CopyMix: Mixture Model Based Single-Cell Clustering and Copy Number Profiling using Variational Inference

Negar Safinianaini, Camila P. E. de Souza[2], Andrew Roth[4,5], Hazal Koptagel[3], Hosein Toosi[1,3], Jens Lagergren[1,3]

[1]School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden [2]Department of Statistical and Actuarial Sciences, University of Western Ontario, London, ON, Canada [3]Science for Life Laboratory, Solna, Sweden [4]Department of Computer Science, University of British Columbia, Vancouver, BC, Canada [5]British Columbia Cancer Agency, Vancouver, BC, Canada

*Correspondence: negar7918@gmail.com

Abstract

Investigating tumor heterogeneity using single-cell sequencing technologies is imperative to understand how tumors evolve since each cell subpopulation harbors a unique set of genomic features that yields a unique phenotype, which is bound to have clinical relevance. Clustering of cells based on copy number data obtained from single-cell DNA sequencing provides an opportunity to identify different tumor cell subpopulations. Accordingly, computational methods have emerged for single-cell copy number profiling and clustering; however, these two tasks have been handled sequentially by applying various ad-hoc pre- and post-processing steps; hence, a procedure vulnerable to introducing clustering artifacts. Moreover, clonal copy number profiling has been missing except for one method, CONET, which unfortunately computes it by a post-processing tool. Finally, a common copy number profiling tool, HMMcopy, requires parameter tuning. We avoid the clustering artifact issues and provide clonal copy number profiles without the labor of parameter tuning in our method, CopyMix, a Variational Inference for a novel mixture model, by jointly inferring cell clusters and their underlying copy number profile. We evaluate CopyMix using simulated data and published biological data from ovarian cancer. The results reveal high clustering performance and low errors in copy number profiling. These favorable results indicate a considerable potential to obtain clinical impact by using CopyMix in studies of cancer tumor heterogeneity.

Keywords: Mixture Models; Variational Inference; Copy number profiling; Single-cell

1 Introduction

A tumor typically consists of a collection of heterogeneous cell populations, each having distinct genetic and phenotypic properties, in particular, concerning the capacity to promote cancer progression, metastasis, and therapy resistance Eirew et al. (2015); Nowell (1976). Single-cell sequencing technologies Gawad et al. (2016); Navin et al. (2011); Shapiro et al. (2013); Zahn et al. (2017) provide an opportunity to investigate the genomic profile of individual cells regarding both single nucleotide variation (SNV) and copy number variation (CNV). CNVs and SNVs are essential contributors to phenotypic variation

relating to health, and disease Baslan et al. (2012); Lawson et al. (2018). Although single-cell SNV profiling is hampered by experimental imperfections such as drop-outs, copy number profiling, i.e., detecting single-cell CNVs, is feasible, at least at coarser resolutions. Clustering cells based on their copy number profiles improves understanding of tumor subpopulations and tumor heterogeneity, issues bound to have clinical relevance.

Current single-cell datasets pose a wealth of computational challenges. As answers to some of those, methods have emerged for single-cell copy number profiling and clustering; some methods infer clustering after copy number profiling, e.g., Garvin et al. (2015); Zahn et al. (2017); Leung et al. (2017); Vitak et al. (2017); Zaccaria and Raphael (2021), while a recent method, CONET Markowska et al. (2022), derives copy number profiles by post-processing in addition to a user-defined sequence of breakpoints per cell. Unfortunately, these methods perform single-cell copy number profiling and clustering sequentially with various ad-hoc processes. The sequential approach is vulnerable to artifacts since preprocessing decisions, typically irreversible, constrain all later analyses. Even if each task performs optimally, the final result may still fall short of the best possible performance Blocker and Meng (2013). Another problem is when using HMMcopy Shah et al. (2006); Vitak et al. (2017); Zahn et al. (2017) for copy number profiling—copy number profiles can naturally be modeled by a sequence of latent variables forming Hidden Markov Models (HMMs)—that has limitations such as requiring manual calibration of more than ten parameters Mallory et al. (2020b). Moreover, as addressed recently by CONET Markowska et al. (2022), one should account for the fact that copy number profiles are generated by a clonal process; however, mixtures of Hidden Markov Models (MHMMs) Smyth (1997), a joint inference alternative for performing the two biological tasks, consider a different copy number profile per cell, thus lacking the clonal copy number profiling.

By a joint inference solution, also motivated by future directions of computational modeling in single-cell cancer genomics Zhang and Campbell (2020), we propose a novel framework referred to as CopyMix to alleviate three problems: (1) the sequential treatment of copy number profiling and clustering; (2) the labor of HMMcopy parameter tuning; (3) lack of clonal copy number profiling by MHMMs. CopyMix performs the joint inference of copy number profiling and clustering, and as to the best of our knowledge, no earlier work has simultaneously performed these two tasks. Similarly, joint inference has been considered for single-cell DNA methylation Kapourani and Sanguinetti (2019); de Souza et al. (2020), single-cell SNV data Roth et al. (2016), and bulk chIP-seq data from several replicates Zuo et al. (2016). Due to the model-based treatment, CopyMix enjoys the advantages of a fully probabilistic framework, such as transparency, uncertainty measurements, and modeling flexibility. CopyMix, a biologically meaningful tangent of MHMMs, uses a novel mixture model with components (expressing the clonal process as opposed to cell-specific MHMMs) corresponding to clusters, each having a specific copy number profile, revealing the copy number variation pattern behind each cluster. We deploy a Bayesian treatment and infer all quantities of interest using Variational Inference (VI) Jordan et al. (1999), which typically yields faster inference methods than methodologies like Markov chain Monte Carlo sampling Blei et al. (2017). Compared to Expectation-Maximization (EM), VI has multiple advantages; e.g., it estimates the posterior distributions rather than point estimates, protects against over-fitting and allows for principled model selection, i.e., identifying the optimal number of mixture components Bishop (2006). Finally, our novel VI simplifies the calculations needed for the posterior approximation of the HMM part, while earlier methods McGrory and Titterington (2009b) use complex solutions to normalize the posterior approximation.

We evaluate CopyMix by experiments on biologically inspired simulated data and two published bio-

logical data Laks and McPherson (2019); Leung et al. (2017). In Laks and McPherson (2019), HMMcopy is used as a copy number profiling tool proceeding by an ad-hoc combination of clustering methods: K-means, Hierarchical Density-Based Spatial Clustering, and UMAP. We also compare CopyMix results with Ginkgo Garvin et al. (2015)—a state-of-art framework for clustering of single-cells based on previously estimated copy number profiles obtained by circular binary segmentation method Olshen et al. (2004), outperforming HMMCopy in copy number profiling although HMMCopy provides higher quality for breakpoint detection Mallory et al. (2020a). Lastly, we extend CopyMix to a so-called SNV-CopyMix for exploiting the SNV signals by incorporating SNVs in the proposed model.

A sequential version of CopyMix, resembling Shah et al. (2006); Vitak et al. (2017) but excluding their ad-hoc processes, is compared to CopyMix, confirming the hypothesis that a joint inference framework is superior over a sequential version of the framework.

2 CopyMix

CopyMix is a probabilistic clustering method based on a mixture of Markov chains. CopyMic is a mixture model with components corresponding to clusters, each having a specific copy number profile modeled by a sequence of latent variables. Similarly, as in Shah et al. (2006); Vitak et al. (2017); Zahn et al. (2017), we assume that each latent copy number sequence is governed by a Markov chain (describing a sequence of possible copy numbers in which the probability of each copy number value, defined as a state, depends only on the previous state in the sequence). We provide two observational models for CopyMix. The first one is based on Poisson emission, and the second one is based on Gaussian emission. In the following sections, we describe these two choices of emission modeling.

After presenting our method, CopyMix, we extend CopyMix to incorporate single nucleotide data; we refer to this method as SNV-CopyMix. The purpose of designing SNV-CopyMix is to exploit CNV-independent signals, i.e., SNVs, when performing single-cell clustering.

2.1 Poisson emission

We assume that the data (read counts) follow a Poisson distribution as in Witten (2011); we consider read counts per fixed equal size genomic bin as in Leung et al. (2017). Moreover, we assume the read counts are emitted from a latent sequence of copy number states. Fig. 1 illustrates an overview of CopyMix; it analyzes the input cell sequences, containing read counts per predefined genomic bin, and produces clusters of cells with their corresponding copy number sequences (profiles) explaining the clusters. Fig. 2 illustrates CopyMix graphical model where Y denotes the observable variables, read counts per predefined bins, C the latent copy number states forming a Markov chain, and Z the latent cell-specific cluster assignment variables. We notice in Fig. 2 that each Y has two levels of dependencies, which are reflections of the assumption that the Poisson distribution over the read counts depends on a latent cell-specific cluster assignment, Z, and a corresponding latent copy number state, C. Intuitively, a higher copy number should correspond to a higher read count. We incorporate this assumption in our model by defining the rate of Poisson distribution as the product of copy number state and cell-specific rate, θ_n, see Eq. (1); this also implies that θ_n corresponds to the average sequencing coverage for a haploid genome. Due to this multiplicative structure for the Poisson rate, a copy number state of zero results in the lowest rate implying copy number deletion events; however, since the rate of Poisson cannot be zero,

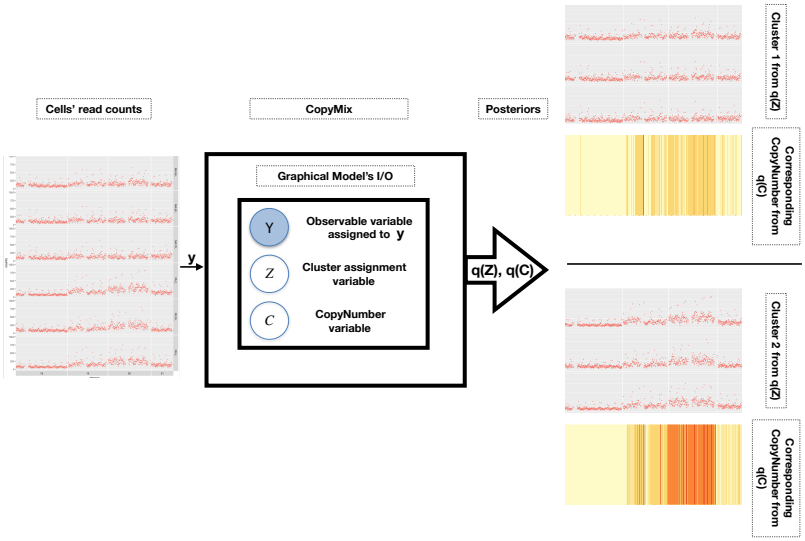

Figure 1: The overview of CopyMix with: input y, cells' reads; outputs $q(Z)$ and $q(C)$, posterior distributions resulting in clusters within y with their corresponding copy number profiles illustrated by heatmaps (this is an example of binary clustering).

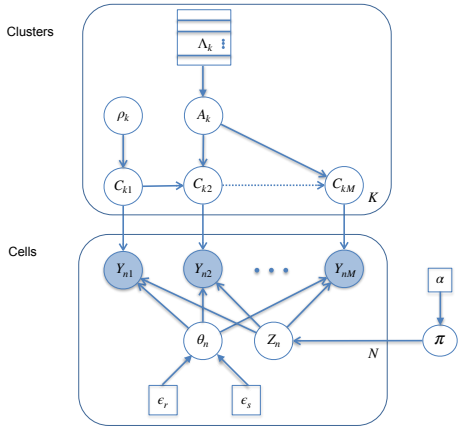

Figure 2: Probabilistic graphical model representing our proposed model. Shaded nodes represent observed values, the unshaded ones are the latent variables and the squares are the hyperparameters of the model; a posterior distribution over the values of the unshaded nodes is approximated using Variational Inference. Y_{nm}, observed read counts from cell n and bin m; C_{km}, corresponding latent copy number state forming a Markov chain; θ_n, cell-specific rate; Z_n, latent cell-specific cluster assignment variable. π and A_k, the priors over Z_n and C_{km} respectively; ρ_k, prior over the starting copy number state.

our implementation handles this by assigning a tiny number instead of zero. Our model is described in more detail in what follows, using conjugacy regarding the priors.

The genome considered is partitioned into M equally sized segments of consecutive positions called bins. Let $\mathbf{Y}_n = (Y_{n1}, \ldots, Y_{nM})$, where Y_{nm} is the random but observable number of reads aligned to bin m for cell n taking values in $\{0, 1, 2, \ldots\}$ for $m \in [M] = \{1, \ldots, M\}$ and $n \in [N] = \{1, \ldots, N\}$. We assume that $\mathbf{Y}_1, \ldots, \mathbf{Y}_N$ are independent and given a vector of true copy number states, $Y_{n1}, \ldots, Y_{nM}$ are also independent with the distribution of Y_{nm} depending on the true copy number state at bin m. The cluster membership of cell n is indicated by the hidden variable Z_n that takes values in $[K] = \{1, \ldots, K\}$. We assume there are $K << N$ vectors of true hidden copy number states, i.e., one for each cluster. The variables $Z_1, \ldots, Z_N$ are independent following a categorical distribution with $P(Z_n = k) = \pi_k$ and $\sum_{k=1}^{K} \pi_k = 1$. If $Z_n = k$ then the distribution of $\mathbf{Y}_n$ depends on the k-th vector of true hidden copy number states, defined as $\mathbf{C}_k = (C_{k1}, \ldots, C_{kM})$, with each C_{km} taking values in $[J] = \{1, \ldots, J\}$. We assume that $C_{k1}, \ldots, C_{kM}$ follow a discrete-time homogeneous Markov chain with initial probabilities $\rho_{kj} = P(C_{k1} = j)$, $j \in [J]$ and transition probabilities $a_{ij}^k = P(C_{km} = j | C_{km-1} = i)$, $i, j \in [J]$. Consequently, given the cluster assignment and the corresponding true hidden vector of copy number states, $Y_{n1}, \ldots, Y_{nM}$ are independent with Y_{nm} following a distribution with parameters depending on the hidden true state at bin m for cluster k, that is, $Y_{nm} | Z_n = k, C_{km} = j \sim F_{j\theta_n}$. As mentioned earlier, we assume $F_{j\theta_n}$ to be a Poisson distribution with rate $j \times \theta_n$, that is,

$$F_{j\theta_n}(y) \equiv \frac{(j \times \theta_n)^y e^{-(j \times \theta_n)}}{y!} \text{ with } j\theta_n > 0. \tag{1}$$

We, w.l.o.g., consider the initial probabilities, ρ_{kj}'s, to be fixed and known. We let $\mathbf{\Psi}$ be the set containing all the unknown model parameters, i.e., $\mathbf{\Psi} = \{\mathbf{A}, \boldsymbol{\theta}, \boldsymbol{\pi}\}$, where $\mathbf{A} = \{\mathbf{A}_k : k \in [K]\}$ with $\mathbf{A}_k = \{a_{ij}^k : i, j \in [J]\}$; $\boldsymbol{\theta} = \{\theta_n : n \in [N]\}$; and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$. In order to infer $\mathbf{\Psi}$, the hidden states $\mathbf{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_K\}$, and $\mathbf{Z} = (Z_1, \ldots, Z_n)$ we apply VI; that is, we derive an algorithm that, for given data, approximates the posterior distribution of the parameters by finding the Variational Distribution (VD), $q(\mathbf{Z}, \mathbf{C}, \mathbf{\Psi})$, with smallest Kullback-Leibler divergence to the posterior distribution $P(\mathbf{Z}, \mathbf{C}, \mathbf{\Psi} | \mathbf{Y})$, which is equivalent to maximizing the evidence lower bound (ELBO) Blei et al. (2017) given by

$$\text{ELBO}(q) = \text{E}[\log P(\mathbf{Y}, \mathbf{Z}, C, \mathbf{\Psi})] - \text{E}[\log q(\mathbf{Z}, \mathbf{C}, \mathbf{\Psi})]. \tag{2}$$

We consider the following prior distributions for the parameters in $\mathbf{\Psi}$.

- $\mathbf{a}_i^k = (a_{i1}^k, \ldots, a_{iJ}^k) \sim \text{Dirichlet}(\mathbf{\Lambda}_i^{k0})$.
- $\theta_n \sim \text{Gamma}(\epsilon_{sn}^0, \epsilon_{rn}^0)$, with ϵ_{sn}^0, shape, and ϵ_{rn}^0, rate, hyperparameters.
- $\boldsymbol{\pi} \sim \text{Dirichlet}(\alpha_1^0, \ldots, \alpha_K^0)$

The main steps of our VI approach, for inferring $\mathbf{Z}, \mathbf{C}$, and $\mathbf{\Psi}$, are described below.

Step 1. VD factorization We assume the following factorization of the VD:

$$q(\mathbf{Z}, \mathbf{C}, \mathbf{\Psi}) = q(\boldsymbol{\pi}) \prod_{n=1}^{N} q(Z_n) \prod_{k=1}^{K} q(\mathbf{C}_k) \prod_{k=1}^{K} \prod_{i=1}^{J} q(\mathbf{a}_i^k) \prod_{n=1}^{N} q(\theta_n) \ . \tag{3}$$

Step 2. Joint distribution

The logarithm of the joint distribution satisfies: $\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \mathbf{\Psi}) = \log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \mathbf{\Psi}) + \log P(\mathbf{C}|\mathbf{\Psi}) + \log P(\mathbf{Z}|\mathbf{\Psi}) + \log P(\mathbf{\Psi})$. For details of the calculations see Appendix A.1.

Step 3. VD computation

We now derive a coordinate ascent algorithm for the VD. That is, we derive an update equation for each term in the factorization, Eq. (3), by calculating the expectation of $\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \mathbf{\Psi})$ over the VD of all random variables except the one currently being updated Bishop (2006). For example, we obtain the update equation for $\boldsymbol{\pi}$, $q(\boldsymbol{\pi})$, by calculating $E_{-\boldsymbol{\pi}}(\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \mathbf{\Psi}))$, where $-\boldsymbol{\pi}$ indicates that the expectation is taken with respect to the VD of all other random variables than $\boldsymbol{\pi}$, i.e., $\mathbf{Z}, \mathbf{C}$ and $\{\mathbf{A}, \boldsymbol{\theta}\}$ except $\boldsymbol{\pi}$. See A.2 and A.3 for the update equation of each term in Eq. (3).

Step 4. Summary of updates

We update the parameters of each posterior distribution presented in Step 3; for the summary of the update equations, see Alg. 2, in Appendix A.2. A shorter version of Alg. 2, providing an overview of CopyMix, is shown in Alg. 1. As shown in lines 8 and 9 of Alg. 1, we aim to find the posterior distributions of cluster assignments and copy number states. Note that the update equations in line 5 of Alg. 1 can be run in a parallel fashion, keeping the algorithm's computational complexity low. The implementation of CopyMix is available at **CopyMix**.

