many clones there are in the sample. There are 10 true clusters in each simulation, so we want to know how close the average number of clusters gets to 10. The DP/VI method tends to overestimate the correct number of clusters at low coverage, but gets closer to the true number of clusters as the coverage increases. SciClone and PyClone both underestimate the number of clusters, and the PyClone never correctly recovers all 10 clusters. The number of samples also seems to bring the estimated number of clusters to the true value, so sample size makes a bigger difference here with the number of clusters than in CFE or ARI.

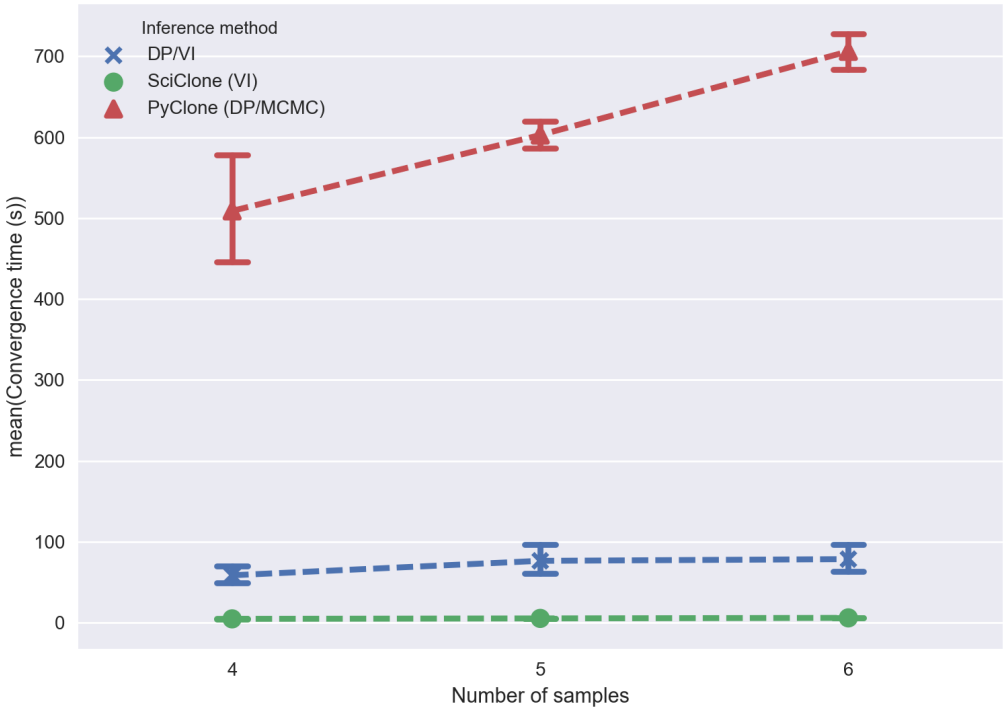

Figure 7: *Mean convergence times and standard error in simulated data sets.*

Since one of the merits of variational inference is speed, we would also like to compare execution times. Figure 7 shows that the variational methods are much faster than PyClone's MCMC inference, especially when more samples causes the dimensionality to increase. However, SciClone is faster because variational inference with the finite-dimensional Dirichlet is faster. Further, SciClone is implemented in R, whereas the DP/VI method is implemented in Python.

8 Discussion

We have shown that variational inference and the Dirichlet Process is accurate, fast, and scalable for solving the SNV clustering problem. Based on experiments, it performs faster and more accurately than the MCMC method presented in PyClone, and performs comparably to the VI method presented in SciClone. However, the DP offers an appealing nonparametric prior that would be better for real-life scenarios where the number of clones is large or unknown. Thus, the DP/VI method is appropriate for real-life tumor sequencing datasets, where the number of SNVs can be quite large (more than 3000), where there may be many samples (ten or more), and where there is no estimate on the true number of clones.

There are natural extensions to both the binomial mixture model and the variational inference used here. It has been shown that in some cases the negative binomial distribution can give rise to a more flexible class of mixture models [16]. Since the negative binomial is an exponential family distribution, it would be feasible to derive a new set of coordinate ascent equations for use with variational inference. Optimizing the code and writing it in a faster language such as C would also close the gap between the nonparametric DP model and the heuristic Dirichlet model in SciClone.

There have also been new developments in variational inference, such as stochastic variational inference [17] which uses a stochastic procedure for the gradient ascent, methods which use richer classes of variational distributions other than the mean-field family [18], and so on. Section 5.4 of [19] provides an overview of open problems in variational inference.

The ultimate purpose of such clustering methods is to be used as part of a pipeline that reconstructs phylogenies. There are many algorithms, such as [2] that can use such a clustering as an input. Naturally, the clusterings produced by this method should be used in order to infer phylogenies in order to be biologically relevant.

9 Acknowledgements

First, I would like to thank Professor Ben Raphael for his mentorship over the past four years. It is due to Ben that I became interested in computational biology and mathematics, and due to Ben that I am happy with my choice to pursue medicine.

