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

Negar Safinianaini[1*], Camila P. E. de Souza[2], 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

*Correspondence: negar7918@gmail.com

Abstract

Single-cell sequencing technologies are becoming increasingly more established, in particular, in the study of tumor heterogeneity. Investigating tumor heterogeneity is imperative to better understand how tumors evolve since each cell subpopulation harbors a unique set of genomic features that yields a unique phenotype, an issue that is bound to have clinical relevance. Clustering of cells based on copy number data, obtained from single-cell DNA sequencing, provides an opportunity to assess different tumor cell subpopulations. Accordingly, computational methods have emerged for single-cell copy number profiling as well as clustering; however, these two tasks have up to now been handled sequentially, and, included various ad-hoc preprocessing steps; hence, prone to introduce clustering artifacts. We propose CopyMix, a mixture model-based method using Variational Inference, to simultaneously cluster cells and infer a single copy number profile for each cluster, revealing the underlying copy number variation pattern for each cluster. CopyMix is evaluated using simulated data as well as 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 a new opportunity to investigate the genomic profile of individual cells both regarding single nucleotide variation (SNV) and copy number variation (CNV). CNVs and SNVs have been shown to be important contributors to phenotypic variation relating to health and disease Baslan et al. (2012); Lawson et al. (2018). Moreover, 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 of

cells based on their individual copy number profiles provides the opportunity for SNV profiling of the clusters and, in general, opportunities to better understand tumor subpopulations and tumor heterogeneity, issues that are 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 as well as clustering of cells based on their already inferred copy number profiles, e.g., Garvin et al. (2015); Zahn et al. (2017); Leung et al. (2017); Vitak et al. (2017); however, these two tasks have up to now been handled sequentially with various ad-hoc preprocessing steps. The sequential approach is vulnerable to clustering artifacts since preprocessing decisions constrain all later analyses and are typically irreversible. Even if each task has maximized its performance, the final result may still fall short of the best possible in the traditional single-phase inference framework Blocker and Meng (2013). Regarding copy number profiling, HMMcopy Shah et al. (2006)—a widely used Hidden-Markov-Model-based method for copy number profiling—has limitations such as requiring manual calibration of more than ten parameters Mallory et al. (2020).

As a one-shot mathematically stable solution, also motivated by future directions of computational modelling of single-cell cancer genomics Zhang and Campbell (2020), to 1) the sequential treatment of copy number profiling and clustering, and 2) the labor of HMMcopy parameter tuning we propose a novel framework, referred to as CopyMix. CopyMix performs the copy number profiling and clustering simultaneously, and as to the best of our knowledge, no earlier work has simultaneously performed the two tasks. Similar to CopyMix, simultaneous inference have been considered but 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). CopyMix is a model-based approach enjoying the advantages of a fully probabilistic framework such as transparency, uncertainty measurements, and modeling flexibility. CopyMix uses a mixture model with components, corresponding to clusters, each having a specific copy number profile modeled by a sequence of latent variables which follows a Markov structure similar to the state-of-the-art methods using Hidden Markov Models (HMMs) Shah et al. (2006); Vitak et al. (2017); Zahn et al. (2017), 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 than inference methods 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 performing principled determination of the optimal number of mixture components Bishop (2006).

We perform experiments on biologically inspired simulated data. Moreover, we conduct experiments on a published biological data from ovarian cancer Laks and McPherson (2019). We evaluate CopyMix on the ovarian cancer data and compare our results to those obtained by the original paper Laks and McPherson (2019), where HMMcopy is used as a copy number profiling tool.

We, first, show the CopyMix model and experimental results for our earlier work when the emission was assumed to be Poisson. Then, we show our results concerning the Gaussian emission.

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

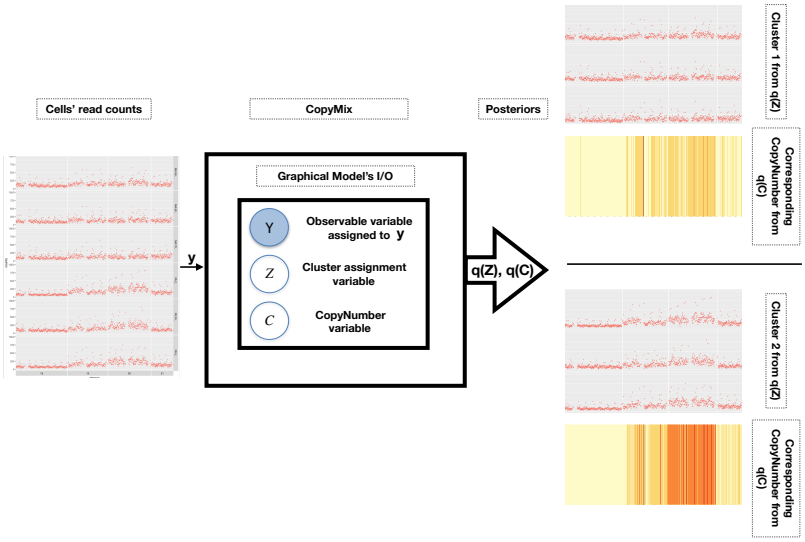

Figure 1: The overview of CopyMix with: input y, cells' reads; outputs $q(Z)$ and $q(C)$, posterior distributions resulting in clusters within y with their corresponding copy number profiles illustrated by heatmaps (this is an example of binary clustering).

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 model for CopyMix. 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 modelling.