Algorithm 1 CopyMix: a Variational Inference Algorithm

1: **procedure** ESTIMATE-POSTERIORS(Y): ▷ Y input is N sequences of length M
2: Assign priors hyperparameters
3: Initialise variational distributions parameters
4: **repeat**
5: Update variational parameters (using **Step 4. Summary of updates**)
6: **until** convergence of the ELBO in Eq. 2
7: Output all posterior distributions including:
8: $q(Z_n = k) \quad \forall n, k;$ ▷ cluster assignment probabilities
9: $q(C_{km} = j) \quad \forall k, m, j;$ ▷ copy number probabilities
10: **return** posterior distribution

2.2 Gaussian emission

In this version, our input data, read count ratios (or read ratios), are assumed to be GC-corrected—a necessary bias correction due to the dropping of read coverage at the regions with extreme GC contents Yoon et al. (2009). The read counts are integers, and the GC-corrected read counts become positive real numbers (read ratios) through the GC-correction process. We consider read ratios over M fixed, typically equal-sized, genomic bins, as in Leung et al. (2017); Laks and McPherson (2019). We assume that the read ratios follow a Gaussian distribution; for the details supporting this choice, see Appendix A.10. Moreover, we presume that the read ratios are emitted from a latent sequence of copy number states. The probabilistic graphical model is illustrated in Fig. 3. The read ratios are modeled as a Gaussian

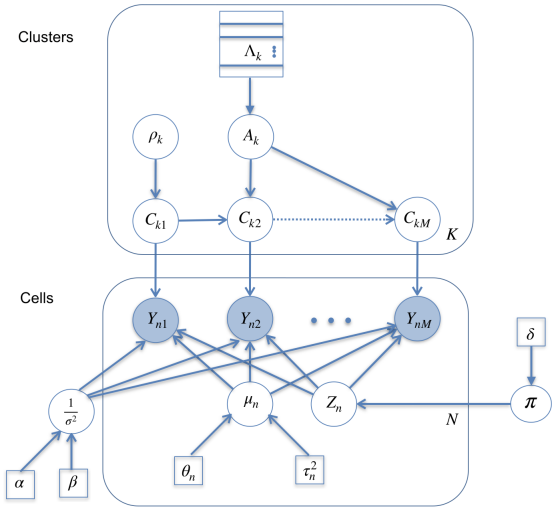

Figure 3: In this version Y_{nm} are observed corrected read ratios from cell n and bin m; C_{km}, corresponding latent copy number state forming a Markov chain; μ_n, cell-specific rate; Z_n, latent cell-specific cluster assignment variable. π and A_k, the priors over Z_n and C_{km} respectively; ρ_k, prior over the starting copy number state.

distribution with independent conjugate priors; that is, the mean of the Gaussian follows a Gaussian distribution and the precision of the Gaussian follows a Gamma distribution. Y denotes the observable variables, read ratios per predefined bins, C the latent copy number states forming a Markov chain, and Z the latent cell-specific cluster assignment variables. Note that in the Results section, we account for chromosomal copy numbers by assigning initial probabilities to the beginning of each chromosome.
Let Y_{nm} be the observed gc-corrected reads ratio to bin m for cell n for $n = 1, \ldots, N$ and $m = 1, \ldots, M$. Let $\mathbf{Y}_n = (Y_{n1}, \ldots, Y_{nM})$ be vector with all observed data for cell n. Given the cluster assignment and the corresponding true hidden vector of copy number states, $Y_{n1}, \ldots, Y_{nM}$ are independent with Y_{nm} following a distribution with parameters depending on the hidden true state at bin m for cluster population k, that is,

$$Y_{nm}|Z_n = k, C_{km} = j \sim F_{j\mu_n,\sigma^2}.$$

We assume $F_{j\mu_n,\sigma^2}$ to be a Gaussian distribution with $j\mu_n = j \times \mu_n$, i.e.,

$$F_{j\mu_n,\sigma^2}(Y_{nm}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(Y_{nm} - j\times\mu_n)^2}{2\sigma^2}}. \tag{4}$$

The use of the multiplicative structure in the mean of the Gaussian distribution can be found in a recent work in which a distribution mean is dependent on copy number events by a multiplication operation Malekpour et al. (2018). Note that one needs B-allele frequencies (BAF) to perform ploidy estimation, and the methods not using BAF data apply restrictive assumptions to select among many equally plausible solutions to estimated ploidy Zaccaria and Raphael (2021); this may result in selecting copy numbers that contradict the underlying allelic balance/imbalance. In the Concluding remarks, we refer to allele-based CopyMix as a future work.
Let $\mathbf{\Psi}$ be the set containing all the unknown model parameters, i.e., $\mathbf{\Psi} = \{\mathbf{A}, \boldsymbol{\mu}, \sigma^2, \boldsymbol{\pi}\}$, where

- $\mathbf{A} = \{\mathbf{A}_k \text{ for } k = 1, \ldots, K\}$ with $\mathbf{A}_k = \{a_{ij}^k \text{ for } i = 1, \ldots, J \text{ and } j = 1, \ldots, J\}$;
- $\boldsymbol{\mu} = \{\mu_n \text{ for } n = 1, \ldots, N\}$;
- σ^2;
- $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$.

We consider the following prior distributions for the parameters in $\boldsymbol{\Psi}$.

- $\mathbf{a}_i^k = (a_{i1}^k, \ldots, a_{iJ}^k) \sim \text{Dirichlet}(\boldsymbol{\Lambda})$
- $\mu_n \sim \text{Normal}(\theta_n, \tau_n^2)$. The conjugate prior concerning the mean of Normal distribution is Normal distribution.
- $\frac{1}{\sigma^2} \sim \text{Gamma}(\alpha, \beta)$. The conjugate prior concerning the precision of Normal distribution is Gamma distribution.
- $\boldsymbol{\pi} \sim \text{Dirichlet}(\boldsymbol{\delta})$

The main steps of our VI approach, for inferring $\mathbf{Z}, \mathbf{C}$, and $\boldsymbol{\Psi}$, are described below.

Step 1. VD factorization We assume the following factorization of the VD:

$$q(\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) = \Big[\prod_{n=1}^{N} q(Z_n)\Big]\Big[\prod_{k=1}^{K} q(\mathbf{C}_k)\Big]\Big[\prod_{k=1}^{K}\prod_{i=1}^{J} q(\mathbf{a}_i^k)\Big]\Big[\prod_{n=1}^{N} q(\mu_n)\Big] q(\frac{1}{\sigma^2}) q(\boldsymbol{\pi}). \tag{5}$$

Step 2. Joint distribution

The logarithm of the joint distribution satisfies: $\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) = \log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) + \log P(\mathbf{C}|\boldsymbol{\Psi}) + \log P(\mathbf{Z}|\boldsymbol{\Psi}) + \log P(\boldsymbol{\Psi})$. For details of the calculations see Appendix A.4.

Step 3. VD computation

We now derive a coordinate ascent algorithm for the VD; as we described this for the Poisson version, we derive an update equation for each term in the factorization, Eq. (5), by calculating the expectation of $\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi})$ over the VD of all random variables except the one currently being updated Bishop (2006). See A.5 and A.6 for the update equation of each term in Eq. (5).

Step 4. Summary of updates

We update the parameters of each posterior distribution presented in Step 3; for the details of the update equations, see Appendix A.5. We summarize the updates as shown in Alg. 1.

2.3 SNV-CopyMix

In this section, we incorporate SNV data into the model (see Fig. 4). We augment CopyMix graphical model by a copy-number-independent SNV process; we refer to this version of CopyMix as SNV-CopyMix. In Fig. 4, the new components are colored in magenta. X_{nl} denotes the observable variable, corresponding to a nucleotide in the genome, that we assume to be dependent on cluster-specific latent point mutation S_{kl}. Naturally, each S_{kl} is distributed as Bernoulli, where mutation corresponds to value of zero, and non-mutation corresponds to value of one. The prior distribution of S_{kl} is ξ, a Beta distribution. Note that the SNV data contains L sites such that $L >> M$. In the following subsection, the calculation of the probability of X_{nl} is described; for more detail see the work by Koptagel et al. (2018). The VI update equations are calculated analogously to those in CopyMix; see A.7.

2.3.1 The likelihood model for SNVs

Here, we briefly describe the computations related to the likelihood model for the SNVs, i.e., the probability of each X_{nl} is calculated. For more details, see the earlier work Koptagel et al. (2018). The genotype of a site l is denoted by $G_l = (g_l^1, g_l^2)$ where g_l^i is the nucleotide of allele i. The site genotype consists of two copies of the reference nucleotide in the case of no mutation. It consists of one reference (denoted as ref) and one alternate (denoted as alt) nucleotide in the case of a heterozygous SNV, and two alternates in the case of homozygous SNV. Note that the sites are assumed to be diploid. Let t be an arbitrary nucleotide in the read. Also, assume that r is an enumeration of the number of reads; e.g., if a cell has five reads at a site, then $r = 1, .., 5$. Moreover, we define ϵ to be a the error probability. It is computed by processing the Phred scores in data—Phred score Ewing et al. (1998) is a measure of the quality of the identification of the nucleotides generated by DNA sequencing. The calculation of genotype likelihood is similar to Monovar Zafar et al. (2014), but it is a generalized version of Monovar, where we explicitly distinguish allele info for $refref$ and $altalt$ cases. To this end, the likelihood of a single read of cell n at site l (X_{nl}^r) with the corresponding error probability (E_{nl}^r) is

$$p(X_{nl}^r = t | E_{nl}^r = \epsilon, G_l = (g_l^1, g_l^2)) = \sum_{i=1}^{2} \frac{1}{2} \times (1-\epsilon)^{I[t=g_l^i]} \times (\frac{\epsilon}{3})^{I[t \neq g_l^i]}.$$

The reads are assumed to be i.i.d., so the likelihood of the reads of cell n at site l is

$$p(X_{nl}^{1:|X_{nl}|} | E_{nl}^{1:|X_{nl}|}, G_l) = \prod_{r=1}^{|X_{nl}|} p(X_{nl}^r | E_{nl}^r, G_l).$$

The likelihood for a non-mutation case is $p(X_{nl}^{1:|X_{nl}|} | E_{nl}^{1:|X_{nl}|}, G_l = (ref, ref))$. The likelihood for a heterozygous mutation case is $p(X_{nl}^{1:|X_{nl}|} | E_{nl}^{1:|X_{nl}|}, G_l = (ref, alt))$, and a homozygous mutation case is $p(X_{nl}^{1:|X_{nl}|} | E_{nl}^{1:|X_{nl}|}, G_l = (alt, alt))$. Note we allow for two versions of SNV-CopyMix, that is, (1) likelihood conditioned on non-mutation versus homozygous mutation, and (2) likelihood conditioned on non-mutation versus heterozygous mutation. In the experiment section, we test both of these versions when referring to SNV-CopyMix.

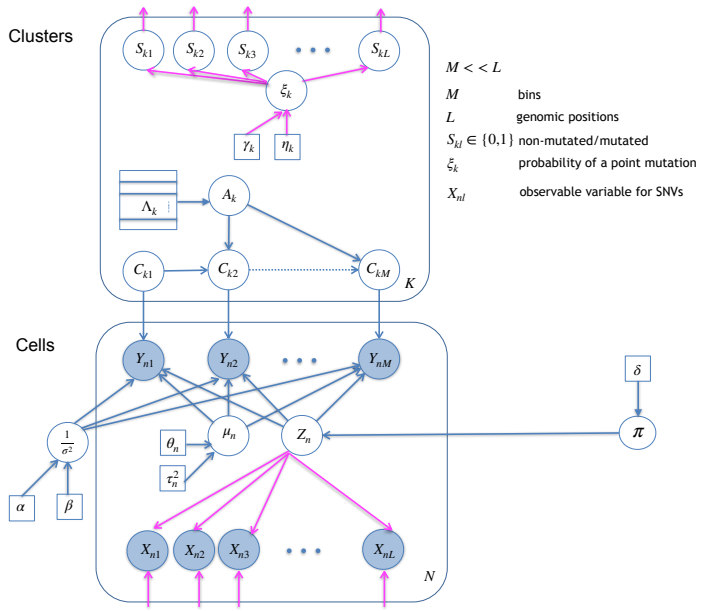

Figure 4: The graphical model with SNV process included.

3 Empirical investigation

3.1 Experimental setup

3.1.1 Dataset

To adequately evaluate CopyMix, as opposed to earlier literature lacking the evaluation against ground truth and hence subjective, we conduct experiments on simulated data, where we have access to the ground truth. We also evaluate CopyMix on two biological datasets Laks and McPherson (2019); Leung et al. (2017): (1) Read count data from Leung et al. (2017) (available from the NCBI Sequence Read Archive (SRA) under accession number SRP074289); (2) Read ratio data from Laks and McPherson (2019) accessed via DLP.

The first biological data stem from single-cell whole-genome sequencing from 18 primary colon cancer tumor cells and 18 metastatic cells from matched liver samples for one patient referred to as CRC2 in Leung et al. (2017). In this work, we mainly analyze CRC2 patient data (more complex than CRC1 according to Leung et al. (2017)) considering chromosomes 18, 19, 20, and 21 comprising a total of 904 genomic bins of an approximate size of 200Kb each. Each data point corresponds to the number of reads aligned per bin per cell after GC correction Leung et al. (2017). These counts stemming from the tumor cell's genome reflect the hidden copy numbers—copy number integers are conventionally ranged from 0 to 6, as in Leung et al. (2017). The higher the counts are, the higher the copy numbers are expected to be. Regarding clustering in Leung et al. (2017), it is reasonable that the primary tumor cells from the colon and the metastatic cells from the liver should cluster separately, as inter-tumor copy number differences are expected to dominate over minor intra-tumor copy number variations; therefore, we use the terms primary and metastatic clusters.

The second biological data stems from single-cell whole-genome sequencing using DLP technology. The data is from three cell lines from ovarian cancer with a total number of 891 cells Laks and McPherson (2019). It is meaningful to infer that each cell line is a separate cluster; however, the goal is to find sub-clusters as well.

For the simulated data, we examine both Poisson and Gaussian models. For the Poisson version, we randomly generate 30 datasets for each test case, following our graphical model and inspired by biological datasets from Leung et al. (2017); for data simulation details, see A.8. We examine different test cases, orthogonal to each other, to show our approach's robustness; each scenario, for both clustering and copy number state inference, is reproducible. Regarding the Gaussian model, we similarly show the robustness of CopyMix by varying the number of clusters and clustering scenarios. We simulate data under various clustering scenarios, varying the number of clusters and copy number profiles. We assume a diploid cell (normal cell) can be affected by various copy number events (comprising various duplications, deletions, whole genome doubling (WGD), and their combinations). We use an HMM, a generative model with latent copy number states, to generate the observed read ratios—HMM, as in HMMcopy Shah et al. (2006), is a well-known model for copy number profiling. Hence, we generate 18 configurations of simulated single-cell clustering (CONF 1 to CONF 18 shown in Fig. 5), where each color-coded cluster corresponds to a unique CNV pattern, i.e., state transition pattern in the Markov chain. We use 150 cells and 200 bins, where the sequence is divided into relative chromosomes, i.e., the proportions of the chromosome lengths in the sequence are the same as those in a human genome. Relative chromosomes account for the independence of copy number variations between chromosomes. For the details of the simulated data, see Appendix A.8.

3.1.2 Experimental Protocol

The VI framework allows for determining the number of clusters Bishop (2006); we run VI with a maximum number of clusters, and the framework results in zero probabilities for the extra clusters not detected by VI. We run VI using 50 different random initializations, as recommended by Blei et al. (2017). We choose the best VI run based on the highest ELBO.

For the Poisson version, we run VI for different numbers of clusters considering various random and informative initialization methods, e.g., k-means and Baum-Welch, as recommended by Blei et al. (2017). For each given number of clusters, we choose the best VI run based on the highest log-likelihood. Next, we select the number of clusters by an elbow selection method Johnson and Wichern (2007) based on the Deviance Information Criterion Spiegelhalter et al. (2002) over different numbers of clusters. We assess the clustering performance of CopyMix on 30 simulated datasets for each test case via V-measure Rosenberg and Hirschberg (2007). The V-measure, a value between 0 and 1, quantifies to which extent a cluster only contains data points that are members of a single class (the higher V-measure, the better clustering). Regarding inference of copy number profiles, we evaluate the performance of CopyMix on simulated data by calculating the proportion of discordant (unequal) position values between true and inferred vectors, as in de Souza et al. (2020), of copy number states. For biological data evaluation, V-measure is used to assess the clustering performance where the true clusters are assumed to correspond to the primary and metastatic cells; note that these may not be ground truth, but we believe there is certain concordance with the unknown ground truth. Regarding state inference, we compare the copy number profiles between the results from CopyMix and the original paper Leung et al. (2017).

For the Gaussian version, we compare the clustering results of CopyMix with the ground truth using

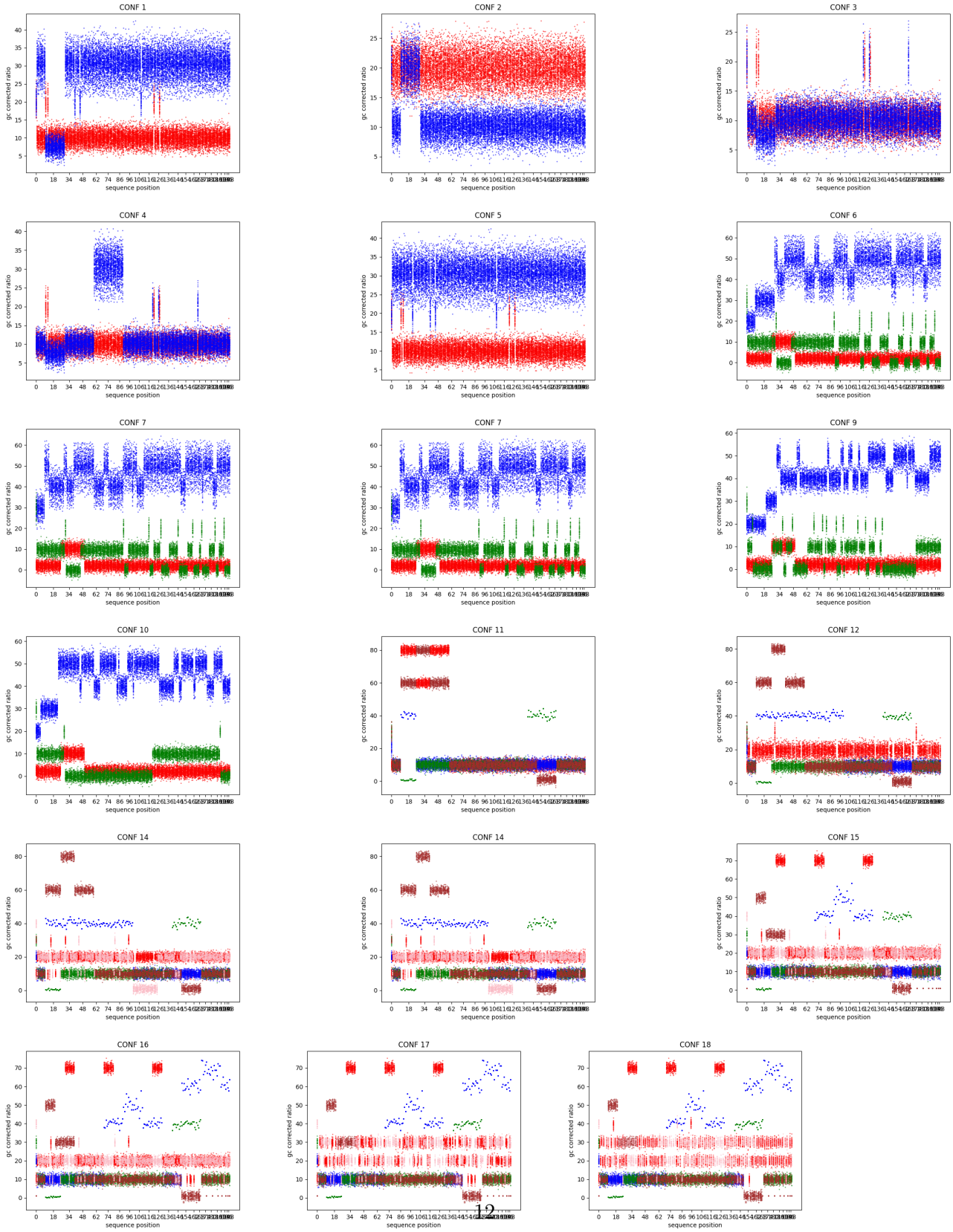

Figure 5: The datasets for CONF 1 to 18 are shown. CONF 1 to 5, CONF 6 to 10, CONF 11 to 13, and CONF 14 to 18 correspond to scenarios where 2, 3, 4, and 5 clusters are color-coded, respectively.

V-measure for simulated data. For biological data evaluation, V-measure is used to compare our inferred clusters to the ones reported in the original paper Laks and McPherson (2019). In addition to merely comparing to the clustering result in the original paper, we examine if the report in the original article reveals meaningful clusters; we obtain this by calculating the CH Index Calinski and Harabasz (1974) and Silhouette Rousseeuw (1987) metrics on the raw data. Next, we use the likelihood ratio test to examine if the null hypothesis (CopyMix) hypothesis is rejected or not. Considering also SNV data, which are independent from CNVs, we run the SNV-CopyMix on the DLP data. Finally, we run GinkgoGarvin et al. (2015)—a state-of-art framework in copy number based clustering of single-cells by using circular binary segmentation method for copy number profiling—on the DLP data. For evaluating copy number profiles, we calculate the distance between the inferred and true copy number using total variation.

CH Index is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters; the higher the score, the better the performance. Silhouette is calculated using two scores $a, \quad b$, as shown below. Note that a is the mean distance between a sample and all other points in the same class. This score measures the closeness of points in the same cluster. And, b is the mean distance between a sample and all other points in the next nearest cluster. This score measures the distance of points of different clusters.

$$s = \frac{b - a}{max(a, b)} \tag{6}$$

The s score is bounded between -1 for incorrect clustering and +1 for highly dense clustering. Scores around zero indicate overlapping clusters. The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster.

Finally, the likelihood ratio test assesses the goodness of fit of two competing statistical models based on the ratio of their likelihoods. In our experiments, we examine how statically significant the winning model is; by the winning model, we mean the one with the higher likelihood. The significance is measured by p-value; a high p-value indicates that the null hypothesis is preferred over the winning hypothesis.

3.2 Experimental Results for Poisson Version

We perform experiments on simulated data and biological data, and we evaluate a sequential version of CopyMix to verify the superiority of single-phase inference over the sequential framework.