I would also like to thank Gryte Satas, who has been teaching me the ropes since freshman year; also thanks to Mohammed El-Kebir, who taught me so much about tumor heterogeneity this past summer. Mike Hughes was instrumental in helping me understand variational inference. I enjoyed the time this summer with my fellow undergraduate labmates over the summer: Michael Mueller, Alex Wong, Jonathan Chang, Eddie Williams. My experience at Brown would not have been the same without the general curiosity and intelligence of everyone in Raphael lab.

Finally, I would like to acknowledge the warmth and support of my family and my friends. To my parents for giving me a boost when I struggled over winter break, and to my fellow thesis-mates at school: Uthsav Chitra, Vaki Nikitopoulos, Keshav Vemuri, and Matt Chin. You guys are amazing.

References

[1] P. Nowell, "The clonal evolution of tumor cell populations," *Science*, vol. 194, no. 4260, pp. 23–28, 1976.

[2] M. El-Kebir, L. Oesper, H. Acheson-Field, and B. J. Raphael, "Reconstruction of clonal trees and tumor composition from multi-sample sequencing data," *Bioinformatics*, vol. 31, no. 12, p. i62, 2015.

[3] Gerlinger *et al.*, "Intratumor heterogeneity and branched evolution revealed by multiregion sequencing," *New England Journal of Medicine*, vol. 366, no. 10, pp. 883–892, 2012. PMID: 22397650.

[4] N. McGranahan and C. Swanton, "Biological and therapeutic impact of intratumor heterogeneity in cancer evolution," *Cancer Cell*, vol. 27, no. 1, pp. 15 – 26, 2015.

[5] B. J. Raphael, J. R. Dobson, L. Oesper, and F. Vandin, "Identifying driver mutations in sequenced cancer genomes: computational approaches to enable precision medicine," *Genome Medicine*, vol. 6, no. 1, p. 5, 2014.

[6] Ding *et al.*, "Clonal evolution in relapsed acute myeloid leukaemia revealed by whole-genome sequencing," *Nature*, vol. 481, pp. 506–510, 01 2012.

[7] E. S. Lander and M. S. Waterman, "Genomic mapping by fingerprinting random clones: A mathematical analysis," *Genomics*, vol. 2, no. 3, pp. 231 – 239, 1988.

[8] C. A. Miller *et al.*, "Sciclone: Inferring clonal architecture and tracking the spatial and temporal patterns of tumor evolution," *PLOS Computational Biology*, vol. 10, pp. 1–15, 08 2014.

[9] A. Roth, J. Khattra, D. Yap, A. Wan, E. Laks, J. Biele, G. Ha, S. Aparicio, A. Bouchard-Cote, and S. P. Shah, "Pyclone: statistical inference of clonal population structure in cancer," *Nat Meth*, vol. 11, pp. 396–398, 04 2014.

[10] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine Learning*, vol. 37, no. 2, pp. 183–233, 1999.

[11] C. E. Antoniak, "Mixtures of dirichlet processes with applications to bayesian nonparametric problems," *Ann. Statist.*, vol. 2, pp. 1152–1174, 11 1974.

[12] D. M. Blei and M. I. Jordan, "Variational inference for dirichlet process mixtures," *Bayesian Anal.*, vol. 1, pp. 121–143, 03 2006.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[14] M. C. Hughes, *Reliable and scalable variational inference for nonparametric mixtures, topics, and sequences*. PhD thesis, Brown University, May 2016.

[15] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[16] M. Zhou and L. Carin, "Negative binomial process count and mixture modeling." arXiv preprint.

[17] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, vol. 14, pp. 1303–1347, 2013.

[18] D. J. Rezende and S. Mohamed, "Variational inference with normalizing flows." arXiv preprint.

[19] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians." Preprint, 2016.

[20] Alizadeh *et al.*, "Toward understanding and exploiting tumor heterogeneity," *Nat Med*, vol. 21, pp. 846–853, 08 2015.

[21] H. Zare, J. Wang, A. Hu, K. Weber, J. Smith, D. Nickerson, C. Song, D. Witten, C. A. Blau, and W. S. Noble, "Inferring clonal composition from multiple sections of a breast cancer," *PLOS Computational Biology*, vol. 10, pp. 1–15, 07 2014.

Appendices

A Allocation model update equations

The coordinate ascent equations for the allocation model are standard [12], as they follow from the fact that the stick-breaking process is in the exponential family. The allocation model has variational parameters $\{\eta_{k0}, \eta_{k1}\}_{k=1}^{K}$ for the cluster proportions and $\{\hat{r}_{nk}\}_{n=1,k=1}^{N,K}$ for the cluster responsibilities. On each iteration, the coordinate update is

$$\eta_{k1} = 1 + \sum_{n} \hat{r}_{nk} = 1 + N_k \tag{22}$$

$$\eta_{k0} = \gamma + \sum_{n} \sum_{j=k+1}^{K} \hat{r}_{nj} = N_k^{>} \tag{23}$$

$$\hat{r}_{nk} \propto \exp(S_k) \tag{24}$$

for $n = 1, \ldots, N$, $k = 1, \ldots, K$, and where

$$S_k = \mathrm{E}_q[\log \boldsymbol{v}_k] + \sum_{i=1}^{k-1} \mathrm{E}_q \log(1 - \boldsymbol{v}_i) + \mathrm{E}_q[\log p(x_n | \alpha_{nk}, \beta_{nk})] \tag{25}$$

and

$$\mathrm{E}_q[\log \boldsymbol{v}_i] = \Psi(\eta_{k0}) - \Psi(\eta_{k0} + \eta_{k1}) \tag{26}$$

$$\mathrm{E}_q[\log(1 - \boldsymbol{v}_i)] = \Psi(\eta_{k1}) - \Psi(\eta_{k0} + \eta_{k1}) \tag{27}$$

The digamma functions come from the fact that derivative of the cumulant is the expectation, and the cumulant of a beta has gamma functions. Since the $\hat{r}_{nk}$ sum to 1 over $n = 1, \ldots, N$ then we renormalize at every step as well.