Simulated data Here, we evaluate CopyMix, and compare it to the initialization methods used by VI on 30 simulated datasets for each test scenario. The comparison is to illustrate, firstly, that VI results are not already achieved by the initialization methods, and secondly, that VI can recover the poor initialization performance and achieve optimal performance. We select the number of cells, sequence lengths, rate parameters, transition probabilities based on Leung et al. (2017). We assume equal amplification on the read counts across bins since the bin size is high enough (200kb as in Leung et al. (2017)), meaning that the effect of unequal amplification is negligible. Figures 6, 7 and 8 provide summaries of V-measures and proportions of discordant positions; note that the median values in the boxplots, highlighted in red for initialisation method and in blue for CopyMix, aim to facilitate the comparisons. The figures illustrate the performance comparison of CopyMix and the chosen initialization method, by CopyMix, for different settings: copy number transition matrices and number of clusters (Fig. 3); sequence length (Fig.4); and number of cells (Fig. 5). For all the tests, CopyMix results in V-measure of 1 except for one case (Fig. 6) where V-measure is in an interval of [.85, 1] with the median being 1; proportions of discordant positions

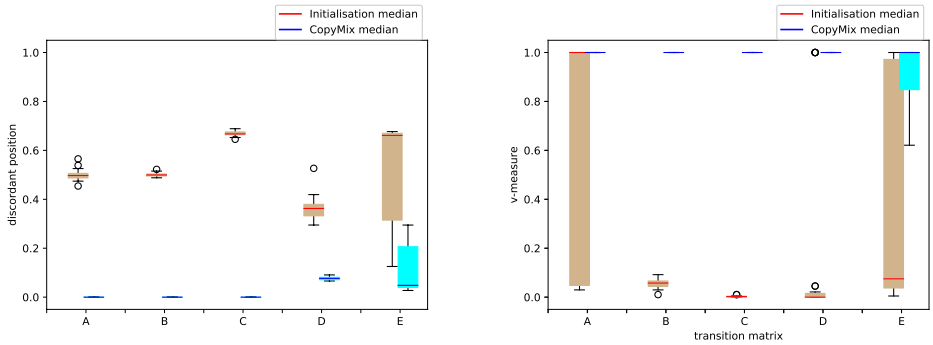

Figure 6: Performance of initialization methods (brown bars with red median values) and CopyMix (cyan bars with blue median values) across different sets of transition matrices. LHS: discordant positions between true and inferred vectors of copy number states. RHS: V-measure between true and inferred cell clustering assignments.

are mostly 0 and otherwise are in the interval of [0, .2]. These results are optimal as the copy number profiling errors are near 0 and clustering near 1 (or 100%).

The results in Fig. 6 are obtained considering 45 cells—each with a sequence length of 800 bins and the mean value around 100—clustered into two and three clusters, wherein each clustering scenario, each cluster bears a different transition pattern than other clusters. The copy number transition patterns (forming the transition matrix of a Markov chain) vary across configurations A to E, which are orthogonal to each other; Configurations A to E are formed by combining the following state patterns.

- **single state**, a copy number state is inclined to stay at one certain state;
- **self-transition state**, a copy number state is inclined to stay at its current state for awhile;
- **oscillating state**, a copy number state changes shortly after being at a state;
- **spike state**, a sudden change of copy number state arises for a specific short range of positions in the copy number sequence;
- **scaled state**, a copy number changes by a multiplicative or additive constant.

The configurations A to E are: **A**, similar to chromosomes 11 in CRC1 and 4 in CRC2, (one cluster follows (**scaled state**); **B**, similar to chromosome 11 in CRC1 (one cluster follows **self-transition state** and the other one **single state**); **C**, similar to chromosome 3 in CRC1 (two clusters have identical transition matrices but one arrive at **spike state**); **D**, similar to chromosomes 18 to 21 in CRC2 (clusters combine **self-transition state**, **oscillating state**, and **scaled state**); **E**, two of the clusters have the transitions mentioned in case **C**, and, the third cluster has a combination of **oscillating state** and **single state**.

We can observe that, in Fig. 6, CopyMix has a perfect performance in terms of clustering and copy number inference for easier cases and also a high performance concerning the harder cases (**D** and **E**). Next, however not shown here, we investigate decreasing the rate parameter, θ_n, for configuration **A** to see how our model handles shallow data as this can be a challenge in copy number profiling Zahn

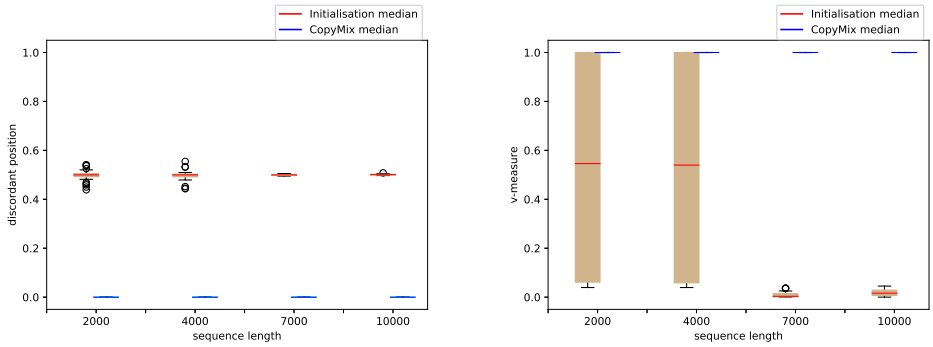

Figure 7: Performance of initialization methods and CopyMix, varying the sequence length. LHS and RHS figures, as well as color codes, are similar to Fig. 6.

et al. (2017). Our model is robust towards lower rates up to $\theta_n = 3$, without reducing the quality of performance concerning V-measure and discordant positions. As the third set of tests, we rerun the test using configuration **A**, but we increase the size of the sequences up to 10000 (inspired by the total length of the genomic bins from biological data in Leung et al. (2017)), see Fig. 7. We can observe that our model handles these sequences without any decrease in performance. Finally, Fig. 8 illustrates the perfect performance of CopyMix when varying the number of cells from 15 to 500 when, again, using configuration **A**.

Biological data We evaluate CopyMix on the biological data from Leung et al. (2017). As aforementioned, we do not compare CopyMix to the earlier state-of-the-art methods due to their inclusion of ad-hoc processes in their framework; however, to evaluate the CopyMix results on the biological data, we use biologically verifiable results from the method in Leung et al. (2017). Using chromosomes 18 to 21, CopyMix clusters the cells into primary and metastatic groups, perfectly matching the corresponding primary and metastatic organs—V-measure is 1 w.r.t. the cells found in the two organs Leung et al. (2017). Adding chromosomes 3 and 8 to our test, metastatic tumor cells are clustered into two major subclusters as in Leung et al. (2017). As mentioned in Leung et al. (2017), amplifications on chromosomes 3 and 8, where some known genes are present, contribute to the detection of the metastatic subclusters. Note that, Leung et al. (2017) does not present a method to determine the threshold in hierarchical clustering, which results in the reported subclusters; moreover, there is no ground truth to verify these results.

Regarding the copy number profiling, Fig. 9 illustrates the copy number profiles inferred by CopyMix and the original work by Leung et al. (2017), here referred to as Navin. The heatmaps indicate a higher copy number state by a darker color. Note that in Navin, they perform ploidy scaling, which contributes to scaling differences compared to the results of CopyMix. Moreover, the Navin heatmap, here, illustrates an average of the heatmap in the original work across the cells since there is no single copy number profile per cluster. That is, Navin, unlike CopyMix, lacks a single copy number profile per cluster, revealing the copy number variation pattern behind each cluster. CopyMix detects the breakpoints (copy number changes) similar to Navin, with an accuracy of 76%, using the breakpoint margin (of a size corresponding to 5% of the sequence length). Again, there is no ground truth to verify the Navin results from Leung et al. (2017).

A sequential CopyMix Applying a sequential version of CopyMix (1. performing copy number profiling,

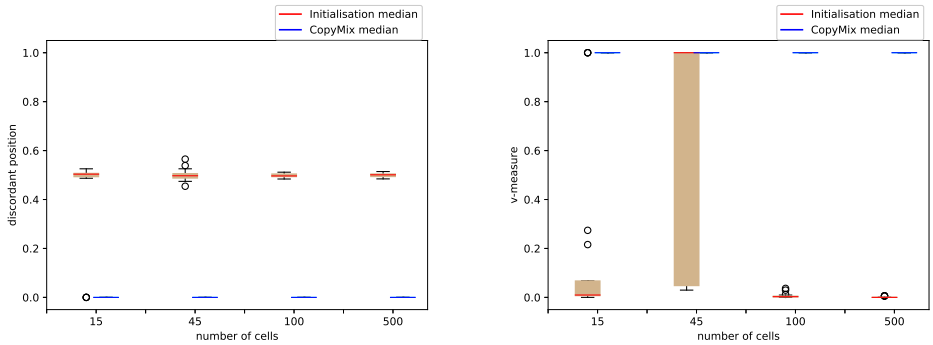

Figure 8: Performance of initialization methods and CopyMix varying the number of cells. LHS and RHS figures, as well as color codes, are similar to Fig. 6.

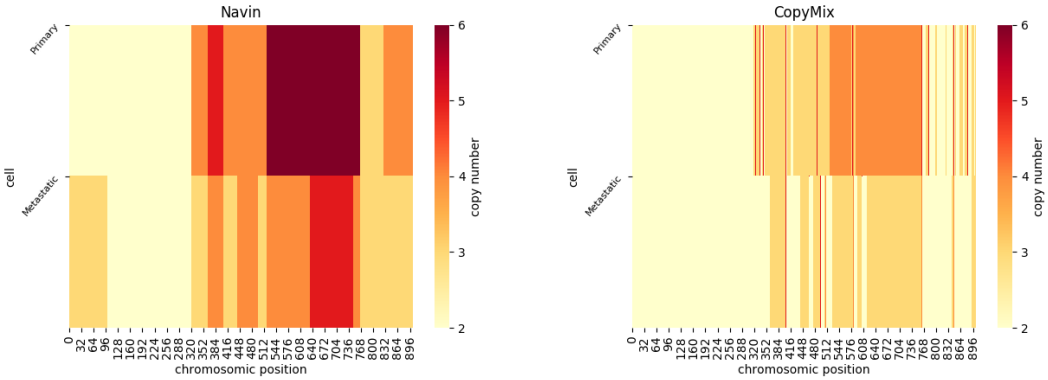

Figure 9: Copy number profiling comparison between the results from Navin Leung et al. (2017) (*LHS*) and CopyMix (*RHS*) for both primary and metastatic clusters.

using EM for HMMs; 2. performing clustering over the inferred copy numbers, using EM for a mixture model), resembling Shah et al. (2006); Vitak et al. (2017) but excluding their ad-hoc processes, on the biological data, results in poor clustering results (V-measure of .24) while, as mentioned, CopyMix gains V-measure of 1. This result confirms that simultaneous detection of copy number profiles and clustering is superior to the sequential approach. In particular, we conclude that methods that use EM for HMMs, using HMMcopy Shah et al. (2006) framework, benefit from the multiple ad-hoc transformations within the HMMcopy framework, much more than the pure computational model, e.g., HMM. Adding multiple ad-hoc rules within such computational models lacks elegance, integrity, and potential model-based approach benefits.

3.3 Experimental Results for Gaussian Version

We perform experiments on simulated data and biological data. For the biological data we evaluate CopyMix against the original paper, referring to as DLP, using CH Index, Silhouette, and likelihood ratio test.

Simulated data We evaluated the performance of CopyMix on 16 simulated dataset configurations; see Fig. 5. In Table 1, we report V-measure for each configuration. We can observe that CopyMix, except for CONF 5, has a good clustering performance with V-measures ranging from 87% to 100%, Table 1. The exception, i.e., CONF 5, consists of two simulated genomes where one is a whole-genome duplication (WGD) of the other, *a whole-genome duplication pair*. Typically, when a genome undergoes a whole-genome duplication, the copy numbers across the sequence are scaled to a larger value, i.e., usually, diploid becomes tetraploid. This issue of CopyMix in distinguishing the ploidy can be explained partially by the notion of *unidentifiability*—inability to derive the true labels in unsupervised learning—of the transition matrix in a Markov chain Murphy (2012).

Another important observation from Table 1 is that Copymix is robust w.r.t. increasing the number of clusters and the complexity of copy number patterns. We, however, notice a possible limitation of our purposeful modeling approach employed in CopyMix, i.e., each bin for each cell has a mean equal to the product of the cell-specific rate and the underlying latent copy number at that bin. Namely, despite providing strength in differentiating clusters, the multiplicative structure of copy number state and cell-specific rate can produce the same means for different clusters, e.g., $2 \times 4 = 1 \times 8$, where 2 and 1 are the copy numbers of two clusters, and 4 and 8 are two cell rates corresponding to those two clusters, respectively; therefore, the higher cell rates may not be assigned to the higher copy number states because of this unidentifiability issue. The other source of clustering error is when the clusters have too overlapping sequence patterns, i.e., copy number profiles (CONF 14 and CONF 15 in Fig. 5). The cluster-overlapping in CONF 16 to CONF 18 is less than in CONF 15, hence perfect performance. As it may be hard to detect, we highlight that CONF 17 and CONF 18 differ in CNV in the last third part of the genome concerning clusters red and pink.

Finally, total variations between the true copy number and those inferred by CopyMix are shown in Table 1. CopyMix reveals small errors (distances to the ground truth copy numbers). However, we observe that slightly larger errors begin to manifest by increasing the number of clusters. Also, we notice that the error slightly increases when there are many breakpoints in the copy number profiles (the blue-colored genomes in CONF 6 to 10). This is because the possibility of differing copy numbers increases as copy number breakpoints increase.

Biological data

Table 1: **V-measure (clustering result %) and total variation (copy number inference result 0.0) for CONF 1 to CONF 18.**

CONF 1	**CONF 2**	**CONF 3**	**CONF 4**	**CONF 5**	**CONF 6**
100% — 0.09	100% — 0.05	100% — 0.05	100% — 0.20	0% — 0.11	100% — 2.45
CONF 7	**CONF 8**	**CONF 9**	**CONF 10**	**CONF 11**	**CONF 12**
100% — 2.45	100% — 2.45	100% — 2.36	100% — 2.45	87% — 1.02	100% — 0.96
CONF 13	**CONF 14**	**CONF 15**	**CONF 16**	**CONF 17**	**CONF 18**
100% — 0.93	88% — 1.20	87% — 0.96	100% — 1.03	100% — 1.04	100% — 1

We evaluated CopyMix on the DLP data Laks and McPherson (2019), where each genome is partitioned into 6206 bins (bin size of 500K) [3]. The clusters should agree with cell lines; no cluster contains cells from two different cell lines. It turns out that CopyMix succeeds in clustering the main clusters since the V-measure when comparing CopyMix clusters with the cell lines is 98%. The V-measure of overall CopyMix clusters based on those reported by the original clusters is 67%.

Using the LR test, the number of clusters, nine, reported by the original paper is statistically non-significant, i.e., CopyMix as a null hypothesis is favored over the method in the original paper. Assuming that each cluster follows an empirical Gaussian distribution—the mean and variance can be calculated by maximum likelihood estimate, nine clusters result in a log-likelihood of -8980284.42, and CopyMix log-likelihood is -9023905.94. The test results in p-value one, i.e., the original likelihood, is negligibly higher than that obtained by CopyMix.

Table 2 shows that CopyMix outperforms the original paper w.r.t. clustering according to CH Index and Silhouette metrics. We also evaluated Silhouette for fewer clusters reported by the original paper; these clusters are obtained according to the reported phylogenetic tree in the original paper Laks and McPherson (2019). Regarding Silhouette, even though the CopyMix score, 0.49, is not the perfect score, it is far better than those obtained by the method in the original paper. These metrics show that the method in the original paper has detected overlapping clusters, which is not sufficiently supported by the data. The SNV-CopyMix did not detect more clusters due to shallow data; for the details, see Appendix A.11. Note that the clusters reported by the original paper are not ground truth; hence, we cannot conclude that any diverging result is poor.

Finally, we compared CopyMix's clustering results with those of Ginkgo. Unlike CopyMix, Ginkgo applies multiple ad-hoc processes, such as outlier removal from cells and bins and various corrections to the processed data. After removing 80% of the cells (717 out of 891 cells), Ginkgo detects three major clusters using hierarchical clustering (V-measure of 98% like CopyMix). CopyMix is more powerful than Ginkgo because CopyMix does not need to remove 80% of the data to obtain the V-measure of 98%. For more details, see Appendix A.9.

Regarding the copy number profiling, we obtained a total variation of 91 when comparing the copy number reported by CopyMix and the original paper. We can calculate the ratio of this total variation over the sequence length, $\frac{91}{6206} = 0.014$), indicating a low error or relative total variation w.r.t. the sequence length (6206 bins of the genome). Since Ginkgo does not report the copy number profiles more

[3]CopyMix performance improves when the sequence is longer (the smaller the bin size, the longer the sequence) due to increasing the signal for the Markov chain. However, the low coverage in the DLP data increases the noise and missing data.

Table 2: **CH Index and Silhouette for different numbers of clusters.**

Method	**# of clusters**	**CH index**
Original Paper	9	462
Cell lines from Original Paper	3	623
CopyMix	4	**1082**

Method	**# of clusters**	**Silhouette**	**Description**
Original Paper	9	0.06	Overlapping clusters
Original Paper w/o 2 minor clones	7	0.21	
Original Paper w/o 4 minor clones	5	0.25	
CopyMix	4	**0.49**	Separated clusters

than visually (see Appendix A.9), it is impossible to calculate its total variation.

4 Concluding remarks

We introduced CopyMix, a novel mixture model to jointly perform single-cell copy number profiling and clustering. CopyMix, while enjoying the advantages of Bayesian inference, addresses the following issues: (1) the sequential treatment of copy number profiling and single-cell clustering, prone to introduce clustering artifacts; (2) the labor of HMMcopy parameter tuning; (3) the lack of clonal copy number profiling by MHMMs; (4) the complicated posterior approximation when deriving VI for HMMs.

We evaluated our approach on both simulated and biological data, which indicated that CopyMix performs well when estimating both single-cell clustering and the corresponding copy number profiles.

CopyMix can be extended for phylogenetic inference, improving clustering, and augmenting and refining the model. Designing an allele-specific copy number based model can enrich the DLP results even more, and, finally, modeling biallelic deletions and aneuploidy, inspired by the recently proposed CopyKat Gao et al. (2021), is also a biologically desirable task.

Author contributions statement

N.S. partially designed CopyMix, partially developed the mathematics, fully implemented CopyMix, fully wrote the manuscript, partially selected the evaluations metrics, and fully conducted experiments on CopyMix. C.P.E.S. partially designed CopyMix, partially developed the mathematics, helped with the design and evaluation of the study, partially wrote and reviewed the manuscript. A.R. provided useful suggestions for the evaluation of CopyMix and was involved in the model discussions. H.K. provided SNV emissions for the SNV-CopyMix and provided the main text for the section "The likelihood model for SNVs". H.T. ran Ginkgo on the biological DLP data. J.L. fully supervised and reviewed the paper, was involved in the entire project, and partially wrote the manuscript.

References

Baslan, T., Kendall, J., Rodgers, L., Cox, H., Riggs, M., Stepansky, A., ., . ., Navin, N., and Hicks, J. (2012), "Genome-wide copy number analysis of single cells," *Nature Protocols*, 7, 1024–10241.

Bishop, C. (2006), "Pattern recognition and machine learning," *Information science and statistics, New York, NY: Springer*.

Blei, D., Kucukelbir, A., and Mcauliffe, J. (2017), "Variational Inference: A Review for Statisticians," *Journal of the American Statistical Association*, 112, 859–877.

Blocker, A. W. and Meng, X.-L. (2013), "The potential and perils of preprocessing: Building new foundations," *Bernoulli*, 19, 1176–1211.

Calinski, T. and Harabasz, J. (1974), "A dendrite method for cluster analysis," *Communications in Statistics*, 1–27.

de Souza, C. P., Andronescu, M., Masud, T., Kabeer, F., Biele, J., Laks, E., Lai, D., Brimhall, J., Wang, B., Su, E., et al. (2020), "Epiclomal: probabilistic clustering of sparse single-cell DNA methylation data," *PLOS Computational Biology*.

Eirew, P., Steif, A., Khattra, J., Ha, G., Yap, D., Farahani, H., Gelmon, K., Chia, S., Mar, C., Wan, A., Laks, E., Biele, J., Shumansky, K., Rosner, J., McPherson, A., Nielsen, C., Roth, A., Lefebvre, C., Bashashati, A., de Souza, C., Siu, C., Aniba, R., et al. (2015), "Dynamics of genomic clones in breast cancer patient xenografts at single cell resolution," *Nature*, 518, 422.

et al, G. (2008), "Bayesian Hidden Markov Modeling of Array CGH Data," *J Am Stat Assoc*, 109.

Ewing, B., Hillier, L., Wendl, M., and Green, P. (1998), "Base-calling of automated sequencer traces using phred. I. Accuracy assessment," *Genome Research*, 8, 175—-185.

Gao, R., Bai, S., Henderson, Y. C., ..., and Navin, N. E. (2021), "Delineating copy number and clonal substructure in human tumors from single-cell transcriptomes," *Nat Biotechnology*.

Garvin, T., Aboukhalil, R., and Kendall, J. e. a. (2015), "Interactive analysis and assessment of single-cell copy-number variations," *Nat Methods*, 1058–11060.

Gawad, C., Koh, W., and Quake, S. R. (2016), "Single-cell genome sequencing: current state of the science," *Nature Reviews Genetics*, 17, 175.

Guiasu, S. and Shenitzer, A. (1985), "The principle of maximum entropy". The Mathematical Intelligencer," 7.

Johnson, R. A. and Wichern, D. W. (2007), *Applied multivariate statistical analysis, 6th Ed.*, Pearson Prentice Hall.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999), "An introduction to variational methods for graphical models," *Machine learning*, 37, 183–233.

Kapourani, C.-A. and Sanguinetti, G. (2019), "Melissa: Bayesian clustering and imputation of single-cell methylomes," *Genome biology*, 20, 61.

Koptagel, H., Jun, S., and Lagergren, J. (2018), "SCuPhr: A Probabilistic Framework for Cell Lineage Tree Reconstruction," *https://www.biorxiv.org/content/10.1101/357442v1*.

Laks, E. and McPherson, A. e. a. (2019), "Clonal Decomposition and DNA Replication States Defined by Scaled Single-Cell Genome Sequencing," *Cell*.

Lawson, D., Kessenbrock, K., Davis, R., Pervolarakis, N., and Werb, Z. (2018), "Tumour heterogeneity and metastasis at single-cell resolution," *Nature Cell Biology*, 20, 1349–1360.

Leung, M., Davis, A., Gao, R., Casasent, A., Wang, Y., Sei, E., ., . ., and Navin, N. (2017), "Single-cell DNA sequencing reveals a late-dissemination model in metastatic colorectal cancer," *Genome Research*, 27, 1287–1299.

MacKay, D. J. (1997), "Ensemble Learning for Hidden Markov Models," Tech. rep.

Malekpour, S., Pezeshk, H., and Sadeghi, M. (2018), "MSeq-CNV: accurate detection of Copy Number Variation from Sequencing of Multiple samples," *Nature Scientific Reports*.

Mallory, X., Edrisi, M., Navin, N., and Nakhleh, L. (2020a), "Assessing the performance of methods for copy number aberration detection from single-cell DNA sequencing data," *PLOS Computational Biology*.

— (2020b), "Methods for copy number aberration detection from single-cell DNA-sequencing data," *Genome Biology*.

Markowska, M., Cakała, T., Miasojedow, B., and et al (2022), "CONET: copy number event tree model of evolutionary tumor history for single-cell data," *Genome Biology*.

McGrory, C. A. and Titterington, D. (2009a), "Variational Bayesian analysis for hidden Markov models," *Australian & New Zealand Journal of Statistics*, 51, 227–244.

McGrory, C. A. and Titterington, D. M. (2009b), "VARIATIONAL BAYESIAN ANALYSIS FOR HIDDEN MARKOV MODELS," *Australian and New Zealand Journal of Statistics*.

Murphy, K. P. (2012), "Machine Learning: A Probabilistic Perspective," .

Navin, N., Kendall, J., Troge, J., Andrews, P., Rodgers, L., McIndoo, J., Cook, K., Stepansky, A., Levy, D., Esposito, D., et al. (2011), "Tumour evolution inferred by single-cell sequencing," *Nature*, 472, 90.

Nowell, P. C. (1976), "The clonal evolution of tumor cell populations," *Science*, 194, 23–28.

Olshen, A. B., Venkatraman, E. S., Lucito, R., and Wigler, M. (2004), "Circular binary segmentation for the analysis of array-based dna copy number data," *Biostatistics*.

Rosenberg, A. and Hirschberg, J. (2007), "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*.

Roth, A., McPherson, A., Laks, E., Biele, J., Yap, D., Wan, A., Smith, M. A., Nielsen, C. B., McAlpine, J. N., Aparicio, S., et al. (2016), "Clonal genotype and population structure inference from single-cell tumor sequencing," *Nature methods*, 13, 573.

Rousseeuw, P. (1987), "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Comput. Appl. Math.*, 20, 53–65.

Shah, S. P., Xuan, X., DeLeeuw, R. J., Khojasteh, M., Lam, W. L., Ng, R., and Murphy, K. P. (2006), "Integrating copy number polymorphisms into array CGH analysis using a robust HMM," *Bioinformatics*, 22, e431–e439.

Shapiro, E., Biezuner, T., and Linnarsson, S. (2013), "Single-cell sequencing-based technologies will revolutionize whole-organism science," *Nature Reviews Genetics*, 14, 618.

Smyth, P. (1997), "Clustering sequences with hidden markov models," *Advances in Neural Information Processing Systems*.

Spiegelhalter, D., Best, N., Carlin, B., and Van Der Linde, A. (2002), "Bayesian measures of model complexity and fit," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64, 583–639.

Vitak, S., Torkenczy, K., Rosenkrantz, J., Fields, A., Christiansen, L., ..., and Adey, A. (2017), "Sequencing thousands of single-cell genomes with combinatorial indexing," *Nature methods*, 14.

Witten, D. (2011), "Classification and clustering of sequencing data using a Poisson model," *Ann. Appl. Stat.*, 5, 2493–2518.

Yoon, S., Xuan, Z., Makarov, V., Ye, K., and Sebat, J. (2009), "Sensitive and accurate detection of copy number variants using read depth of coverage," *Genome Research*.

Zaccaria, S. and Raphael, B. (2021), "Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL," *Nat Biotechnol*.

Zafar, H., Wang, Y., Nakhleh, L., Navin, N., and Chen, K. (2014), "Monovar: single-nucleotide variant detection in single cells," *Nature Methods*.

Zahn, H., Steif, A., Laks, E., Eirew, P., VanInsberghe, M., Shah, S. P., Aparicio, S., and Hansen, C. L. (2017), "Scalable whole-genome single-cell library preparation without preamplification," *Nature methods*, 14, 167.

Zhang, A. and Campbell, K. (2020), "Computational modelling in single-cell cancer genomics: methods and future directions," *Phys. Biol.*

Zuo, C., Chen, K., Hewitt, K., Bresnick, E., and Keleş, S. (2016), "A hierarchical framework for state-space matrix inference and clustering," *Ann. Appl. Stat.*, 10, 1348–1372.

A Appendix: Derivation details

A.1 Step 2 of Section 2.1

In this section, we provide the calculation of each decomposed term of $\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \mathbf{\Psi})$. Each term of $\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \mathbf{\Psi})$ presented in Step 2 of Section 2 is calculated as follows: $\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \mathbf{\Psi}) =$
$\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{I}(Z_n = k)\mathrm{I}(C_{km} = j) \log F_{j\theta_n}(y_{nm}) =$
$\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{I}(Z_n = k)\mathrm{I}(C_{km} = j)\Big[y_{nm}\log(j\theta_n) - j\theta_n - \log(y_{nm}!))\Big]$;
$\log P(\mathbf{C}|\mathbf{\Psi}) = \sum_{k=1}^{K}\big[\sum_{j=1}^{J}\mathrm{I}(C_{k1} = j)\log\rho_{kj} + \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{I}(C_{km-1} = i, C_{km} = j)\log a_{ij}^{k}\big]$;

$$\begin{aligned}
\log P(\mathbf{Z}|\mathbf{\Psi}) &= \sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{I}(Z_n = k)\log\pi_k; \\
\log P(\mathbf{\Psi}) &= \log P(\mathbf{A}) + \log P(\boldsymbol{\theta}) + \log P(\boldsymbol{\pi}); \qquad (7)
\end{aligned}$$

where,

$$\log P(\boldsymbol{\theta}) = \sum_{n=1}^{N}\big[(\epsilon_{sn}^{0} - 1)\log\theta_n - \epsilon_{rn}^{0}\theta_n\big] + C,$$

$$\log P(\mathbf{A}) = \sum_{k=1}^{K}\sum_{i=1}^{J}\big[\sum_{j=1}^{J}(\Lambda_{ij}^{k0} - 1)\log a_{ij}^{k}\big] + C,$$

and

$$\log P(\boldsymbol{\pi}) = \sum_{k=1}^{K}(\alpha_k^0 - 1)\log\pi_k + C.$$

A.2 Step 3 and 4 of Section 2.1

In this section we first present the details of the step 3 of section 2, i.e., the derivation of each posterior update equation (VD computation). We then summarise them in the form of a pseudo code, covering the details of step 4 in Section 2. For convenience, we use $_{+}\!\approx$ to denote equality up to a constant additive factor.

Update equation for $\boldsymbol{\pi}$: $\log q(\boldsymbol{\pi}) {}_{+}\!\approx \mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \mathbf{\Psi})\big)$
${}_{+}\!\approx \mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\mathbf{Z}|\mathbf{\Psi})\big) + \mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\boldsymbol{\pi})\big)$
$= \sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big)\log\pi_k + \log P(\boldsymbol{\pi})$
$= \sum_{k=1}^{K}\log\pi_k\Big[\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big)\Big] + \sum_{k=1}^{K}\log\pi_k(\alpha_k^0 - 1)$
$= \sum_{k=1}^{K}\log\pi_k\Big[\Big(\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big) + \alpha_k^0\Big) - 1\Big]$. Let $\mathrm{I}(\cdot)$ be the indicator function. $q(\boldsymbol{\pi})$ is a Dirichlet distribution with parameters $\alpha = (\alpha_1, \ldots, \alpha_K)$, where

$$\alpha_k = \alpha_k^0 + \sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big). \qquad (8)$$

Update equation for Z_n*:*

$$\begin{aligned}
\log q(Z_n) &{}_{+}\!\approx \mathrm{E}_{-Z_n}\big(\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \mathbf{\Psi})\big) \\
&{}_{+}\!\approx \mathrm{E}_{-Z_n}\big(\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \mathbf{\Psi})\big) + \mathrm{E}_{-Z_n}\big(\log P(\mathbf{Z}|\mathbf{\Psi})
\end{aligned}$$

Let y_{nm} be the observed number of reads aligned to bin m for cell n corresponding to the random variable Y_{nm} and let $D_{n,m,j} \equiv y_{nm}\log\theta_n + y_{nm}\log j - j\theta_n - \log(y_{nm}!)$. Note that $\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi})$ and $\log P(\mathbf{Z}|\mathbf{\Psi})$ can be written as the sum of two terms, one that depends on Z_n and one that does not, i.e., $\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi}) = \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{I}(Z_n=k)\mathrm{I}(C_{km}=j)D_{n,m,j} + \sum_{l\neq n}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{I}(Z_l=k)\mathrm{I}(C_{km}=j)D_{n,m,l}$
and

$$\log P(\mathbf{Z}|\mathbf{\Psi}) = \sum_{k=1}^{K}\mathrm{I}(Z_n=k)\log\pi_k + \sum_{l\neq n}\sum_{k=1}^{K}\mathrm{I}(Z_l=k)\log\pi_k.$$

Consequently, $\log q(Z_n) \overset{+}{\approx} \sum_{k=1}^{K}\mathrm{I}(Z_n=k)\big\{\mathrm{E}_{q(\boldsymbol{\pi})}(\log\pi_k) + \sum_{m=1}^{M}\sum_{j=1}^{J}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\times\mathrm{E}_{q(\theta_n)}(D_{n,m,j})\big\}$.

The distribution $q(Z_n)$ is a Categorical distribution with parameters $\boldsymbol{\pi}_n = (\pi_{n1},\ldots,\pi_{nK})$, where

$$\pi_{nk} = \frac{\exp(\tilde{\pi}_{nk})}{\sum_{k'=1}^{K}\exp(\tilde{\pi}_{nk'})} \tag{9}$$

with

$$\tilde{\pi}_{nk} = \mathrm{E}_{q(\boldsymbol{\pi})}(\log\pi_k) + \sum_{m=1}^{M}\sum_{j=1}^{J}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\mathrm{E}_{q(\theta_n)}\big(D_{n,m,j}\big).$$

Update equation for θ_n:

$$\begin{aligned}\log q(\theta_n) &\overset{+}{\approx} \mathrm{E}_{-\theta_n}\big(\log P(\mathbf{Y},\mathbf{Z},\mathbf{C},\mathbf{\Psi})\\ &\overset{+}{\approx} \mathrm{E}_{-\theta_n}\big(\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi})\big) + \mathrm{E}_{-\theta_n}\big(\log P(\boldsymbol{\theta})\big)\end{aligned}$$

We can write $\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi})$ as in Eq. A.7 and $\log P(\boldsymbol{\theta})$ which depends on θ_n as,

$$\log P(\boldsymbol{\theta}) = (\epsilon^0_{sn}-1)\log\theta_n - \epsilon^0_{rn}\theta_n + \sum_{l\neq n}\big[(\epsilon^0_{sn}-1)\log\theta_l - \epsilon^0_{rn}\theta_l\big] + C.$$

Therefore, $\log q(\theta_n) \overset{+}{\approx} \log\theta_n\Big[\sum_{m=1}^{M}y_{nm}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\Big] - \theta_n\Big[\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)j\Big] + \log\theta_n(\epsilon^0_{sn}-1) - \theta_n\epsilon^0_{rn}$.
Note that $\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big) = 1$.
The distribution $q(\theta_n)$ is a Gamma distribution with shape and rate parameters:

$$\epsilon_{sn} = \epsilon^0_{sn} + \sum_{m=1}^{M}y_{nm} \tag{10}$$

$$\epsilon_{rn} = \epsilon^0_{rn} + \Big\{\sum_{m=1}^{M}\sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\sum_{j=1}^{J}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)j\Big\}.$$

Update equation for $\mathbf{a}^k_i$:

$$\begin{aligned}\log q(\mathbf{a}^k_i) &\overset{+}{\approx} \mathrm{E}_{-\mathbf{a}^k_i}\big(\log P(\mathbf{Y},\mathbf{Z},\mathbf{C},\mathbf{\Psi})\big)\\ &\overset{+}{\approx} \mathrm{E}_{-\mathbf{a}^k_i}\big(\log P(\mathbf{C}|\mathbf{\Psi})\big) + \mathrm{E}_{-\mathbf{a}^k_i}\big(\log P(\mathbf{A})\big)\end{aligned}$$

Disregarding the terms that do not depend on $\mathbf{a}_i^k$ in $\log P(\mathbf{C}|\mathbf{\Psi})$ and $\log P(\mathbf{A})$, we obtain:
$\log q(\mathbf{a}_i^k) {}_{+}\!\approx$
$\sum_{j=1}^{J}\sum_{m=2}^{M} \mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i, C_{km}=j)\big)\log a_{ij}^k + \sum_{j=1}^{J}(\Lambda_{ij}^{k0}-1)\log a_{ij}^k =$
$\sum_{j=1}^{J}\log a_{ij}\Big\{\Big[\Lambda_{ij}^{k0}+\sum_{m=2}^{M}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i, C_{km}=j)\big)\Big]-1\Big\}$. The distribution $q(\mathbf{a}_i^k)$ is Dirichlet with parameters $\mathbf{\Lambda}_i^k = (\Lambda_{i1}^k, \ldots, \Lambda_{iJ}^k)$, where

$$\Lambda_{ij}^k = \Lambda_{ij}^{k0} + \sum_{m=2}^{M}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i, C_{km}=j)\big). \tag{11}$$

Update equation for $\mathbf{C}_k$: In the calculation of $q(\mathbf{C}_k)$, the resulting distribution resembles the posterior probability of the hidden variables in an HMM and only differs in the normalization constant. Similarly to the approach adopted in MacKay (1997), we use a slightly different implementation of forward-backward algorithm. More specifically, we define a graph (one per cluster k) $G = \{V, E\}$ with vertices $V = \{C_{mj} : m \in [M], j \in [J]\}$, having weights denoted as $w_k(C_{mj})$, and edges $E = \{C_{m-1i}C_{mj} : m \in \{2, \ldots, M\}, j \in [J]\}$, having weights denoted as $w_k(C_{m-1i}C_{mj})$. Note that for the simplicity of notation we use C_{mj} instead of $C_{km} = j$ for edges and vertices. We define forward (ϕ_{mj}^k) and backward (β_{mj}^k) quantities similar to HMMs Bishop (2006). Instead of using the terminologies of transition and emission probabilities, we formulate a corresponding algorithm in terms of the weights of the graph, $w_k(C_{m-1i}C_{mj})$ and $w_k(C_{mj})$. The initial transition probability can be defined as $w_k(C_0C_{1j})$. Notice that k indicates the cluster which the graph belongs to. We calculate forward and backward quantities using the weights, and there is no need to normalize the graph weights since a normalization is performed later when calculating the VD. Having the weights, we compute the two posterior probabilities $q(C_{km} = j)$ and $q(C_{km-1} = i, C_{km} = j)$; note that they are normalized by summing over all $j \in [J]$. McGrory and Titterington (2009a) derive the exact normalization of the graph weights, for a similar but less complex graphical model; however, this is unnecessary since the normalization can be performed later, as we do.

$$\begin{aligned}\log q(\mathbf{C}_k) \quad &{}_{+}\!\approx \quad \mathrm{E}_{-\mathbf{C}_k}\big(\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \mathbf{\Psi})\big)\\ &{}_{+}\!\approx \quad \mathrm{E}_{-\mathbf{C}_k}\big(\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \mathbf{\Psi})\big) + \mathrm{E}_{-\mathbf{C}_k}\big(\log P(\mathbf{C}|\mathbf{\Psi})\big).\end{aligned}$$

We can write $\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \mathbf{\Psi})$ as the sum of two terms, one that depends on $\mathbf{C}_k$ and another that does not, that is, $\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \mathbf{\Psi}) = \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{j=1}^{J}\mathrm{I}(Z_n = k)\mathrm{I}(C_{km} = j)D_{n,m,j}$
$+ \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k'\neq k}\sum_{j=1}^{J}\mathrm{I}(Z_n = k)\mathrm{I}(C_{k'm} = j)D_{n,m,j}$.

Let $f(k)$ be a function of k as $\sum_{j=1}^{J}\mathrm{I}(C_{k1} = j)\log\rho_{kj} + \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{I}(C_{km-1} = i, C_{km} = j)\log a_{ij}^k$, We can write $\log P(\mathbf{C}|\mathbf{\Psi}) = f(k) + \sum_{k'\neq k} f(k')$.

Thus, disregarding the terms that do not depend on $\mathbf{C}_k$ we obtain: $\log q(\mathbf{C}_k) {}_{+}\!\approx$
$\sum_{m=1}^{M}\sum_{j=1}^{J}\mathrm{I}(C_{km} = j)\Big\{\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big)\mathrm{E}_{q(\theta_n)}\big(D_{n,m,j}\big)\Big\} +$
$\sum_{j=1}^{J}\mathrm{I}\big(C_{k1} = j\big)\log\rho_{kj} + \sum_{m=2}^{M}\sum_{j=1}^{J}\sum_{i=1}^{J}\mathrm{I}\big(C_{km-1} = i, C_{km} = j\big) \times \mathrm{E}_{q(\mathbf{a}_i^k)}\big(\log a_{ij}^k\big)$. Calculations regarding directed graph are as follows: $\mathrm{u}_k(\mathbf{C}_{1:m-1}, C_m = j) = \prod_{i=1}^{m-1}\prod_{s=1}^{J}\exp\Big\{\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big)\mathrm{E}_{q(\theta_n)}\big(D_{n,i,s}\big)\Big\}^{\mathrm{I}(C_{ki}=s)}$
$\exp\Big\{\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big)\mathrm{E}_{q(\theta_n)}\big(D_{n,m,j}\big)\Big\}$
$\prod_{j=1}^{J}\exp\Big\{\log\rho_{kj}\Big\}^{\mathrm{I}\left(C_{k1}=j\right)}$
$\prod_{t=2}^{m-1}\prod_{s=1}^{J}\prod_{i=1}^{J}\exp\Big\{\mathrm{E}_{q(\mathbf{a}_i^k)}\big(\log a_{is}^k\big)\Big\}^{\mathrm{I}\left(C_{kt-1}=i, C_{kt}=s\right)}$

$\prod_{i=1}^{J} \exp\Big\{\mathrm{E}_{q(\mathbf{a}_i^k)}\big(\log a_{ij}^k\big)\Big\}^{\mathrm{I}\big(C_{km-1}=i, C_{km}=j\big)}$.
and
$\mathrm{v}_k(\mathbf{C}_{m+1:M}, C_m = j) = \prod_{i=m+1}^{M}\prod_{s=1}^{J} \exp\Big\{\sum_{n=1}^{N} \mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big)\mathrm{E}_{q(\theta_n)}\big(D_{n,i,s}\big)\Big\}^{\mathrm{I}(C_{ki}=s)}$
$\prod_{r=m+1}^{M}\prod_{s=1}^{J}\prod_{i=1}^{J} \exp\Big\{\mathrm{E}_{q(\mathbf{a}_i^k)}\big(\log a_{is}^k\big)\Big\}^{\mathrm{I}\big(C_{kr}=i, C_{kr+1}=s\big)}$
$\prod_{i=1}^{J} \exp\Big\{\mathrm{E}_{q(\mathbf{a}_i^k)}\big(\log a_{ji}^k\big)\Big\}^{\mathrm{I}\big(C_{km}=j, C_{km+1}=i\big)}$.