B Observation model update equations

Following the derivations in [14], we derive the coordinate ascent equations for our observation model, taking advantage of the fact that $q(\boldsymbol{\phi}_k)$ is in the exponential family, which we show below.

Claim 1. *$q(\boldsymbol{\phi}_k)$ is in the exponential family.*

Proof. We know that

$$q(\boldsymbol{\phi}_k) = \prod_{m=1}^{M} q(\boldsymbol{\phi}_{mk}) = \prod_{m=1}^{M} \text{Beta}(\boldsymbol{\phi}_k | \alpha_{mk}, \beta_{mk}).$$

The beta distribution is in the exponential family, with parameterization

$$\begin{aligned} \text{Beta}(\boldsymbol{\phi}_k | \alpha_{mk}, \beta_{mk}) = \frac{1}{\phi_{mk}(1-\phi_{mk})} \exp\Bigg(& \begin{bmatrix} \log \phi_{mk} & \log(1-\phi_{mk}) \end{bmatrix} \begin{bmatrix} \alpha_{mk} \\ \beta_{mk} \end{bmatrix} \\ & + \log\Gamma(\alpha_{mk}+\beta_{mk}) - \log\Gamma(\alpha_{mk}) - \log\Gamma(\beta_{mk}) \Bigg) \end{aligned} \tag{28}$$

and thus $q(\boldsymbol{\phi}_k)$ is in the exponential family with the form

$$\begin{aligned} q(\boldsymbol{\phi}_k) = \left(\prod_{m=1}^{M} \frac{1}{\phi_{mk}(1-\phi_{mk})} \right) \exp\Bigg(& \begin{bmatrix} \begin{bmatrix} \log \phi_{1k} & \log(1-\phi_{1k}) \end{bmatrix} & \cdots & \begin{bmatrix} \log \phi_{Mk} & \log(1-\phi_{Mk}) \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \alpha_{1k} \\ \beta_{1k} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \alpha_{Mk} \\ \beta_{Mk} \end{bmatrix} \end{bmatrix} \\ & + \sum_{m=1}^{M} \log\Gamma(\alpha_{mk}+\beta_{mk}) - \log\Gamma(\alpha_{mk}) - \log\Gamma(\beta_{mk}) \Bigg) \end{aligned}$$

where we have abused notation to show the tuple nature of the sufficient statistics. Thus, for our variational distribution $q(\boldsymbol{\phi}_k)$ we have

$$\text{Natural parameters} = \begin{bmatrix} \begin{bmatrix} \alpha_{1k} \\ \beta_{1k} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \alpha_{Mk} \\ \beta_{Mk} \end{bmatrix} \end{bmatrix} \tag{29}$$

$$\text{Cumulant} = \sum_{m=1}^{M} \log\Gamma(\alpha_{mk}+\beta_{mk}) - \log\Gamma(\alpha_{mk}) - \log\Gamma(\beta_{mk}) \quad \square \tag{30}$$

B.1 Exponential factorization of data model

$$
\begin{aligned}
p(\mathbf{x}_n|\mathbf{z}_n) &= \prod_{m=1}^{M} \mathrm{Bin}(v_{mn}; \phi_{mn}, d_{mn}) \\
&= \prod_{m=1}^{M} \exp\left(v_{mn} \log\left(\frac{\phi_{mn}}{1-\phi_{mn}}\right) + (d_{mn} - v_{mn}) \log(1-\phi_{mn}) \right) \\
&= \left(\prod_{m=1}^{M} \binom{d_{mn}}{v_{mn}} \right) \exp\Bigg(\begin{bmatrix} \log\left(\frac{\phi_{1n}}{1-\phi_{1n}}\right) & \cdots & \log\left(\frac{\phi_{Mn}}{1-\phi_{Mn}}\right) \end{bmatrix} \begin{bmatrix} v_{1n} \\ \vdots \\ v_{Mn} \end{bmatrix} + \\
&\qquad\qquad \begin{bmatrix} \log(1-\phi_{1n}) & \cdots & \log(1-\phi_{Mn}) \end{bmatrix} \begin{bmatrix} d_{1n} \\ \vdots \\ d_{Mn} \end{bmatrix} \Bigg) \qquad (31) \\
&= \left(\prod_{m=1}^{M} \binom{d_{mn}}{v_{mn}} \right) \exp\Bigg(\begin{bmatrix} \begin{bmatrix} v_{1n} & d_{1n} \end{bmatrix} & \cdots & \begin{bmatrix} v_{Mn} & d_{Mn} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \log(1-\phi_{1n}) \\ \log\left(\frac{\phi_{1n}}{1-\phi_{1n}}\right) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \log(1-\phi_{Mn}) \\ \log\left(\frac{\phi_{Mn}}{1-\phi_{Mn}}\right) \end{bmatrix} \end{bmatrix} \Bigg) \qquad (32) \\
&\qquad (33)
\end{aligned}
$$

so that the sufficient statistics are

$$
T(\mathbf{x}_n) = \begin{bmatrix} \begin{bmatrix} v_{1n} & d_{1n} \end{bmatrix} & \cdots & \begin{bmatrix} v_{Mn} & d_{Mn} \end{bmatrix} \end{bmatrix} \qquad (34)
$$