We define $\phi_{mj}^k = \sum_{C_1=1}^{J} .. \sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-1}, C_m = j)$ and
$\beta_{mj}^k = \sum_{C_{m+1}=1}^{J} .. \sum_{C_M=1}^{J} v_k(\mathbf{C}_{m+1:M}, C_m = j)$.

The graph weights are (we assume $w_k(C_0 C_{1j})$ to be fixed):
$w_k(C_{m-1i}C_{mj}) = \exp\big\{\mathrm{E}_{q(\mathbf{a}_i^k)}(\log a_{ij}^k)\big\}$
and
$$w_k(C_{mj}) = \exp\Big\{\sum_{n=1}^{N} \mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big)\mathrm{E}_{q(\theta_n)}\big(D_{n,m,j}\big)\Big\}.$$

Note that we skip writing i, j in the calculations to make them short and more readable:
$\phi_m^k = \sum_{C_1=1}^{J} .. \sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-1}, C_m) = \sum_{C_1=1}^{J} .. \sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-2}, C_{m-1}, C_m) =$
$\sum_{C_1=1}^{J} .. \sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-2}, C_{m-1}) w_k(C_m) w_k(C_{m-1}C_m) = w_k(C_m) \sum_{C_{m-1}=1}^{J} \phi_{m-1}^k w_k(C_{m-1}C_m)$.

$\beta_m^k = \sum_{C_{m+1}=1}^{J} .. \sum_{C_M=1}^{J} v_k(\mathbf{C}_{m+1:M}, C_m) = \sum_{C_{m+1}=1}^{J} .. \sum_{C_M=1}^{J} v_k(\mathbf{C}_{m+2:M}, C_{m+1}, C_m) =$
$\sum_{C_{m+1}=1}^{J} .. \sum_{C_M=1}^{J} v_k(\mathbf{C}_{m+2:M}) w_k(C_m C_{m+1}) w_k(C_{m+1}) = \sum_{C_{m+1}=1}^{J} \beta_{m+1}^k w_k(C_m C_{m+1}) w_k(C_{m+1})$.

We now calculate the posteriors: $\mathrm{q}(\mathrm{C}_{km} = j) = \sum_{C_{k,1:m-1}}^{J} \sum_{C_{k,m+1:M}}^{J} q(C_{k,1:m-1}, C_{km} = j, C_{k,m+1:M}) =$
$\sum_{C_{1:m-1}}^{J} u_k(\mathbf{C}_{1:m-1}, C_m = j) \sum_{C_{m+1:M}}^{J} v_k(\mathbf{C}_{m+1:M}, C_m = j) = \phi_{mj}^k \beta_{mj}^k$.
$q(C_{km-1} = i, C_{km} = j) =$
$\sum_{C_{k,1:m-2}}^{J} \sum_{C_{k,m+1:M}}^{J} q(C_{k,1:m-2}, C_{k,m-1} = i, C_{km} = j, C_{k,m+1:M}) =$
$\sum_{C_{1:m-2}}^{J} u_k(\mathbf{C}_{1:m-2}, C_{m-1} = i) w_k(C_{m-1i}C_{mj}) w_k(C_{mj})$
$\sum_{C_{m+1:M}}^{J} v_k(\mathbf{C}_{m+1:M}, C_m = j) = w_k(C_{m-1i}C_{mj}) w_k(C_{mj}) \phi_{m-1i}^k \beta_{mj}^k$.

Finally, the following are the resulted update equations. Subsequently, the detailed algorithm used in CopyMix is shown in Algorithm 2.
$\mathrm{w}_k(C_{m-1i}C_{mj}) = \exp\big\{\mathrm{E}_{q(\mathbf{a}_i^k)}(\log a_{ij}^k)\big\}$
$$w_k(C_{mj}) = \exp\Big\{\sum_{n=1}^{N} \mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big)\mathrm{E}_{q(\theta_n)}\big(D_{n,m,j}\big)\Big\}$$
$$q(C_{km} = j) = \frac{\phi_{mj}^k \beta_{mj}^k}{\sum_{j=1}^{J} \phi_{mj}^k \beta_{mj}^k}$$
$q(C_{km-1} = i, C_{km} = j) =$
$$\frac{w_k(C_{m-1i}C_{mj}) w_k(C_{mj}) \phi_{m-1i}^k \beta_{mj}^k}{\sum_{j=1}^{J} w_k(C_{m-1i}C_{mj}) w_k(C_{mj}) \phi_{m-1i}^k \beta_{mj}^k}$$

Algorithm 2 CopyMix: a Variational Inference Algorithm

procedure ESTIMATE-POSTERIORS(Y): ▷ Y input is N sequences of length M
Initialization and assigning priors ϵ^0_{sn}, ϵ^0_{rn}, $\mathbf{\Lambda}^{k0}_i$, α^0_k, π_{nk}, $w_k(C_0C_{1j})$
Update ϵ_{sn} using y and ϵ^0_{sn} in Eq. 10
Update $w_k(C_{m-1i}C_{mj})$ using $\mathbf{\Lambda}^{k0}_i$ and π_{nk} in Eq. A.2
Update $w_k(C_{mj})$ using $\mathbf{\Lambda}^{k0}_i$, π_{nk}, ϵ^0_{rn}, ϵ_{sn} and y in Eq. A.2
  repeat
    Update $\alpha^{(c)}_k$ using $\pi^{(c-1)}_{nk}$ and α^0_k in Eq. 8
    Update $q(C_{km}=j)^{(c)}$ and $q(C_{km-1}=i, C_{km}=j)^{(c)}$ using
      $w_k(C_0C_{1j})^{(c-1)}$, $w_k(C_{m-1i}C_{mj})^{(c-1)}$ and
      $w_k(C_{mj})^{(c-1)}$ (Eq. A.2 to A.2)
    Update $\mathbf{\Lambda}^{k(c)}_i$ using $\mathbf{\Lambda}^{k0}_i$ and $q(C_{km-1}=i, C_{km}=j)^{(c)}$ in Eq. 11
    Update $\epsilon^{(c)}_{rn}$ using ϵ^0_{rn}, $\pi^{(c-1)}_{nk}$ and $q(C_{km}=j)^{(c)}$ in Eq. A.2
    Update $\pi^{(c)}_{nk}$ using $\alpha^{(c)}_k$, ϵ_{sn}, $\epsilon^{(c)}_{rn}$, $q(C_{km}=j)^{(c)}$ and y in Eq. 9
    Update $w_k(C_{mj})^{(c)}$ using $\mathbf{\Lambda}^{k(c)}_i$, $\pi^{(c)}_{kn}$, $\epsilon^{(c)}_{rn}$, ϵ_{sn} and y in Eq. A.2
    Update $w_k(C_{m-1i}C_{mj})^{(c)}$ using $\mathbf{\Lambda}^{k(c)}_i$ and $\pi^{(c)}_{nk}$ in Eq. A.2
    $c \leftarrow c+1$
  until convergence of the ELBO in Eq. 2
Output all posterior distributions including:
$q(Z_n=k) \quad \forall n,k;$ ▷ cluster assignment probabilities
$q(C_{km}=j) \quad \forall k,m,j;$ ▷ copy number probabilities
return posterior distribution

A.3 Calculating expectations for section 2.1

In this section we list the calculations of the expected values that are used in the update equations in the previous Appendix. Let $\mathbf{\Phi}$ be the digamma function defined as $\mathbf{\Phi}(x) = \frac{d}{dx}\log\Gamma(x)$ which can be easily calculated via numerical approximation. The values of the expectations taken with respect to the approximated distributions are given as follows where $\mathbf{\Phi}$ is used for some of them.

- $\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big) = \pi_{nk}$
- Due to computational issues in calculating $\mathrm{E}_{q(\theta_n)}\big(D_{n,m,j}\big) = y_{nm}E_{q(\theta_n)}(\log\theta_n) + y_{nm}\log j - jE_{q(\theta_n)}(\theta_n) - \log(y_{nm}!)$, we can instead compute it using the fact that
  $\mathrm{E}_{q(\theta_n)}(\theta_n) = \epsilon_{sn}/\epsilon_{rn}$ and
  $\mathrm{E}_{q(\theta_n)}\big(\log\theta_n\big) = \mathbf{\Phi}(\epsilon_{sn}) + \log E_{q(\theta_n)}(\theta_n) - \log\epsilon_{sn}$.
  Therefore, we obtain $\mathrm{E}_{q(\theta_n)}\big(D_{n,m,j}\big) = \log F_{\tilde{\theta}}(y_{nm}) + y_{nm}\times\big[\mathbf{\Phi}(\epsilon_{sn}) - \log\epsilon_{sn}\big]$, with $F_{\tilde{\theta}}$ being a Poisson p.m.f. with rate parameter $\tilde{\theta} = j\times(\epsilon_{sn}/\epsilon_{rn})$.
- $\mathrm{E}_{q(\boldsymbol{\pi})}(\log\pi_k) = \mathbf{\Phi}(\alpha_k) - \mathbf{\Phi}\Big(\sum_{k=1}^K \alpha_k\Big)$
- $\mathrm{E}_{q(\mathbf{a}^k_i)}(\log a^k_{ij}) = \mathbf{\Phi}(\Lambda^k_{ij}) - \mathbf{\Phi}\Big(\sum_{j=1}^J \Lambda^k_{ij}\Big)$
- $\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big) = q(C_{km}=j)$ and $\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i, C_{km}=j)\big) = q(C_{km-1}=i, C_{km}=j)$, which are calculated in Eq. A.2 and A.2.

A.4 Step 2 of Section 2.2

The logarithm of the joint distribution of $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{C}$ and $\mathbf{\Psi}$ satisfies

$$\log P(\mathbf{Y},\mathbf{Z},\mathbf{C},\mathbf{\Psi}) = \log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi}) + \log P(\mathbf{C}|\mathbf{\Psi}) + \log P(\mathbf{Z}|\mathbf{\Psi}) + \log P(\mathbf{\Psi}), \tag{12}$$

where

$$\begin{aligned}\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi}) &= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{I}(Z_n=k)\mathrm{I}(C_{km}=j)\log F_{\mu_{nj},\sigma^2}(r_{nm}) \\ &= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{I}(Z_n=k)\mathrm{I}(C_{km}=j)(-\frac{1}{2})\Big[\log(2\pi\sigma^2)+\frac{(r_{nm}-j\mu_n)^2}{\sigma^2}\Big], \end{aligned} \tag{13}$$

$$\log P(\mathbf{C}|\mathbf{\Psi}) = \sum_{k=1}^{K}\left[\sum_{j=1}^{J}\mathrm{I}(C_{k1}=j)\log\rho_{kj} + \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{I}(C_{km-1}=i,C_{km}=j)\log a_{ij}^{k}\right], \tag{14}$$

and

$$\log P(\mathbf{Z}|\mathbf{\Psi}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{I}(Z_n=k)\log\pi_k \tag{15}$$

Moreover,

$$\log P(\mathbf{\Psi}) = \log P(\mathbf{A}) + \log P(\boldsymbol{\mu}) + \log P(\frac{1}{\sigma^2}) + \log P(\boldsymbol{\pi}), \tag{16}$$

where, if B is the multivariate Beta function, i.e.,

$$B(\boldsymbol{x}) = \frac{\prod_{k=1}^{K}\Gamma(x_k)}{\Gamma(\sum_{k=1}^{K}x_k)},$$

$$\log P(\boldsymbol{\mu}) = \sum_{n=1}^{N}\big[(-\frac{1}{2})\Big[\log(2\pi\tau^2)+\frac{(\mu_n-\theta)^2}{\tau^2}\Big]\big], \tag{17}$$

$$\log P(\frac{1}{\sigma^2}) = \big[-\frac{\beta}{\sigma^2}+(\alpha-1)\log(\frac{1}{\sigma^2})\big]+C, \tag{18}$$

$$\log P(\mathbf{A}) = \sum_{k=1}^{K}\sum_{i=1}^{J}\big[\sum_{j=1}^{J}(\Lambda_j-1)\log a_{ij}^{k} - \log B(\mathbf{\Lambda})\big], \tag{19}$$

and

$$\log P(\boldsymbol{\pi}) = \sum_{k=1}^{K}(\delta_k-1)\log\pi_k - \log B(\boldsymbol{\delta}). \tag{20}$$

A.5 Step 3 and 4 of Section 2.2

Update equation for π: Since (15) and (20) are the only terms in (12) that depend on $\boldsymbol{\pi}$, the update equation $q^*(\boldsymbol{\pi})$ can be derived as follows.

$$\begin{aligned}&\log q^*(\boldsymbol{\pi}) \\ &= \mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\mathbf{Y},\mathbf{Z},\mathbf{C},\mathbf{\Psi})\big) \\ &+\approx \mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\mathbf{Z}|\mathbf{\Psi})\big) + \mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\boldsymbol{\pi})\big) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\log\pi_k + \log P(\boldsymbol{\pi})\end{aligned}$$

$$
\begin{aligned}
&= \sum_{k=1}^{K} \log \pi_k \Big[\sum_{n=1}^{N} \mathrm{E}_{q(Z_n)} \big(\mathrm{I}(Z_n = k) \big) \Big] + \sum_{k=1}^{K} \log \pi_k (\delta_k - 1) \\
&= \sum_{k=1}^{K} \log \pi_k \Big[\Big(\sum_{n=1}^{N} \mathrm{E}_{q(Z_n)} \big(\mathrm{I}(Z_n = k) \big) + \delta_k \Big) - 1 \Big].
\end{aligned}
$$

Therefore, $q^*(\boldsymbol{\pi})$ is a Dirichlet distribution with parameters $\delta^* = (\delta_1^*, \ldots, \delta_K^*)$, where

$$
\delta_k^* = \delta_k + \sum_{n=1}^{N} \mathrm{E}_{q(Z_n)} \big(\mathrm{I}(Z_n = k) \big). \tag{21}
$$

Update equation for Z_n: Since (A.7) and (15) are the only terms in (12) that depend on Z_n, $q^*(Z_n)$ can be obtained as follows.

$$
\begin{aligned}
&\log q^*(Z_n) \\
&= \mathrm{E}_{-Z_n} \big(\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) \big) \\
&\overset{+}{\approx} \mathrm{E}_{-Z_n} \big(\log P(\mathbf{Y} | \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) \big) + \mathrm{E}_{-Z_n} \big(\log P(\mathbf{Z} | \boldsymbol{\Psi}) \big)
\end{aligned}
$$

Note that $\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi})$ and $\log P(\mathbf{Z}|\boldsymbol{\Psi})$ can be written as the sum of two terms, one that depends on Z_n and one that does not, i.e.,

$$
\begin{aligned}
\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) &= \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{j=1}^{J} \mathrm{I}(Z_n = k) \mathrm{I}(C_{km} = j) (-\frac{1}{2}) \Big[\log(2\pi\sigma^2) + \frac{(r_{nm} - j\mu_n)^2}{\sigma^2} \Big] \\
&+ \sum_{l \neq n} \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{j=1}^{J} \mathrm{I}(Z_l = k) \mathrm{I}(C_{km} = j) (-\frac{1}{2}) \Big[\log(2\pi\sigma^2) + \frac{(r_{lm} - j\mu_l)^2}{\sigma^2} \Big]
\end{aligned} \tag{22}
$$

and

$$
\log P(\mathbf{Z}|\boldsymbol{\Psi}) = \sum_{k=1}^{K} \mathrm{I}(Z_n = k) \log \pi_k + \sum_{l \neq n} \sum_{k=1}^{K} \mathrm{I}(Z_l = k) \log \pi_k.
$$

Consequently,

$$
\log q^*(Z_n) \overset{+}{\approx} \sum_{k=1}^{K} \mathrm{I}(Z_n = k) \Big\{ \mathrm{E}_{q^*(\boldsymbol{\pi})} (\log \pi_k) + \sum_{m=1}^{M} \sum_{j=1}^{J} \mathrm{E}_{q^*(\mathbf{C}_k)} \big(\mathrm{I}(C_{km} = j) \big) \mathrm{E}_{q^*(\sigma) q^*(\mu)} [D_{nmj}] \Big\},
$$

Where:

$$
D_{nmj} = -\frac{1}{2} \Big[\log(2\pi\sigma^2) + \frac{(r_{nm} - j\mu_n)^2}{\sigma^2} \Big] \tag{23}
$$

We conclude that $q^*(Z_n) \sim \text{Categorical}(\boldsymbol{p}_n^*)$ with parameters $\boldsymbol{p}_n^* = (p_{n1}^*, \ldots, p_{nK}^*)$ where

$$
\tilde{p}_{nk} = \mathrm{E}_{q^*(\boldsymbol{\pi})} (\log \pi_k) + \sum_{m=1}^{M} \sum_{j=1}^{J} \mathrm{E}_{q^*(\mathbf{C}_k)} \big(\mathrm{I}(C_{km} = j) \big) \mathrm{E}_{q^*(\frac{1}{\sigma^2}) q^*(\mu)} [D_{nmj}].
$$

and

$$
p_{nk}^* = \frac{\exp(\tilde{p}_{nk})}{\sum_{k'=1}^{K} \exp(\tilde{p}_{nk'})}. \tag{24}
$$

Update equation for μ: Since (A.7) and (17) are the only terms in (12) that depend on μ_n, $q^*(\mu_n)$ can be obtained as follows.

$$
\begin{aligned}
&\log q^*(\mu_n) \\
&= \mathrm{E}_{-\mu_n} \big(\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi})
\end{aligned}
$$

$$+\approx \quad \mathrm{E}_{-\mu_n}\big(\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi})\big) + \mathrm{E}_{-\mu_n}\big(\log P(\boldsymbol{\mu})\big)$$

We can write $\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi})$ as in Eq. A.7, and $\mathrm{E}_{-\mu_n}\big(\log P(\boldsymbol{\mu})\big) = \log P(\boldsymbol{\mu})$, which is:

$$\log P(\boldsymbol{\mu}) \quad = \quad (-\frac{1}{2})\Big[\log(2\pi\tau^2) + \frac{(\mu_n-\theta)^2}{\tau^2}\Big] + \sum_{l\neq n}(-\frac{1}{2})\Big[\log(2\pi\tau^2) + \frac{(\mu_l-\theta)^2}{\tau^2}\Big]$$

Therefore, $\log q^*(\mu_n)$

$$+\approx(-\tfrac{1}{2})\Bigg(\big(\tfrac{\mu_n^2-2\theta\mu_n}{\tau^2}\big) + \textstyle\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\mathrm{E}_{q^*(\frac{1}{\sigma^2})}[\tfrac{j^2\mu_n^2-2j\mu_n r_{nm}}{\sigma^2}]\Bigg)$$

$$+\approx(-\tfrac{1}{2})\Bigg(\big(\tfrac{\mu_n^2-2\theta\mu_n}{\tau^2}\big) + \textstyle\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\tfrac{\alpha^*}{\beta^*}(j^2\mu_n^2-2j\mu_n r_{nm})\Bigg)$$

$$+\approx(-\tfrac{1}{2})\Bigg(\mu_n^2\Bigg[\tfrac{1}{\tau^2} + \tfrac{\alpha^*}{\beta^*}\textstyle\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)j^2\Bigg]$$
$$-\,2\mu_n\Bigg[\tfrac{\theta}{\tau^2} + \tfrac{\alpha^*}{\beta^*}\textstyle\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)jr_{nm}\Bigg]\Bigg)$$

Thus, $q^*(\mu_n)$ is a Normal distribution with parameters:

$$\theta_n^* = \frac{\frac{\theta}{\tau^2} + \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)jr_{nm}\frac{\alpha^*}{\beta^*}}{\frac{1}{\tau^2} + \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\frac{\alpha^*}{\beta^*}j^2} \tag{25}$$

$$\tau_n^{2*} = \frac{1}{\frac{1}{\tau^2} + \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\frac{\alpha^*}{\beta^*}j^2} \tag{26}$$

Update equation for $\frac{1}{\sigma^2}$: Similar to the previous calculations, we do as follows.

$$\begin{aligned}
&\log q^*(\frac{1}{\sigma^2})\\
&\quad = \quad \mathrm{E}_{-\frac{1}{\sigma^2}}\big(\log P(\mathbf{Y},\mathbf{Z},\mathbf{C},\mathbf{\Psi})\\
&\quad +\approx \quad \mathrm{E}_{-\frac{1}{\sigma^2}}\big(\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi})\big) + \mathrm{E}_{-\frac{1}{\sigma^2}}\big(\log P(\frac{1}{\sigma^2})\big)
\end{aligned}$$

We can write $\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi})$ and $\log P(\sigma)$ as they are in Eq. A.7, 18. Putting them together, we achieve the following.

$$\begin{aligned}
&\log q^*(\tfrac{1}{\sigma^2})\\
&+\approx\big[-\tfrac{\beta}{\sigma^2} + (\alpha-1)\log(\tfrac{1}{\sigma^2})\big] + \textstyle\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\mathrm{E}_{q^*(\mu_n)}[D_{nmj}]\\
&+\approx -\tfrac{\beta}{\sigma^2} + (\alpha-1)\log(\tfrac{1}{\sigma^2})\\
&-\tfrac{1}{2}\textstyle\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\big[-\log(\tfrac{1}{\sigma^2}) + \tfrac{r_{nm}^2-2jr_{nm}\theta_n^*+j^2(\tau_n^{*2}+\theta_n^{*2})}{\sigma^2}\big]\\
&=\log(\tfrac{1}{\sigma^2})\Big\{\alpha^*-1\Big\} - \tfrac{1}{\sigma^2}\Big\{\beta^*\Big\}
\end{aligned}$$

Therefore, $q^*(\frac{1}{\sigma^2})$ is a Gamma distribution with parameters:

$$\alpha^* = \alpha + \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big) = \alpha + \frac{NM}{2} \tag{27}$$

$$\beta^* = \beta + \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\Big(r_{nm}^2 - 2jr_{nm}\theta_n^* + j^2(\tau_n^{*2}+\theta_n^{*2})\Big) \tag{28}$$

Update equation for $\mathbf{a}_i^k$: Because (14) and (19) are the only terms in (12) that depend on $\mathbf{a}_i^k$, we calculate $q^*(\mathbf{a}_i^k)$ as follows.

$$\begin{aligned}
&\log q^*(\mathbf{a}_i^k) \\
&\quad = \quad \mathrm{E}_{-\mathbf{a}_i^k}\big(\log P(\mathbf{Y},\mathbf{Z},\mathbf{C},\mathbf{\Psi})\big) \\
&\quad {}_{+}\!\approx \quad \mathrm{E}_{-\mathbf{a}_i^k}\big(\log P(\mathbf{C}|\mathbf{\Psi})\big) + \mathrm{E}_{-\mathbf{a}_i^k}\big(\log P(\mathbf{A})\big)
\end{aligned}$$

Disregarding the terms that do not depend on $\mathbf{a}_i^k$ in $\log P(\mathbf{C}|\mathbf{\Psi})$ and $\log P(\mathbf{A})$, we obtain:

$$\begin{aligned}
\log q^*(\mathbf{a}_i^k) \quad {}_{+}\!\approx \quad & \sum_{j=1}^{J}\sum_{m=2}^{M} \mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i, C_{km}=j)\big)\log a_{ij}^k + \sum_{j=1}^{J}(\Lambda_j - 1)\log a_{ij}^k \\
= \quad & \sum_{j=1}^{J}\log a_{ij}\Big\{\Big[\Lambda_j + \sum_{m=2}^{M}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i, C_{km}=j)\big)\Big] - 1\Big\}.
\end{aligned}$$

Therefore, $q^*(\mathbf{a}_i^k)$ is Dirichlet with parameters $\mathbf{\Lambda}_i^{k*} = (\Lambda_{i1}^{k*}, \ldots, \Lambda_{iJ}^{k*})$ where

$$\Lambda_{ij}^{k*} = \Lambda_j + \sum_{m=2}^{M}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i, C_{km}=j)\big) \tag{29}$$

Update equation for $\mathbf{C}_k$: Since only (A.7) and (14) depend on $\mathbf{C}_k$, $\log q^*(\mathbf{C}_k)$ can be calculated as follows.

$$\begin{aligned}
&\log q^*(\mathbf{C}_k) \\
&\quad = \quad \mathrm{E}_{-\mathbf{C}_k}\big(\log P(\mathbf{Y},\mathbf{Z},\mathbf{C},\mathbf{\Psi})\big) \\
&\quad {}_{+}\!\approx \quad \mathrm{E}_{-\mathbf{C}_k}\big(\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi})\big) + \mathrm{E}_{-\mathbf{C}_k}\big(\log P(\mathbf{C}|\mathbf{\Psi})\big).
\end{aligned}$$

Similarly we can write $\log P(\mathbf{C}|\mathbf{\Psi})$ as

$$\begin{aligned}
\log P(\mathbf{C}|\mathbf{\Psi}) \quad = \quad & \sum_{j=1}^{J}\mathrm{I}(C_{k1}=j)\log\rho_{kj} + \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{I}(C_{km-1}=i, C_{km}=j)\log a_{ij}^k, \\
& + \sum_{k'\neq k}\Big[\sum_{j=1}^{J}\mathrm{I}(C_{k'1}=j)\log\rho_{k'j} + \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{I}(C_{k'm-1}=i, C_{k'm}=j)\log a_{ij}^{k'}\Big].
\end{aligned}$$

Thus, disregarding the terms that do not depend on $\mathbf{C}_k$ we obtain:

$$\begin{aligned}
&\log q^*(\mathbf{C}_k) \\
&\quad {}_{+}\!\approx \quad \sum_{m=1}^{M}\sum_{j=1}^{J}\mathrm{I}(C_{km}=j)\Big\{\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q^*(\mu_n),q^*(\frac{1}{\sigma^2})}(D_{n,m,j})\Big\} \\
&\qquad + \sum_{j=1}^{J}\mathrm{I}\big(C_{k1}=j\big)\mathrm{E}_{q^*(\boldsymbol{\rho}_k)}\big(\log\rho_{kj}\big) \;+\; \sum_{m=2}^{M}\sum_{j=1}^{J}\sum_{i=1}^{J}\mathrm{I}\big(C_{km-1}=i, C_{km}=j\big)\mathrm{E}_{q^*(\mathbf{a}_i^k)}\big(\log a_{ij}^k\big),
\end{aligned}$$

Calculations regarding directed graph: We define graph $G = \{V, E\}$ where we have each vertex as $V = \{C_{mj} : m \in M, j \in J\}$ with weight $w(C_{mj})$ and each edge as $E = \{C_{mj}C_{m+1j} : m \in M, j \in J\}$ with weight $w(C_{mj}C_{m+1j})$. We define quantities of forward, ϕ_{mj}, and backward, β_{mj}, similar to HMM as below (the calculations are based on $\log q^*(\mathbf{C}_k)$):

$$\begin{aligned}
&u_k(\mathbf{C}_{1:m-1}, C_m = j) \\
&\quad {}_{+}\!\approx \quad \prod_{i=1}^{m-1}\prod_{s=1}^{J}\exp\Big\{\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)E_{q^*(\mu_n),q^*(\frac{1}{\sigma^2})(D_{n,i,s})}\Big\}^{\mathrm{I}(C_{ki}=s)}\;\exp\Big\{\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q^*(\mu_n),q^*(\frac{1}{\sigma^2})}(D_{n,m,j})\Big\}
\end{aligned}$$

$$\prod_{j=1}^{J} \exp\Big\{\mathrm{E}_{q^*(\boldsymbol{\rho}_k)}\big(\log \rho_{kj}\big)\Big\}^{\mathrm{I}\left(C_{k1}=j\right)}$$
$$\prod_{t=2}^{m-1}\prod_{s=1}^{J}\prod_{i=1}^{J} \exp\Big\{\mathrm{E}_{q^*(\mathbf{a}_i^k)}\big(\log a_{is}^k\big)\Big\}^{\mathrm{I}\left(C_{kt-1}=i,C_{kt}=s\right)} \prod_{i=1}^{J} \exp\Big\{\mathrm{E}_{q^*(\mathbf{a}_i^k)}\big(\log a_{ij}^k\big)\Big\}^{\mathrm{I}\left(C_{km-1}=i,C_{km}=j\right)}$$

$\phi_{mj} = \sum_{C_1=1}^{J} .. \sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-1}, C_m = j)$ and
$\beta_{mj} = \sum_{C_{m+1}=1}^{J} .. \sum_{C_M=1}^{J} v_k(\mathbf{C}_{m+1:M}, C_m = j)$.

$$v_k(\mathbf{C}_{m+1:M}, C_m = j)$$
$$+\approx \prod_{i=m+1}^{M}\prod_{s=1}^{J} \exp\Big\{\sum_{n=1}^{N} \mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q^*(\mu_n),q^*(\frac{1}{\sigma^2})}(D_{n,i,s})\Big\}^{\mathrm{I}(C_{ki}=s)}$$
$$\prod_{r=m+1}^{M}\prod_{s=1}^{J}\prod_{i=1}^{J} \exp\Big\{\mathrm{E}_{q^*(\mathbf{a}_i^k)}\big(\log a_{is}^k\big)\Big\}^{\mathrm{I}\left(C_{kr}=i,C_{kr+1}=s\right)} \prod_{i=1}^{J} \exp\Big\{\mathrm{E}_{q^*(\mathbf{a}_i^k)}\big(\log a_{ji}^k\big)\Big\}^{\mathrm{I}\left(C_{km}=j,C_{km+1}=i\right)}$$

Instead of using terminologies of transition and emission probabilities we formulate those as weights of the graph; $w_k(C_{mj}C_{m+1j})$ and $w_k(C_{mj})$ respectively. The initial transition probability can be define as $w_k(C_0C_{1j})$ assuming the source starts at 1. Notice that k shows the cluster which the graph belongs to. We can calculate forward and backwards using dynamic programming; having those values we can compute the two posterior probabilities of $q_k^*(C_m = j)$ and $q_k^*(C_{m-1} = i, C_m = j)$ which are the expectations of the indicator functions; we then normalized them by summing over all $j \in J$. The graph weights are: $w_k(C_0C_{1j}) = \exp\big\{\mathrm{E}_{q^*(\boldsymbol{\rho}_k)}(\log \rho_{kj})\big\}$; $w_k(C_{m-1i}C_{mj}) = \exp\big\{\mathrm{E}_{q^*(\mathbf{a}_i^k)}(\log a_{ij}^k)\big\}$ and $w_k(C_{mj}) = \exp\big\{\sum_{n=1}^{N} \mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q^*(\mu_n),q^*(\frac{1}{\sigma^2})}(D_{n,i,s})\big\}$.
Note that we skip writing i, j, k in the calculations to make them short and more readable:
$\phi_m = \sum_{C_1=1}^{J} .. \sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-1}, C_m) = \sum_{C_1=1}^{J} .. \sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-2}, C_{m-1}, C_m) =$
$\sum_{C_1=1}^{J} .. \sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-2}, C_{m-1})w(C_m)w(C_{m-1}C_m) = w(C_m)\sum_{C_{m-1}=1}^{J} \phi_{m-1}w(C_{m-1}C_m)$
$\beta_m = \sum_{C_{m+1}=1}^{J} .. \sum_{C_M=1}^{J} v_k(\mathbf{C}_{m+1:M}, C_m) = \sum_{C_{m+1}=1}^{J} .. \sum_{C_M=1}^{J} v_k(\mathbf{C}_{m+2:M}, C_{m+1}, C_m) =$
$\sum_{C_{m+1}=1}^{J} .. \sum_{C_M=1}^{J} v_k(\mathbf{C}_{m+2:M})w(C_mC_{m+1})w(C_{m+1}) = \sum_{C_{m+1}=1}^{J} \beta_{m+1}w(C_mC_{m+1})w(C_{m+1})$

We now calculate the posteriors:
$\mathrm{q}_k(C_m = j) = \sum_{C_{1:m-1}}^{J} \sum_{C_{m+1:M}}^{J} q_k(C_{1:m-1}, C_m = j, C_{m+1:M})$
$= \sum_{C_{1:m-1}}^{J} u_k(\mathbf{C}_{1:m-1}, C_m = j) \sum_{C_{m+1:M}}^{J} v_k(\mathbf{C}_{m+1:M}, C_m = j) = \phi_{mj}\beta_{m,j}$

$\mathrm{q}_k(C_{m-1} = i, C_m = j) = \sum_{C_{1:m-2}}^{J} \sum_{C_{m+1:M}}^{J} q_k(C_{1:m-2}, C_{m-1} = i, C_m = j, C_{m+1:M}) =$
$\sum_{C_{1:m-2}}^{J} u_k(\mathbf{C}_{1:m-2}, C_{m-1} = i)w(C_{m-1i}C_{mj})w(C_{mj}) \sum_{C_{m+1:M}}^{J} v_k(\mathbf{C}_{m+1:M}, C_m = j) = w(C_{m-1i}C_{mj})w(C_{mj})\phi_{m-1j}\beta_{m,j}$

A.6 Calculating expectations and ELBO for Section 2.2

Let $\boldsymbol{\Psi}$ be the digamma function defined as

$$\boldsymbol{\Psi}(x) = \frac{d}{dx}\log\Gamma(x), \tag{30}$$

which can be easily calculated via numerical approximation. The values of the expectations above taken with respect to the approximated distributions are given as follows.

$$\mathrm{E}_{q(Z_n)}(\mathrm{I}(Z_n = k)) = \pi_{nk}^*$$

$$\mathrm{E}_{q(\mu_n)}(\mu_n) = \theta_n^*$$

$$E_{q(\frac{1}{\sigma^2})}(\log \frac{1}{\sigma^2}) = \mathbf{\Psi}(\alpha^*) - \log \beta^*$$

$$\mathrm{E}_{q(\mu_n)}(\mu_n^2) = \theta_n^{*2} + \tau_n^{*2}$$

$$\mathrm{E}_{q(\frac{1}{\sigma^2})}(\frac{1}{\sigma^2}) = \frac{\alpha^*}{\beta^*}$$

$$\mathrm{E}_{q(\frac{1}{\sigma^2})}(D_{nmj}) = -\tfrac{1}{2}\Bigg[\mathrm{E}\Big[\log(2\pi\sigma^2) + \tfrac{(r_{nm}-j\mu_n)^2}{\sigma^2}\Big]\Bigg] =$$

$$-\tfrac{1}{2}\Bigg[\log(2\pi) - \mathrm{E}[\log(\tfrac{1}{\sigma^2})] + \mathrm{E}[\tfrac{r_{nm}^2}{\sigma^2}] - 2jr_{nm}\mathrm{E}[\tfrac{\mu_n}{\sigma^2}] + \mathrm{E}[\tfrac{j^2\mu_n^2}{\sigma^2}]\Bigg] =$$

$$-\tfrac{1}{2}\Bigg[\log(2\pi) - \mathrm{E}_{q(\sigma)}(\log \tfrac{1}{\sigma^2}) + r_{nm}^2\mathrm{E}_{q(\sigma)}(\tfrac{1}{\sigma^2}) - 2jr_{nm}(\mu_n)\mathrm{E}_{q(\sigma)}(\tfrac{1}{\sigma^2}) + (j^2\mu_n^2)\mathrm{E}_{q(\sigma)}(\tfrac{1}{\sigma^2})\Bigg]$$

$$\mathrm{E}_{q(\mu_n)}(D_{nmj}) = -\tfrac{1}{2}\Bigg[\mathrm{E}\Big[\log(2\pi\sigma^2) + \tfrac{(r_{nm}-j\mu_n)^2}{\sigma^2}\Big]\Bigg] =$$

$$-\tfrac{1}{2}\Bigg[\log(2\pi) - \log(\tfrac{1}{\sigma^2}) + \tfrac{r_{nm}^2}{\sigma^2} - 2jr_{nm}\mathrm{E}[\tfrac{\mu_n}{\sigma^2}] + \mathrm{E}[\tfrac{j^2\mu_n^2}{\sigma^2}]\Bigg] =$$

$$-\tfrac{1}{2}\Bigg[\log(2\pi) - \log \tfrac{1}{\sigma^2} + \tfrac{r_{nm}^2}{\sigma^2} - \tfrac{2jr_{nm}}{\sigma^2}\mathrm{E}_{q(\mu_n)}(\mu_n) + \tfrac{j^2}{\sigma^2}\mathrm{E}_{q(\mu_n)}(\mu_n^2)\Bigg]$$

$$\mathrm{E}_{q(\frac{1}{\sigma^2}),q(\mu_n)}(D_{nmj}) = -\tfrac{1}{2}\Bigg[\mathrm{E}\Big[\log(2\pi\sigma^2) + \tfrac{(r_{nm}-j\mu_n)^2}{\sigma^2}\Big]\Bigg] =$$

$$-\tfrac{1}{2}\Bigg[\log(2\pi) - \mathrm{E}[\log(\tfrac{1}{\sigma^2})] + \mathrm{E}[\tfrac{r_{nm}^2}{\sigma^2}] - 2jr_{nm}\mathrm{E}[\tfrac{\mu_n}{\sigma^2}] + \mathrm{E}[\tfrac{j^2\mu_n^2}{\sigma^2}]\Bigg] =$$

$$-\tfrac{1}{2}\Bigg[\log(2\pi) - \mathrm{E}_{q(\frac{1}{\sigma^2})}(\log \tfrac{1}{\sigma^2}) + r_{nm}^2\mathrm{E}_{q(\frac{1}{\sigma^2})}(\tfrac{1}{\sigma^2}) - 2jr_{nm}\mathrm{E}_{q(\mu_n)}(\mu_n)\mathrm{E}_{q(\frac{1}{\sigma^2})}(\tfrac{1}{\sigma^2}) + j^2\mathrm{E}_{q(\mu_n)}(\mu_n^2)\mathrm{E}_{q(\frac{1}{\sigma^2})}(\tfrac{1}{\sigma^2})\Bigg]$$

$$\mathrm{E}_{q(\boldsymbol{\pi})}(\log \boldsymbol{\pi}) = \mathbf{\Psi}(\delta_k^*) - \mathbf{\Psi}\Big(\sum_{k=1}^{K}\delta_k^*\Big) \quad \forall k \in K \tag{31}$$

$$\mathrm{E}_{q(\mathbf{a}_i^k)}(\log \mathbf{a}_i^k) = \mathbf{\Psi}(\Lambda_{ij}^{k*}) - \mathbf{\Psi}\Big(\sum_{j=1}^{J}\Lambda_{ij}^{k*}\Big) \quad \forall j \in J \tag{32}$$

Using the results above regarding the expectations, we update the parameters of the approximated distributions iteratively.
We then conduct many iterations until the convergence of the ELBO in Eq. 2 which is calculated as below. An assumption in the ELBO calculation is that we ignore all the constants contributing in the ELBO value.

$$\mathrm{ELBO(q)} = \mathrm{E}\big[\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi}) + \log P(\mathbf{Z}|\mathbf{\Psi}) + \log P(\mathbf{C}|\mathbf{\Psi}) + \log P(\boldsymbol{A}) + \log P(\boldsymbol{\mu}) + \log P(\tfrac{1}{\sigma^2}) + \log P(\boldsymbol{\pi})\big]$$
$$-\mathrm{E}\big[\log q(\mathbf{Z}) + \log q(\mathbf{C}) + \log q(\boldsymbol{A}) + \log q(\boldsymbol{\mu}) + \log q(\tfrac{1}{\sigma^2}) + \log q(\boldsymbol{\pi}))\big] =$$
$$\mathrm{E}\Big[\textstyle\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{I}(Z_n = k)\mathrm{I}(C_{km} = j)D_{nmj} + \sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{I}(Z_n = k)\log \pi_k$$

$+\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{I}(C_{k1}=j)\log\rho_{kj}+\sum_{k=1}^{K}\sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{I}(C_{km-1}=i,C_{km}=j)\log a_{ij}^{k}$
$+\sum_{k=1}^{K}\sum_{i=1}^{J}\sum_{j=1}^{J}(\Lambda_j-1)\log a_{ij}^{k}+\sum_{n=1}^{N}(\frac{-1}{2})\Big(\log(2\pi\tau^2)+\frac{\mu_n^2-2\theta\mu_n+\theta^2}{\tau^2}\Big)$
$+(\frac{-\beta}{\sigma^2})+(\alpha-1)\log\frac{1}{\sigma^2}+\log P(\boldsymbol{\pi})\Big]$
$-\mathrm{E}\Big[\sum_{n=1}^{N}\log q(Z_n)+\sum_{k=1}^{K}\log q(\mathbf{C}_k)+\sum_{k=1}^{K}\sum_{i=1}^{J}\log q(\mathbf{a}_i^k)+\sum_{n=1}^{N}\log q(\mu_n)+\log q(\frac{1}{\sigma^2})+\log q(\boldsymbol{\pi})\Big]$

We calculate the $\mathrm{E}[\log q(\mathbf{C}_k)]$ by decomposing it into initial and transition components. Note that the initial probabilities, ρ_{kj}s, are fixed and, therefore, the term corresponding to that cancels out the corresponding term in $\mathrm{E}[\log P(\mathbf{C}|\boldsymbol{\Psi})]$. The remaining term, the transition component, is calculated by the following using $q(C_{km}=j|C_{km-1}=i)=\frac{q(C_{km}=j,C_{km-1}=i)}{q(C_{km-1}=i)}$, as below:

$\mathrm{E}_{q(\mathbf{C}_k)}[\log q(\mathbf{C}_k)]=\mathrm{E}_{q(\mathbf{C}_k)}[\log\prod_{m=2}^{M}\prod_{i=1}^{J}\prod_{j=1}^{J}q(C_{km}=j|C_{km-1}=i)^{\mathrm{I}(C_{km-1}=i,C_{km}=j)}]=$
$\sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{E}_{q(C_{km-1},C_{km})}\Big[\mathrm{I}(C_{km-1}=i,C_{km}=j)\Big]\log q(C_{km}=j|C_{km-1}=i)$
$\sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{E}_{q(C_{km-1},C_{km})}\Big[\mathrm{I}(C_{km-1}=i,C_{km}=j)\Big]\log\frac{\mathrm{E}_{q(C_{km-1},C_{km})}\Big[\mathrm{I}(C_{km-1}=i,C_{km}=j)\Big]}{\mathrm{E}_{q(C_{km-1})}\Big[\mathrm{I}(C_{km-1}=i)\Big]}$