Thus, following (Hughes 2015) we have the following coordinate ascent updates for the observation model: (natural parameter plus sufficient statistic S_k^{var} or S_k^{ref})

$$
\begin{aligned}
\alpha_{mk} &= (\alpha_0 - 1) + \sum_{n=1}^{N} \hat{r}_{nk} \begin{bmatrix} v_{1n} \\ \vdots \\ v_{Mn} \end{bmatrix} \\
\beta_{mk} &= (\beta_0 - 1) + \sum_{n=1}^{N} \hat{r}_{nk} \begin{bmatrix} d_{1n} \\ \vdots \\ d_{Mn} \end{bmatrix}
\end{aligned}
$$

B.2 Sufficient statistics

Define

$$S_k = \sum_{n=1}^{N} \hat{r}_{nk} s(x_n) = \sum_{n=1}^{N} \hat{r}_{nk} \begin{bmatrix} \begin{bmatrix} v_{1n} & d_{1n} \end{bmatrix} & \cdots & \begin{bmatrix} v_{Mn} & d_{Mn} \end{bmatrix} \end{bmatrix} \tag{35}$$

$$N_k = \sum_{n=1}^{N} \hat{r}_{nk} \tag{36}$$

$$N_k^{>} = \sum_{k+1}^{K} N_k \tag{37}$$

B.3 Obs Model Likelihoods

$$\begin{aligned} \mathrm{E}_q[\log p(x_n|\alpha_{mk}, \beta_{mk})] &= \mathrm{E}_q[\log \left(\binom{d_{mn} + v_{mn}}{v_{mn}} (\boldsymbol{\phi}_k)^{v_{mn}} (1 - \boldsymbol{\phi}_k)^{d_{mn}} \right)] \\ &= \log \left(\binom{d_{mn} + v_{mn}}{v_{mn}} \right) + d_{mn} \mathrm{E}_q[\log \boldsymbol{\phi}_k] + v_{mn} \mathrm{E}_q[\log(1 - \boldsymbol{\phi}_k)] \end{aligned}$$

where

$$\begin{aligned} \mathrm{E}_q[\log \boldsymbol{\phi}_k] &= \Psi(\alpha_{mk}) - \Psi(\alpha_{mk} + \beta_{mk}) \\ \mathrm{E}_q[\log(1 - \boldsymbol{\phi}_k)] &= \Psi(\beta_{mk}) - \Psi(\alpha_{mk} + \beta_{mk}) \end{aligned}$$

C Computing the ELBO

To test for convergence, we calculate the ELBO until the difference in ELBO between laps is less than some pre-specified number. For the purpose of this model, the convergence threshold was set to 0.01. Note that the ELBO is generally not a convex function, so we cannot make any guarantees about monotonicity.

Following [14], the ELBO can be decomposed into three terms:

$$\text{ELBO} \coloneqq \mathcal{L} = \mathcal{L}_{\text{Obs}} + \mathcal{L}_{\text{DP-Alloc}} + \mathcal{L}_{\text{Entropy}}$$

C.1 Observation model contribution to ELBO

$$\begin{aligned}
\mathcal{L}_{\text{Obs}} &= \mathrm{E}_{\mathbf{z},\boldsymbol{\phi}}[\log p(x|\mathbf{z},\boldsymbol{\phi})] + \mathrm{E}_{\phi}[\log p(\boldsymbol{\phi})] - \mathrm{E}_{\phi}[\log q(\boldsymbol{\phi})] && (38)\\
&= \sum_{n=1}^{N}\sum_{k=1}^{K} \hat{r}_{nk}\mathrm{E}_q[\log p(\mathbf{x}_n|\boldsymbol{\phi}_k)] \\
&\quad + \sum_{k=1}^{K}\sum_{m=1}^{M} \mathrm{E}_q[\log \boldsymbol{\phi}_k^0] && (39)\\
&\quad - \sum_{k=1}^{K}\sum_{m=1}^{M} \mathrm{E}_q[\log \boldsymbol{\phi}_k]
\end{aligned}$$