We calculate the $\mathrm{E}[\log p(Z_n|\Psi)]$ as below:

$$\mathrm{E}_{(q(Z_n),q(\boldsymbol{\pi}))}\Big[\log p(Z_n|\Psi)\Big]=\mathrm{E}_{(q(Z_n),q(\boldsymbol{\pi}))}\Big[\sum_{k=1}^{K}\mathrm{I}(Z_n=k)\log\pi_k\Big]=\sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n=k)\Big]\mathrm{E}_{q(\boldsymbol{\pi})}\Big[\log\pi_k\Big] \tag{33}$$

We calculate the $\mathrm{E}[\log q(Z_n)]$ as below. Note that the expectation of $q(Z_n)$ w.r.t. $q(Z_n)$ is equal to $q(Z_n)$. Also, we know that $\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n=k)\Big]=q(Z_n=k)$.

$$\mathrm{E}_{q(Z_n)}\Big[\log q(Z_n)\Big]=\mathrm{E}_{q(Z_n)}\Big[\sum_{k=1}^{K}\mathrm{I}(Z_n=k)\log q(Z_n=k)\Big]=\sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n=k)\Big]\log q(Z_n=k) \tag{34}$$

We calculate the $\mathrm{E}[\log P(\mu_n)]$ as below:

$$\mathrm{E}_{q(\mu_n)}[\log p(\mu_n)]=(\frac{-1}{2})\Big(\log(2\pi\tau^2)+\frac{\mathrm{E}_{q(\mu_n)}\Big[\mu_n^2\Big]-2\theta\mathrm{E}_{q(\mu_n)}\Big[\mu_n\Big]+\theta^2}{\tau^2}\Big) \tag{35}$$

We calculate the $\mathrm{E}[\log q(\mu_n)]$ as below:

$$\mathrm{E}_{q(\mu_n)}[\log q(\mu_n)]=(\frac{-1}{2})\Big(\log(2\pi\tau_n^{*2})+\frac{\mathrm{E}_{q(\mu_n)}\Big[\mu_n^2\Big]-2\theta_n^*\mathrm{E}_{q(\mu_n)}\Big[\mu_n\Big]+\theta_n^{*2}}{\tau_n^{*2}}\Big)=(\frac{-1}{2})\Big(\log(2\pi\tau_n^{*2})+1\Big) \tag{36}$$

We calculate the $\mathrm{E}[\log P(\boldsymbol{\pi})]$ as below:

$\mathrm{E}_{q(\boldsymbol{\pi})}(\log P(\boldsymbol{\pi}))=\sum_{k=1}^{K}(\delta_k-1)\mathrm{E}_{q(\boldsymbol{\pi})}(\log\boldsymbol{\pi})-\log B(\boldsymbol{\delta})=\sum_{k=1}^{K}(\delta_k-1)\Big[\boldsymbol{\Psi}(\delta_k^*)-\boldsymbol{\Psi}\big(\sum_{k=1}^{K}\delta_k^*\big)\Big]-\log B(\boldsymbol{\delta})$
$=\Big(\sum_{k=1}^{K}\delta_k-K\Big)\boldsymbol{\Psi}\Big(\sum_{k=1}^{K}\delta_k^*\Big)-\sum_{k=1}^{K}\boldsymbol{\Psi}(\delta_k^*)(\delta_k-1)-\log B(\boldsymbol{\delta})$

We calculate the $\mathrm{E}(\log q(\boldsymbol{\pi}))$ as below:

$$\mathrm{E}_{q(\boldsymbol{\pi})}(\log q(\boldsymbol{\pi})) = \Big(\sum_{k=1}^{K}\delta_k^* - K\Big)\boldsymbol{\Psi}\Big(\sum_{k=1}^{K}\delta_k^*\Big) - \sum_{k=1}^{K}\boldsymbol{\Psi}(\delta_k^*)(\delta_k^*-1) - \log B(\boldsymbol{\delta}^*) \tag{37}$$

$$\begin{aligned}
&\mathrm{ELBO(q)} = \textstyle\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n=k)\Big]\mathrm{E}_{q(C_{km})}\Big[\mathrm{I}(C_{km}=j)\Big]\mathrm{E}_{q(\mu_n)q(\frac{1}{\sigma^2})}\Big[D_{nmj}\Big]\\
&+\textstyle\sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n=k)\Big]\mathrm{E}_{q(\boldsymbol{\pi})}\Big[\log\pi_k\Big] + \sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(C_{k1})}\Big[\mathrm{I}(C_{k1}=j)\Big]\log\rho_{kj}\\
&+\textstyle\sum_{k=1}^{K}\sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{E}_{q(C_{km-1},C_{km})}\Big[\mathrm{I}(C_{km-1}=i,C_{km}=j)\Big]\mathrm{E}_{q(\mathbf{a}_i^k)}\Big[\log a_{ij}^k\Big]\\
&+\textstyle\sum_{k=1}^{K}\sum_{i=1}^{J}\sum_{j=1}^{J}(\Lambda_j-1)\mathrm{E}_{q(\mathbf{a}_i^k)}\Big[\log a_{ij}^k\Big] + \sum_{n=1}^{N}(\frac{-1}{2})\Bigg(\log(2\pi\tau^2) + \frac{\mathrm{E}_{q(\mu_n)}\Big[\mu_n^2\Big] - 2\theta\mathrm{E}_{q(\mu_n)}\Big[\mu_n\Big] + \theta^2}{\tau^2}\Bigg)\\
&+(-\beta)\mathrm{E}_{q(\frac{1}{\sigma^2})}\Big[\tfrac{1}{\sigma^2}\Big] + (\alpha-1)\mathrm{E}_{q(\frac{1}{\sigma^2})}\Big[\log\tfrac{1}{\sigma^2}\Big] + \mathrm{E}_{q(\boldsymbol{\pi})}\Big[\log P(\boldsymbol{\pi})\Big]\\
&-\textstyle\sum_{n=1}^{N}\Big[\sum_{k=1}^{K}p_{nk}^*\log p_{nk}^*\Big] - \sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{j=1}^{J}\mathrm{E}_{q(C_{km})}\Big[\mathrm{I}(C_{km}=j)\Big]\mathrm{E}_{q(\mu_n)q(\frac{1}{\sigma^2})}\Big[D_{nmj}\Big]\\
&-\textstyle\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(C_{k1})}\Big[\mathrm{I}(C_{k1}=j)\Big]\log\rho_{kj}\\
&-\textstyle\sum_{k=1}^{K}\sum_{m=2}^{M}\sum_{j=1}^{J}\sum_{i=1}^{J}\mathrm{E}_{q(C_{km-1},C_{km})}\Big[\mathrm{I}(C_{km-1}=i,C_{km}=j)\Big]\log\frac{\mathrm{E}_{q(C_{km-1},C_{km})}\Big[\mathrm{I}(C_{km-1}=i,C_{km}=j)\Big]}{\mathrm{E}_{q(C_{km-1})}\Big[\mathrm{I}(C_{km-1}=i)\Big]}\\
&-\textstyle\sum_{k=1}^{K}\sum_{i=1}^{J}\Big[\sum_{j=1}^{J}\mathrm{E}_{q(\mathbf{a}_i^k)}\Big[\log a_{ij}^k\Big]\Big\{\Lambda_j^*-1\Big\}\Big]\\
&-\textstyle\sum_{n=1}^{N}(\frac{-1}{2})\Bigg(\log(2\pi\tau_n^{*2})+1\Bigg)\\
&-\Bigg(\alpha^*-\ln\beta^*+\ln\Gamma(\alpha^*)+(1-\alpha^*)\boldsymbol{\Psi}(\alpha^*)\Bigg)\\
&-\mathrm{E}_{q(\boldsymbol{\pi})}\Big[\log q(\boldsymbol{\pi})\Big]
\end{aligned}$$

We now approximate the term, $\Bigg(\alpha^*-\ln\beta^*+\ln\Gamma(\alpha^*)+(1-\alpha^*)\boldsymbol{\Psi}(\alpha^*)\Bigg)$ in the above, by using approximations:

1. Stirling's: $\ln\Gamma(\alpha^*) = (\alpha^*-1)\ln(\alpha^*-1)-\alpha^*+1$

2. digamma-approximation: $\boldsymbol{\Psi}(\alpha^*) = \ln(\alpha^*) - \frac{1}{2\alpha^*}$

3. Also: $\ln\Gamma(\alpha^*) = \ln(\frac{\Gamma(\alpha^*+1)}{\alpha^*}) = (\alpha^*\ln(\alpha^*)-\alpha^*) - \ln(\alpha^*)$

Putting 3 and 2 in the original formula we obtain:

$$\alpha^*-\ln\beta^*+\alpha^*\ln(\alpha^*)-\alpha^*-\ln(\alpha^*)+(\ln(\alpha^*)-\tfrac{1}{2\alpha^*})-\alpha^*\ln(\alpha^*)+\tfrac{1}{2} = \tfrac{1}{2}(1-\tfrac{1}{\alpha^*})-\ln\beta^*$$

A.7 VI for SNV-CopyMix

Let $\boldsymbol{\Psi}$ be the set containing all the model parameters, i.e., $\boldsymbol{\Psi} = \{\mathbf{A}, \boldsymbol{\mu}, \sigma, \boldsymbol{\pi}, \xi\}$, where

- $\mathbf{A} = \{\mathbf{A}_k \text{ for } k = 1,\ldots,K\}$ with $\mathbf{A}_k = \{a_{ij}^k \text{ for } i = 1,\ldots,J \text{ and } j = 1,\ldots,J\}$;
- $\boldsymbol{\mu} = \{\mu_n \text{ for } n = 1,\ldots,N\}$;
- σ;
- $\boldsymbol{\pi} = (\pi_1,\ldots,\pi_K)$;

- ξ.

In order to infer $\mathbf{\Psi}$ and the hidden states $\mathbf{Z} = (Z_1, \ldots, Z_n)$, $\mathbf{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_K\}$, and $\mathbf{S} = \{S_1, \ldots, S_K\}$ we apply the Variational Inference (VI) methodology; that is, we derive an algorithm that, for given data, approximates the posterior distribution by finding the Variational Distribution (VD),

$$q(\mathbf{Z}, \mathbf{C}, \mathbf{S}, \mathbf{\Psi}) \tag{38}$$

with smallest Kullback-Leibler divergence to the posterior distribution

$$P(\mathbf{Z}, \mathbf{C}, \mathbf{S}, \mathbf{\Psi} | \mathbf{Y}, \mathbf{X}),$$

which is equivalent to maximizing the evidence lower bound (ELBO) given by

$$\text{ELBO}(q) = \text{E}\big[\log P(\mathbf{Y}, \mathbf{X}, \mathbf{Z}, \mathbf{C}, \mathbf{S}, \mathbf{\Psi})\big] - \text{E}\big[\log q(\mathbf{Z}, \mathbf{C}, \mathbf{S}, \mathbf{\Psi})\big]. \tag{39}$$

We consider the following prior distributions for the parameters in $\mathbf{\Psi}$.

- $\mathbf{A}_i^k = (A_{i1}^k, \ldots, A_{iJ}^k) \sim \text{Dirichlet}(\mathbf{\Lambda})$
- $\mu_n \sim \text{Normal}(\theta_n, \tau_n^2)$. The conjugate prior concerning the mean of Normal distribution is Normal distribution.
- $\frac{1}{\sigma^2} \sim \text{Gamma}(\alpha, \beta)$. The conjugate prior concerning the precision of Normal distribution is Gamma distribution.
- $\boldsymbol{\pi} \sim \text{Dirichlet}(\boldsymbol{\delta})$.
- $\xi_k \sim \text{Beta}(\gamma_k, \eta_k)$.

In what follows we describe the main steps of the VI algorithm for inferring $\mathbf{Z}, \mathbf{C}, \mathbf{S}$ and $\mathbf{\Psi}$.

Step 1. VD factorization

We assume the following factorization of the VD:

$$q(\mathbf{Z}, \mathbf{C}, \mathbf{S}, \mathbf{\Psi}) = \Big[\prod_{n=1}^{N} q(Z_n)\Big]\Big[\prod_{k=1}^{K} q(\mathbf{C}_k)\Big]\Big[\prod_{k=1}^{K}\prod_{i=1}^{J} q(\mathbf{A}_i^k)\Big]\Big[\prod_{n=1}^{N} q(\mu_n)\Big]\Big[\prod_{k=1}^{K} q(\boldsymbol{S}_k)\Big]\Big[\prod_{k=1}^{K} q(\xi_k)\Big] q(\frac{1}{\sigma^2}) q(\boldsymbol{\pi}). \tag{40}$$

Step 2. Joint distribution of observed data, hidden variables, and parameters

The logarithm of the joint distribution of $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{C}$ and $\mathbf{\Psi}$ satisfies $\log P(\mathbf{Y}, \mathbf{X}, \mathbf{Z}, \mathbf{C}, \mathbf{S}, \mathbf{\Psi}) =$
$\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \mathbf{\Psi}) + \log P(\mathbf{X}|\mathbf{Z}, \mathbf{S}, \mathbf{\Psi}) + \log P(\mathbf{C}|\mathbf{\Psi}) + \log P(\mathbf{Z}|\mathbf{\Psi}) + \log P(\mathbf{S}|\mathbf{\Psi}) + \log P(\mathbf{\Psi})$,
where

$\log P(\mathbf{X}|\mathbf{Z}, \mathbf{S}, \mathbf{\Psi}) = \sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{s=0}^{1} \text{I}(Z_n = k)\text{I}(S_{kl} = s) logP(X_{nl}|g = s)$ We assume we have these probabilities given the genotype g being either mutation or non mutation.

$$\begin{aligned}
\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \mathbf{\Psi}) &= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \text{I}(Z_n = k)\text{I}(C_{km} = j) \log F_{\mu_{nj}, \sigma^2}(r_{nm}) \\
&= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \text{I}(Z_n = k)\text{I}(C_{km} = j)(-\frac{1}{2})\Big[\log(2\pi\sigma^2) + \frac{(r_{nm} - j\mu_n)^2}{\sigma^2}\Big],
\end{aligned} \tag{41}$$

$$\log P(\mathbf{C}|\mathbf{\Psi}) = \sum_{k=1}^{K}\left[\sum_{j=1}^{J} \text{I}(C_{k1} = j) \log \rho_{kj} + \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J} \text{I}(C_{km-1} = i, C_{km} = j) \log A_{ij}^k\right], \tag{42}$$

Note that ρ_{kj} is the initial probability in the MC which we fix.

$$\log P(\mathbf{Z}|\mathbf{\Psi}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \text{I}(Z_n = k) \log \pi_k, \tag{43}$$

and

$$\log P(\mathbf{S}|\mathbf{\Psi}) \quad = \quad \sum_{l=1}^{L}\sum_{k=1}^{K} \mathrm{I}(S_{kl}=0)\log\xi_k + \mathrm{I}(S_{kl}=1)\log(1-\xi_k), \tag{44}$$

Moreover,

$$\log P(\mathbf{\Psi}) = \log P(\mathbf{A}) + \log P(\boldsymbol{\mu}) + \log P(\frac{1}{\sigma^2}) + \log P(\boldsymbol{\pi}) + \log P(\boldsymbol{\xi}), \tag{45}$$

Defining, $\mathcal{B}$ the multivariate Beta function, i.e., $\mathcal{B}(\boldsymbol{x}) = \frac{\prod_{k=1}^{K}\Gamma(x_k)}{\Gamma(\sum_{k=1}^{K} x_k)}$, and $\mathcal{C}\ (v,u) = \frac{\Gamma(v)\Gamma(u)}{\Gamma(v,u)}$, we have:

$$\log P(\boldsymbol{\mu}) \quad = \quad \sum_{n=1}^{N}\big[(-\frac{1}{2})\Big[\log(2\pi\tau^2) + \frac{(\mu_n-\theta)^2}{\tau^2}\Big]\big], \tag{46}$$

$$\log P(\frac{1}{\sigma^2}) \quad = \quad \big[-\frac{\beta}{\sigma^2} + (\alpha-1)\log(\frac{1}{\sigma^2})\big] + C, \tag{47}$$

$$\log P(\mathbf{A}) \quad = \quad \sum_{k=1}^{K}\sum_{i=1}^{J}\big[\sum_{j=1}^{J}(\Lambda_j-1)\log A_{ij}^{k} - \log\mathcal{B}(\mathbf{\Lambda})\big], \tag{48}$$

$$\log P(\boldsymbol{\xi}) = \sum_{k=1}^{K}(\eta_k-1)\log\xi_k + (\gamma_k-1)\log(1-\xi_k) - \mathcal{C}(\eta_k,\gamma_k), \tag{49}$$

and

$$\log P(\boldsymbol{\pi}) \quad = \quad \sum_{k=1}^{K}(\delta_k-1)\log\pi_k - \log\mathcal{B}(\boldsymbol{\delta}). \tag{50}$$

Step 3. VD computation by coordinate ascent We now derive a coordinate ascent algorithm for the VD. That is, we derive an update equation for each term in the factorization (in step 1) by calculating the expectation of $\log P(\mathbf{Y},\mathbf{X},\mathbf{Z},\mathbf{C},\mathbf{S},\mathbf{\Psi})$ over the VD of all random variables except the one of interest. Below, the update equation is derived for each random variable. For convenience, we use $_{+}\!\approx$ to denote equality up to a constant additive factor. Note that $\mathbf{X}$ can have missing values at certain sites. Therefore, including this into the model, results in splitting $\mathbf{X}$ into $\mathbf{X_{miss}}$ and $\mathbf{X_{obs}}$. Moreover, we would need a Bernoulli variable, ω, which shows which of the two group is the case in the joint distribution. The joint distribution then takes the following form.

P(Y, $\mathrm{X}_{obs}, \mathbf{X_{miss}}, \omega, \mathbf{S}, \mathbf{Z}, \mathbf{C}, \mathbf{\Psi}$) =
$P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi})P(\mathbf{X_{obs}}|\mathbf{S},\mathbf{Z},\mathbf{\Psi})P(\mathbf{X_{miss}}|\mathbf{S},\mathbf{Z},\mathbf{\Psi})P(\mathbf{C}|\mathbf{\Psi})P(\mathbf{Z}|\mathbf{\Psi})P(\mathbf{S}|\mathbf{\Psi})P(\mathbf{\Psi})MCARP(\omega|\mathbf{Y},\mathbf{X_{obs}},\mathbf{X_{miss}},\mathbf{S},\mathbf{Z},\mathbf{C},\mathbf{\Psi})$

In the above, MCAR is referring to missing completely at random which is our assumption here, hence its probability is independent from all variational parameters. This means that this probability is a constant w.r.t. the variational parameters. We assume that $P(\mathbf{X_{miss}}|\omega,\mathbf{S},\mathbf{Z},\mathbf{\Psi})$ is constant and, therefore, consider $X = X_obs$ in our applications. However, for the ease of notation, in our derivations we assume X is complete. Namely, we can, for the sake of readability, simplify the joint distribution as $P(\mathbf{Y},\mathbf{X},\mathbf{S},\mathbf{Z},\mathbf{C},\mathbf{\Psi})$, where $\mathbf{X}$ refers to $\mathbf{X_{obs}}$.