C.2 Allocation model contribution to ELBO

$$\begin{aligned}
\mathcal{L}_{\text{DP-Alloc}} &= \sum_{k=1}^{K} c_{\text{Beta}}(1,\gamma) - c_{\text{Beta}}(\eta_{k1},\eta_{k0}) \\
&\quad + \sum_{k=1}^{K} (N_k + 1 - \eta_{k1})\mathrm{E}_q[\log \boldsymbol{u}_k]) && (40)\\
&= \sum_{k=1}^{K} \left(N_k^{>} + \gamma - \eta_{k0}\right) \mathrm{E}_q[\log(1-\boldsymbol{u}_k)]
\end{aligned}$$

where the expectations are defined above, and in (18), we showed that c_{Beta} has the form

$$c_{\text{Beta}}(\alpha,\beta) = \log\Gamma(\alpha+\beta) - \log\Gamma(\alpha) - \log\Gamma(\beta) \tag{41}$$

C.3 Entropy contribution to ELBO

$$\mathcal{L}_{\text{Entropy}} = -\sum_{k=1}^{K}\sum_{n=1}^{N} \hat{r}_{nk}\log\hat{r}_{nk}$$

D Examples of clustering posterior plots (DP/VI)

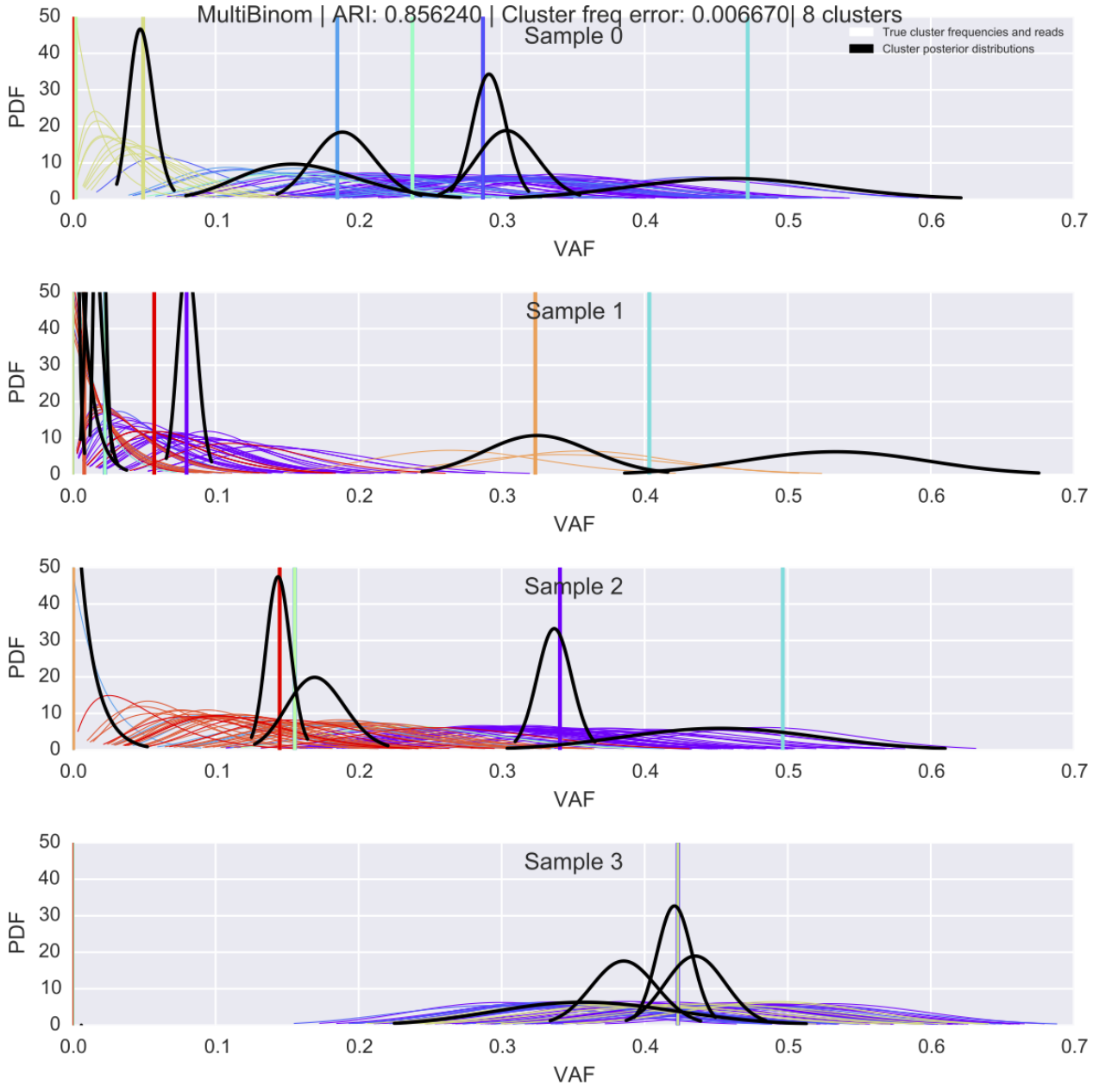

Figure 8: *Low coverage plot, fewer samples.*

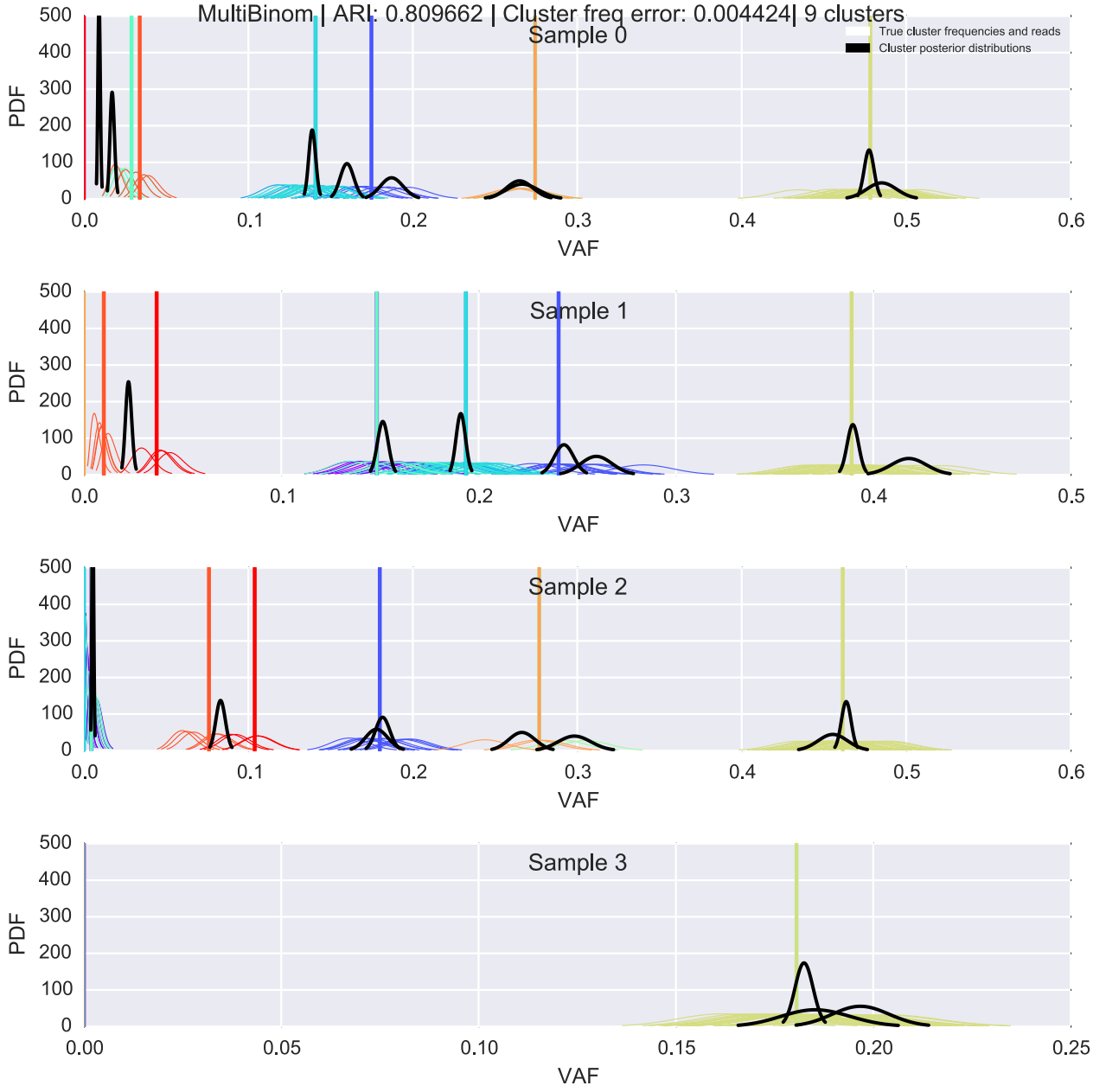

Figure 9: *High coverage plot, fewer samples.*

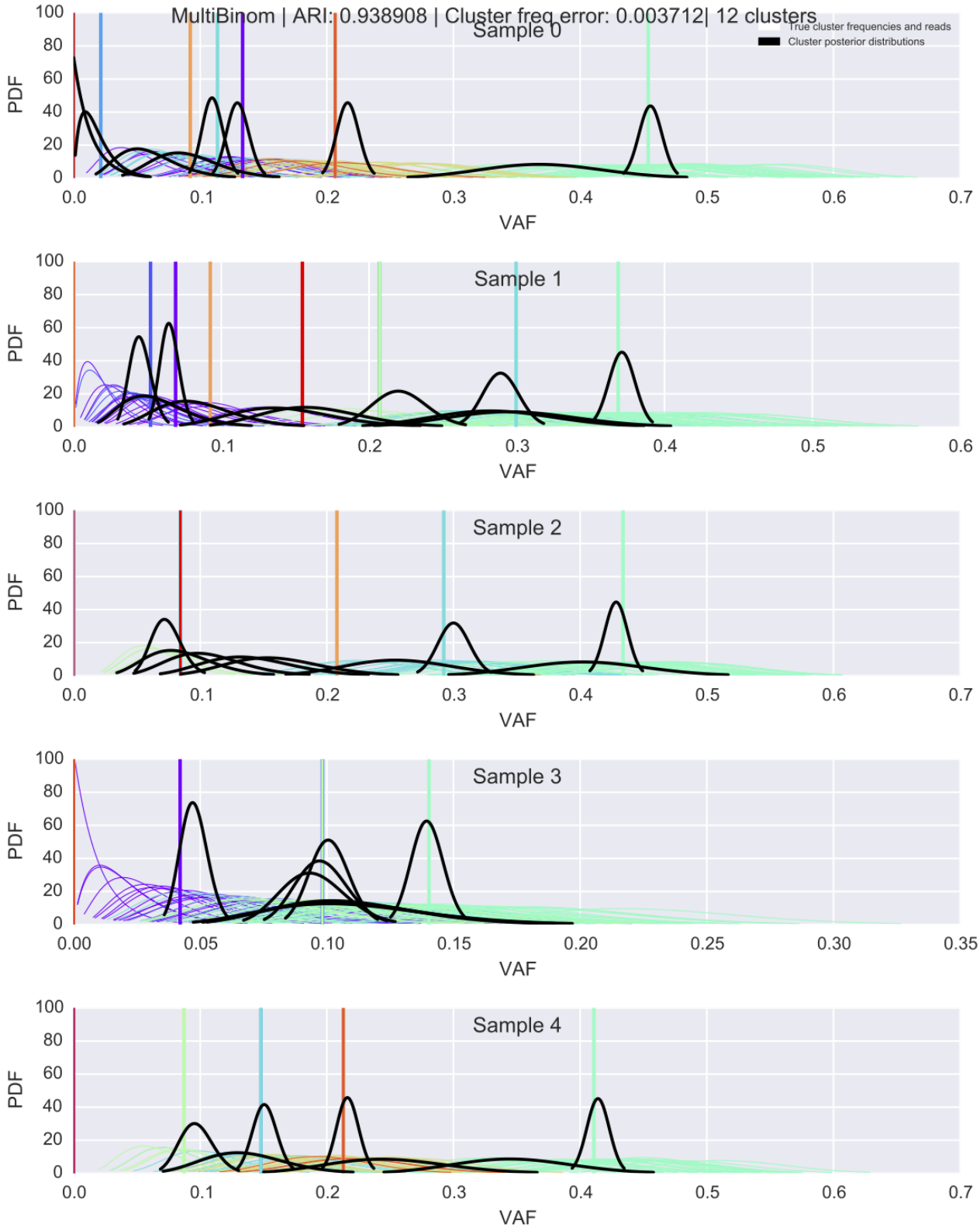

Figure 10: *Medium coverage, medium number of samples.*

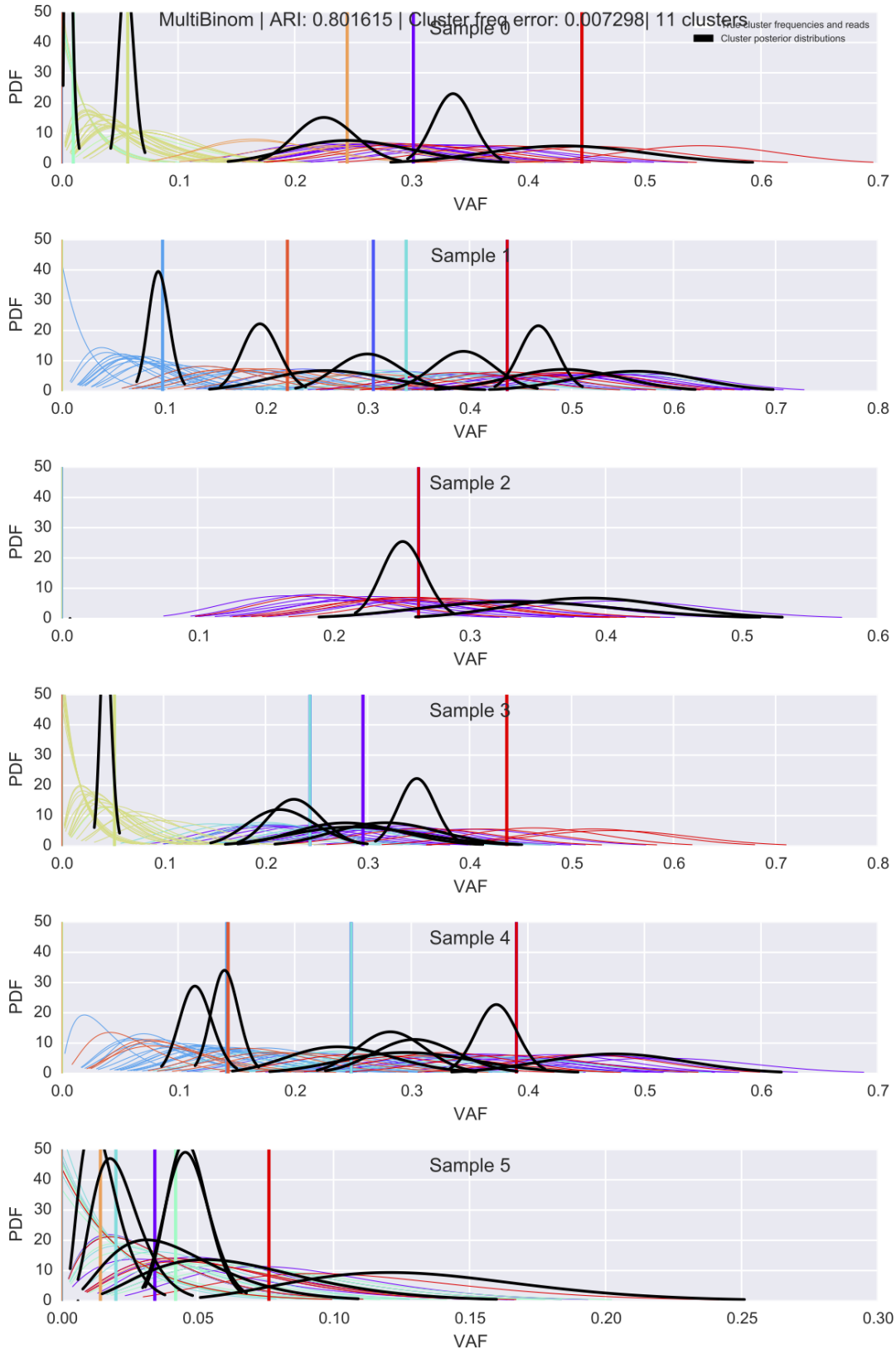

Figure 11: *High coverage plot, more samples.*

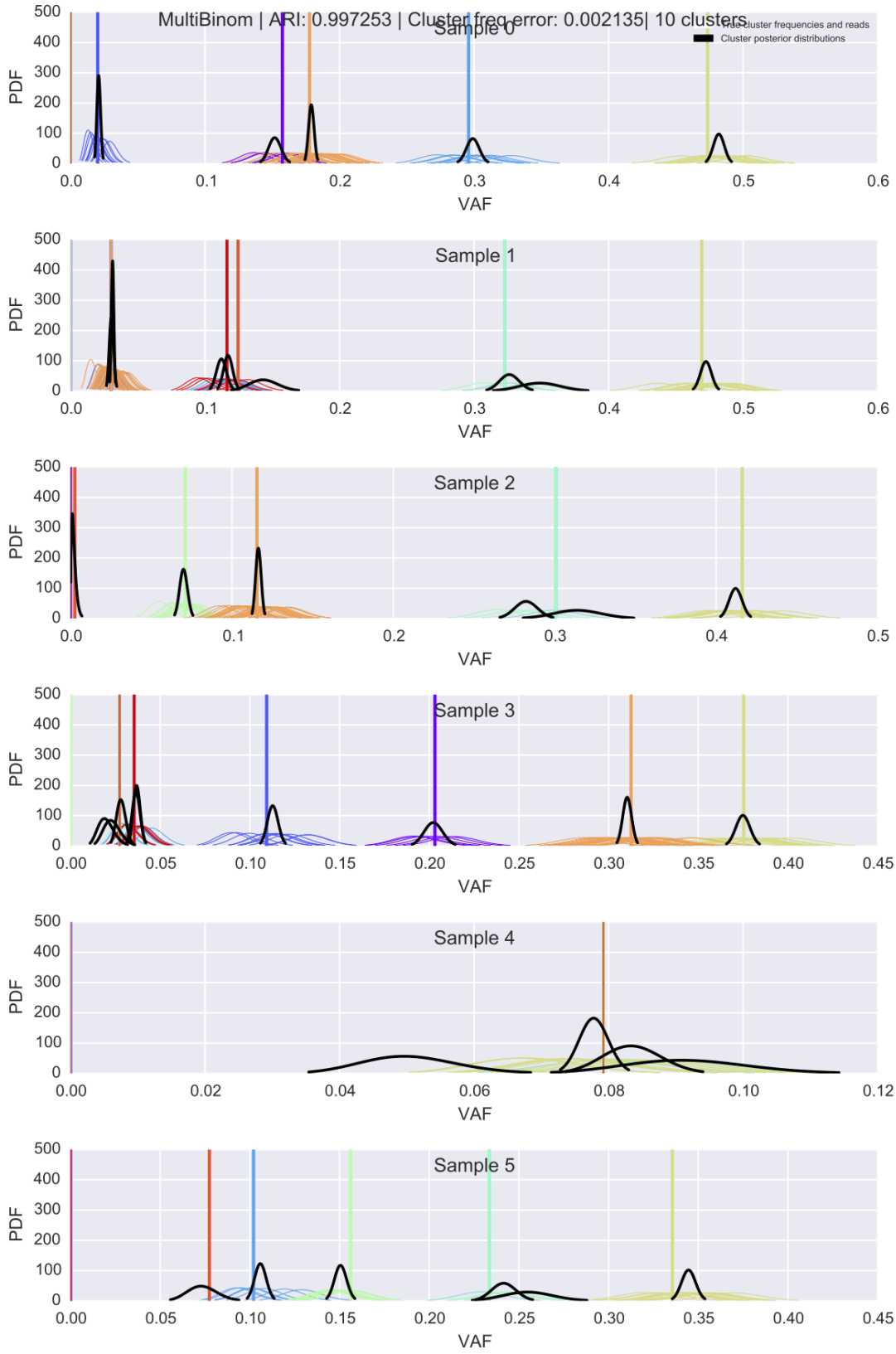

Figure 12: *High coverage plot, more samples.*