Update equation for π: The update equation $q(\boldsymbol{\pi})$ can be derived as follows.

$$\begin{aligned}
&\log q(\boldsymbol{\pi}) \\
&= \quad \mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\mathbf{Y},\mathbf{X},\mathbf{Z},\mathbf{C},\mathbf{S},\mathbf{\Psi})\big) \\
&_{+}\!\approx \quad \mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\mathbf{Z}|\mathbf{\Psi})\big) \ + \ \mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\boldsymbol{\pi})\big) \\
&= \quad \sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\log\pi_k + \log P(\boldsymbol{\pi})
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{k=1}^{K} \log \pi_k \Big[\sum_{n=1}^{N} \mathrm{E}_{q(Z_n)} \big(\mathrm{I}(Z_n = k) \big) \Big] + \sum_{k=1}^{K} \log \pi_k (\delta_k - 1) \\
&= \sum_{k=1}^{K} \log \pi_k \Big[\Big(\sum_{n=1}^{N} \mathrm{E}_{q(Z_n)} \big(\mathrm{I}(Z_n = k) \big) + \delta_k \Big) - 1 \Big].
\end{aligned}
$$

Therefore, $q(\boldsymbol{\pi})$ is a Dirichlet distribution with parameters $\delta^* = (\delta_1^*, \ldots, \delta_K^*)$, where

$$
\delta_k^* = \delta_k + \sum_{n=1}^{N} \mathrm{E}_{q(Z_n)} \big(\mathrm{I}(Z_n = k) \big). \tag{51}
$$

Update equation for Z_n: The update equation $q(Z_n)$ can be obtained as follows.

$$
\begin{aligned}
&\log q(Z_n) \\
&= \mathrm{E}_{-Z_n} \big(\log P(\mathbf{Y}, \mathbf{X}, \mathbf{S}, \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) \big) \\
&\overset{+}{\approx} \mathrm{E}_{-Z_n} \big(\log P(\mathbf{Y} | \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) \big) + \mathrm{E}_{-Z_n} \big(\log P(\mathbf{X} | \mathbf{Z}, \mathbf{S}, \boldsymbol{\Psi}) \big) + \mathrm{E}_{-Z_n} \big(\log P(\mathbf{Z} | \boldsymbol{\Psi}) \big)
\end{aligned}
$$

Note that, each of the logarithms above can be written as the sum of two terms, one that depends on Z_n and one that does not; since we want to form a function of Z_n, we discard all other terms w.r.t. $Z_c : c \in [N] \setminus n$.
$\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{j=1}^{J} \mathrm{I}(Z_n = k) \mathrm{I}(C_{km} = j) (-\frac{1}{2}) \Big[\log(2\pi\sigma^2) + \frac{(r_{nm} - j\mu_n)^2}{\sigma^2} \Big]$
$\log P(\mathbf{X}|\mathbf{Z}, \mathbf{S}, \boldsymbol{\Psi}) \sum_{l=1}^{L} \sum_{k=1}^{K} \sum_{s=0}^{1} \mathrm{I}(Z_n = k) \mathrm{I}(S_{kl} = s) \log P(X_{nl} | g = s)$ Taking the expectation, we have:
$\mathrm{E}_{-Z_n} (\log P(\mathbf{X}|\mathbf{Z}, \mathbf{S}, \boldsymbol{\Psi})) = \sum_{l=1}^{L} \sum_{k=1}^{K} \mathrm{I}(Z_n = k) \Big(\sum_{s=0}^{1} \mathrm{E} \Big\{ \mathrm{I}(S_{kl} = s) \Big\} \log P(X_{nl} | g = s) \Big\} \Big)$

$$
\log P(\mathbf{Z}|\boldsymbol{\Psi}) \sum_{k=1}^{K} \mathrm{I}(Z_n = k) \log \pi_k
$$

Combining the above parts from $\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi})$ and $\log P(\mathbf{Z}|\boldsymbol{\Psi})$, we obtain:

$$
Q_Y = \sum_{k=1}^{K} \mathrm{I}(Z_n = k) \Big\{ \mathrm{E}_{q(\boldsymbol{\pi})} (\log \pi_k) + \sum_{m=1}^{M} \sum_{j=1}^{J} \mathrm{E}_{q(\mathbf{C}_k)} \big(\mathrm{I}(C_{km} = j) \big) \mathrm{E}_{q(\frac{1}{\sigma^2}) q(\mu_n)} [D_{nmj}] \Big\} \tag{52}
$$

Where:

$$
D_{nmj} = -\frac{1}{2} \Big[\log(2\pi\sigma^2) + \frac{(r_{nm} - j\mu_n)^2}{\sigma^2} \Big]. \tag{53}
$$

$\log q(Z_n) \overset{+}{\approx} Q_Y + \sum_{l=1}^{L} \sum_{k=1}^{K} \mathrm{I}(Z_n = k) \Big(\sum_{s=0}^{1} \mathrm{E} \Big\{ \mathrm{I}(S_{kl} = s) \Big\} \log P(X_{nl} | g = s) \Big)$
Reformulating $\log q(Z_n)$, we achieve the below. Note that when processing the input data, for the missing values we merely calculate Q_Y.

$$
\begin{aligned}
&\log q(Z_n) \overset{+}{\approx} \sum_{k=1}^{K} \mathrm{I}(Z_n = k) \sum_{l=1}^{L} \Bigg[\sum_{s=0}^{1} \mathrm{E}_{q(S_k)} (\mathrm{I}(S_{kl} = s)) \log P(X_{nl} | g = s) \\
&+ \mathrm{E}_{q(\boldsymbol{\pi})} (\log \pi_k) + \sum_{m=1}^{M} \sum_{j=1}^{J} \mathrm{E}_{q(\mathbf{C}_k)} \big(\mathrm{I}(C_{km} = j) \big) \mathrm{E}_{q(\frac{1}{\sigma^2}) q(\mu)} [D_{nmj}] \Bigg]
\end{aligned}
$$

We conclude that $q(Z_n) \sim \mathrm{Categorical}(\boldsymbol{p}_n)$ with parameters $\boldsymbol{p}_n = (p_{n1}, \ldots, p_{nK})$, where $\tilde{p}_{nk} = \sum_{l=1}^{L} \sum_{s=0}^{1} \mathrm{E}_{q(S_k)} (\mathrm{I}(S_{kl} = s)) \log P(X_{nl} | g = s) +$
$\mathrm{E}_{q(\boldsymbol{\pi})} (\log \pi_k) + \sum_{m=1}^{M} \sum_{j=1}^{J} \mathrm{E}_{q(\mathbf{C}_k)} \big(\mathrm{I}(C_{km} = j) \big) \mathrm{E}_{q(\frac{1}{\sigma^2}) q(\mu_n)} [D_{nmj}]$

and

$$p_{nk} = \frac{\exp(\tilde{p}_{nk})}{\sum_{k'=1}^{K} \exp(\tilde{p}_{nk'})}. \tag{54}$$

Update equation for ξ_k: We calculate $q(\xi_k)$ as follows.

$$\begin{aligned} &\log q(\xi_k) \\ &= \mathrm{E}_{-\xi_k}\big(\log P(\mathbf{Y}, \mathbf{X}, \mathbf{S}, \mathbf{Z}, \mathbf{C}, \mathbf{\Psi})\big) \\ &\overset{+}{\approx} \mathrm{E}_{-\xi_k}\big(\log P(\mathbf{S}|\mathbf{\Psi})\big) + \mathrm{E}_{-\xi_k}\big(\log P(\boldsymbol{\xi})\big) \end{aligned}$$

Disregarding the terms that do not depend on ξ_k in $\log P(\mathbf{S}|\mathbf{\Psi})$ and $\log P(\boldsymbol{\xi})$, we obtain:

$$\begin{aligned} \log q(\xi_k) \quad &\overset{+}{\approx} \quad (\eta^0 - 1)\log \xi_k + (\gamma^0 - 1)\log(1-\xi_k) + \sum_{l=1}^{L} \mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl}=0))\log \xi_k + \\ &\qquad\qquad \mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl}=1))\log(1-\xi_k) \\ &= \quad \log \xi_k \big\{\eta^0 - 1 + \sum_{l=1}^{L} \mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl}=0))\big\} + \log(1-\xi_k)\big\{\gamma^0 - 1 + \sum_{l=1}^{L} \mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl}=1))\big\} \end{aligned} \tag{55}$$

Therefore, $q(\xi_k)$ is distributed by Beta with parameters:

$$\eta_k^* = \eta^0 - 1 + \sum_{l=1}^{L} \mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl}=0)) \tag{56}$$

$$\gamma_k^* = \gamma^0 - 1 + \sum_{l=1}^{L} \mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl}=1)) \tag{57}$$

Update equation for S_{kl}: We calculate $q(S_{kl})$ as follows.

$$\begin{aligned} &\log q(S_{kl}) \\ &= \mathrm{E}_{-S_{kl}}\big(\log P(\mathbf{Y}, \mathbf{X}, \mathbf{S}, \mathbf{Z}, \mathbf{C}, \mathbf{\Psi})\big) \\ &\overset{+}{\approx} \mathrm{E}_{-S_{kl}}\big(\log P(\mathbf{X}|\mathbf{Z}, \mathbf{S}, \mathbf{\Psi})\big) + \mathrm{E}_{-S_{kl}}\big(\log P(\mathbf{S}|\mathbf{\Psi})\big) \end{aligned}$$

Disregarding the terms that do not depend on S_{kl} in the above, we obtain:

$$\begin{aligned} \log q(S_{kl}) \quad &\overset{+}{\approx} \quad \mathrm{I}(S_{kl}=0)\mathrm{E}_{q(\xi_k)}(\log \xi_k) + \mathrm{I}(S_{kl}=1)\mathrm{E}_{q(\xi_k)}(\log(1-\xi_k)) + \\ &\qquad \sum_{n=1}^{N}\sum_{s=0}^{1} \mathrm{E}_{q(Z_n)}(\mathrm{I}(Z_n=k))\mathrm{I}(S_{kl}=s)\log P(X_{nl}|g=s) \\ &\overset{+}{\approx} \quad \mathcal{A} + \sum_{n=1}^{N} \mathcal{B} \end{aligned} \tag{58}$$

where $\mathcal{A} = \mathrm{I}(S_{kl}=0)\mathrm{E}_{q(\xi_k)}(\log \xi_k) + \mathrm{I}(S_{kl}=1)\mathrm{E}_{q(\xi_k)}(\log(1-\xi_k))$, and $\mathcal{B} = \sum_{s=0}^{1} \mathrm{E}_{q(Z_n)}(\mathrm{I}(Z_n=k))\mathrm{I}(S_{kl}=s)\log P(X_{nl}|g=s)$.
Therefore, $q(S_{kl})$ is distributed by Bernoulli up to an additive constant. The Bernoulli parameter, corresponding to $S_{kl}=1$ is as the following. $\hat{\Sigma_{kl}} = \mathrm{E}_{q(\xi_k)}(\log(1-\xi_k)) + \sum_{n=1}^{N} \mathrm{E}_{q(Z_n)}(\mathrm{I}(Z_n=k))\log P(X_{nl}|g=1)$

$$\Sigma_{kl} = \frac{\exp(\hat{\Sigma_{kl}})}{\sum_{k=1}^{K} \exp(\hat{\Sigma_{kl}})} \tag{59}$$

The rest of update equations are the same as those in CopyMix.

The new expectations that are used in the above update equations are the following.

$\mathrm{E}_{q(\xi_k)(\log \xi_k)} = \mathbf{\Psi}(\eta_k^*) - \mathbf{\Psi}(\eta_k^* + \gamma_k^*)$

$\mathrm{E}_{q(\xi_k)(\log(1-\xi_k))} = \mathbf{\Psi}(\gamma_k^*) - \mathbf{\Psi}(\gamma_k^* + \eta^*)$

A.8 Simulation procedure

Here, we provide the details of the simulations. Because there is no simulation framework available, we generated data using the most reasonable way; that is, we assume a diploid cell (normal cell) can be affected by various copy number events (comprising duplications and deletions). We use an HMM, a generative model with latent copy number states, to generate the observed read ratios. HMM is a well-known model for copy numbers; see HMMCopy Shah et al. (2006) for more details.

For the Guassian emission, the generated configurations (CONF 1 to 18) are shown in the main paper. We modulate the read ratios by introducing subtle noise. Regarding the number of clusters, each of CONF 1 to CONF 5 contains two clusters, CONF 6 to CONF 10 three clusters, CONF 11 to CONF 13 four clusters, and CONF 14 to CONF 18 five clusters. The copy number transition patterns are formed using the following copy number state patterns, which are inspired by the CNV, along with single-cells that are obtained using the DLP technology Laks and McPherson (2019).

- **single state**: a copy number is inclined to remain at one certain state (CONF 1, CONF 2, CONF 3, CONF 4, CONF 5, CONF 12, and CONF 13 include this pattern);
- **inertial state**: a copy number is inclined to remain unchanged (The blue cluster in CONF 6 to CONF 10 includes this pattern);
- **oscillating state**: a copy number fluctuates between two states (The green cluster in CONF 6 and CONF 9 include this pattern);
- **altered state**: a sudden deletion- or duplication-like event of copy number occurs (all configurations include this pattern);
- **scaled state**: for specific ranges of positions in the copy number sequence, copy numbers are increased or decreased by a multiplicative or additive constant (all configurations include this pattern except CONF 3, CONF 4, CONF 6, CONF 11, and CONF 13);
- **whole-genome duplication**: all copy numbers across the genome are scaled (CONF 5 includes this pattern).

For both emissions, for each simulation, the following steps are performed.

1. Set the random seed.
2. Generate cells belonging to clusters by sampling from a multinomial distribution given a vector of cluster-assignment probabilities.
3. Generate rates for all cells. Rates are sampled from a Gaussian distribution with a mean of 10 and a standard deviation of 1.
4. Distribute the rates among the cells for different clusters; this is done based on step 2.
5. Set values to the number of HMM's hidden states, transition matrix of each cluster, and sequence length.
6. Generate a Gaussian HMM for each cluster using cell rates of that cluster, number of hidden states, transition matrix of the cluster, and sequence length; this results in one copy number hidden sequence and different cell ratios emitted from that copy number sequence (the copy number is multiplied by the rate, representing the emission's mean).
7. Accumulate the cells from the previous step and insert them into a dataset. Similarly, their cluster labels and the hidden copy number sequences are stored into datasets.

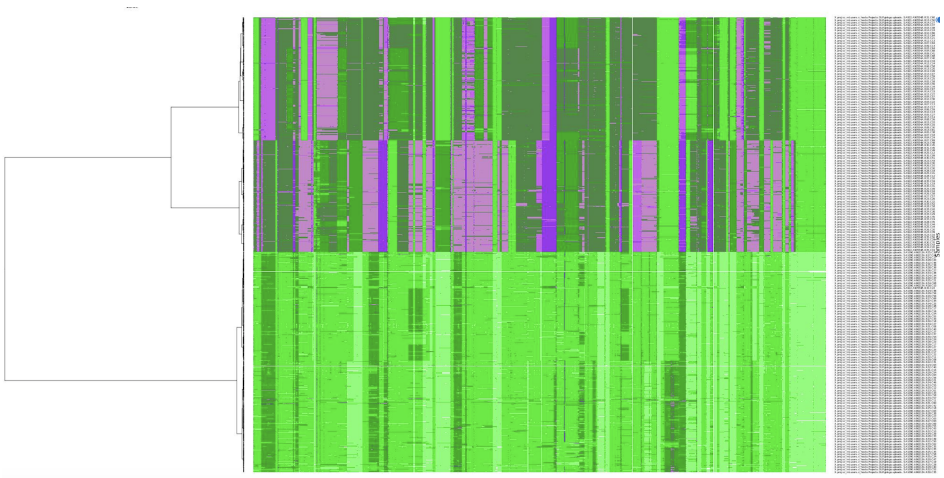

Figure 10: The heat map and the dendrogram show three major clusters detected by Ginkgo. The copy number is color-coded by a scale from green (small) to magenta (large).

A.9 Ginkgo results

In this section, we report the results obtained by Ginkgo.

Ginkgo's hierarchical clustering, after deleting 80% of the cells, detects three main clusters; firstly, the question is how the clustering would be if they didn't exclude so many cells. Secondly, we cannot state that hierarchical clustering detects more than those three clusters. That is because the branch lengths of the smaller clusters, in the hierarchical clustering, are negligible in size compared to the branch length of the main three clusters. To include these branches and, consequently, the smaller clusters, one must choose a very shallow threshold when cutting the dendrogram produced by the hierarchical clustering.

If we, based on shared events in dark green in Fig. 10, assign a threshold such that we include the two subgroups depicted in the bottom and green part of Fig. 10, then Ginkgo detects four clusters. Regarding the four clusters, CopyMix outperforms Ginkgo with V-measure 67% compared to V-measure 55%. The remaining Ginkgo subgroups containing magenta-colored copy numbers require a lower threshold than those for the four clusters. Finally, it is essential to mention that Ginkgo's web application, one advantage of using Ginkgo Mallory et al. (2020b), could not handle the large DLP data. To run Ginkgo, we developed a bash tool that assembled the back-end parts of the Ginkgo into a new script.

A.10 Choice of Gaussian emission

This section discusses why Gaussian is chosen as a distribution for the read ratios. Following the principle of maximum entropy, we choose Gaussian because it has maximum entropy for a specified mean and variance Guiasu and Shenitzer (1985). Also, the histogram plot of DLP data (Fig. 11) confirms a skew-normal distribution. Finally, it is common that the data is assumed to follow Gaussian distribution; see HMMcopy and another HMM-based approach Shah et al. (2006); et al (2008).

A.11 SNV data

In this section, we illustrate the SNV data, confirming too shallow data signals for further cluster detection. As shown in Fig. 12, one can observe that the cells, which are on the Y axis, can be clustered into three main groups. Note that the three plots show in total 15000 sites where SNVs are detected. Performing K-means and hierarchical

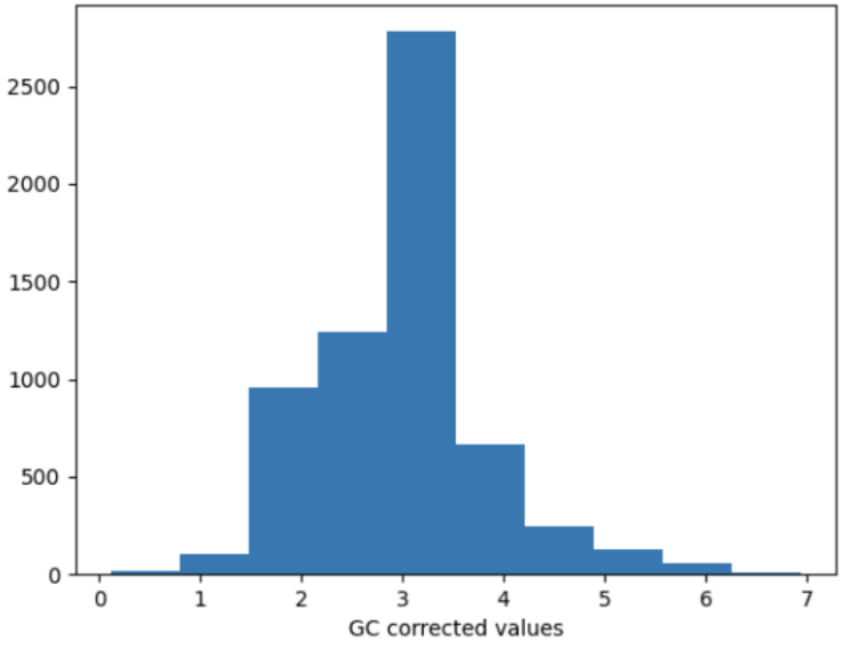

Figure 11: DLP data histogram plot.

clustering resulted in three clusters; hence, it is reasonable that this data has been too shallow to reveal more clusters.

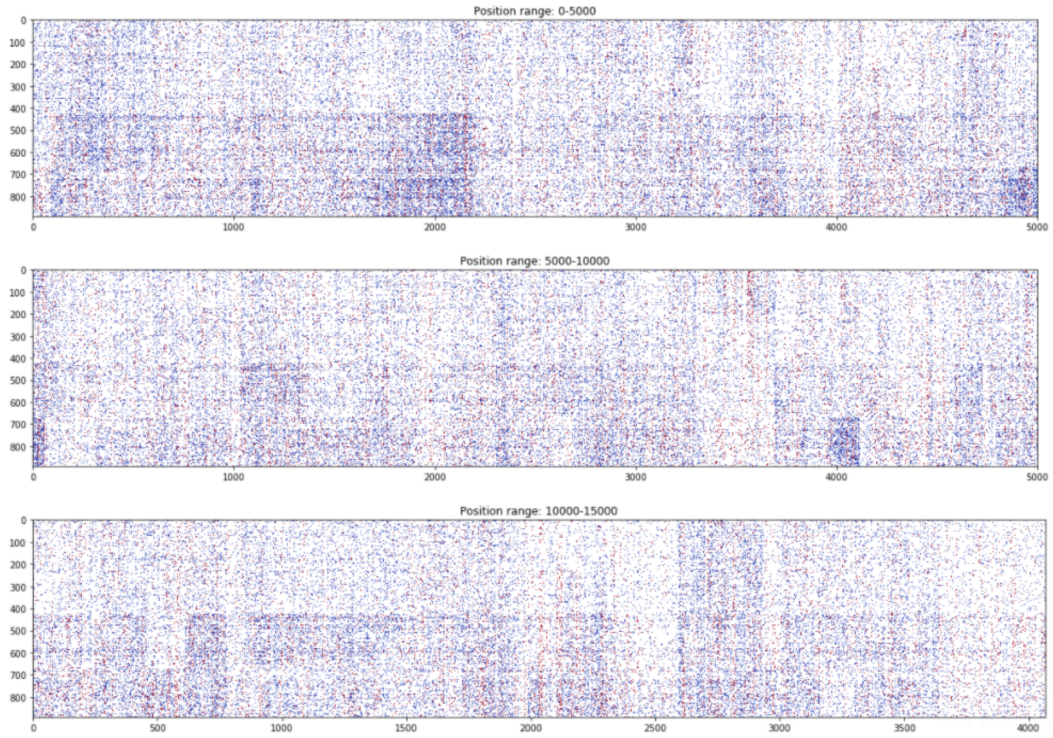

Figure 12: SNV counts are illustrated as colors in the heatmap for 891 cells (Y axis) across the genomic positions (X axis).