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,

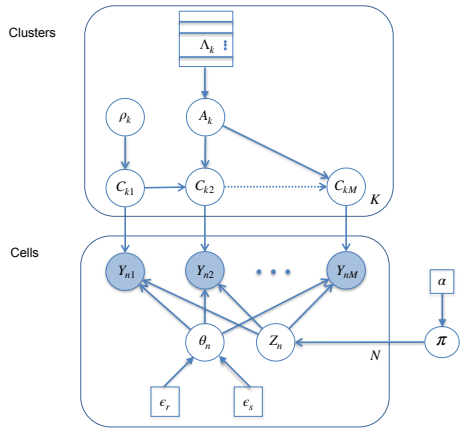

Figure 2: Probabilistic graphical model representing our proposed model. Shaded nodes represent observed values, the unshaded ones are the latent variables and the squares are the hyperparameters of the model; a posterior distribution over the values of the unshaded nodes is approximated using Variational 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.

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, 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 $\boldsymbol{\Psi}$ be the set containing all the unknown model parameters, i.e., $\boldsymbol{\Psi} = \{\mathbf{A}, \boldsymbol{\theta}, \boldsymbol{\pi}\}$, 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 $\boldsymbol{\Psi}$, 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), $q(\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi})$, with smallest Kullback-Leibler divergence to the posterior distribution $P(\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}|\mathbf{Y})$, which is equivalent to maximizing the evidence lower bound (ELBO) Blei et al. (2017) given by

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

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

- $\mathbf{a}_i^k = (a_{i1}^k, \ldots, a_{iJ}^k) \sim \text{Dirichlet}(\boldsymbol{\Lambda}_i^{k0})$.
- $\theta_n \sim \text{Gamma}(\epsilon_{sn}^0, \epsilon_{rn}^0)$, 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 $\boldsymbol{\Psi}$, are described below.

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

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

Step 2. Joint distribution

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

procedure ESTIMATE-POSTERIORS(Y): ▷ Y input is N sequences of length M
Assign priors hyperparameters
Initialise variational distributions parameters
  repeat
    Update variational parameters (using **Step 4. Summary of updates**)
  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

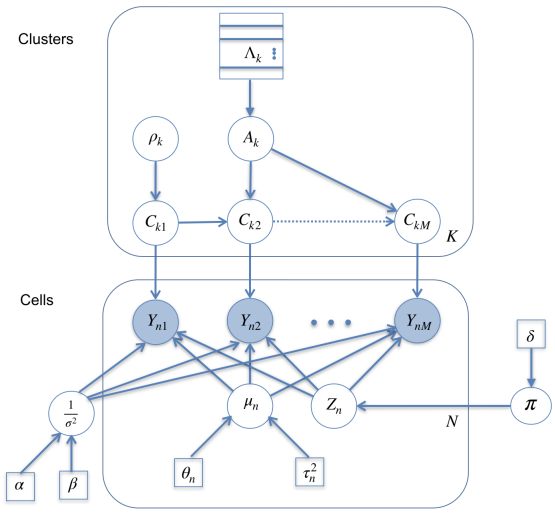

Figure 3: In this version Y_{nm} are observed corrected read ratios from cell n and bin m; C_{km}, corresponding latent copy number state forming a Markov chain; μ_n, cell-specific rate; Z_n, latent cell-specific cluster assignment variable. π and A_k, the priors over Z_n and C_{km} respectively; ρ_k,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

2.2 Gaussian emission

In this version, we assume that the read counts are GC-corrected (a bias correction performed in advance), which means that the data values are real numbers. Thus, it is meaningful to assume that the data follow a Gaussian distribution; again, we assume fixed equal size genomic bin as in the Poisson version. The rest of the model is similar to the previous section. The new probabilistic graphical model is illustrated in Fig. 3, where we model the average sequencing coverage as Gaussian with independent conjugate priors; this means that the mean of the Gaussian follows itself a Gaussian distribution and the precision of the Gaussian follows a Gamma distribution.

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 $j\mu_n = j \times \mu_n$ (we add noise .001 to j=0 for computational purposes), i.e.,

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

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 $\mathbf{\Psi}$.

- $\mathbf{a}_i^k = (a_{i1}^k, \ldots, a_{iJ}^k) \sim \text{Dirichlet}(\mathbf{\Lambda})$
- $\mu_n \sim \text{Normal}(\theta_n, \tau_n^2)$. 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.
- $\boldsymbol{\pi} \sim \text{Dirichlet}(\boldsymbol{\delta})$

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:

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

Step 2. Joint distribution

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

In this section, we briefly describe the metrics used subsequently in the Empirical Investigation section. CH Index Calinski and Harabasz (1974), Silhouette Rousseeuw (1987) and likelihood ratio test are the three metrics we use to evaluate CopyMix clustering results. Total variation Sirazdinov and Mamatov (1962) is the metric that we use to evaluate CopyMix copy number inference results.

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 measure 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 measure 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 a p-value; a high p-value indicates that the null hypothesis is preferred over the winning hypothesis.

One standard distance measure for comparing probabilities (proportions) is *total variation*, calculated as in equation (7) with two sequences C, C', where C_m and C'_m are copy number proportions—the proportions are obtained by normalizing a sequence of copy numbers across the sequence—at position m in the sequences. Total variation is recently used to measure the distance between copy number proportions Zaccaria and Raphael (2020).

$$TV(C, C') = \frac{1}{2} \sum_{m \in M} |C_m - C'_m| \tag{7}$$

3 Empirical investigation

3.1 Experimental setup for CopyMix with Poisson emission

3.1.1 Dataset for CopyMix with Poisson emission

We perform experiments on both simulated data and two published biological datasets: (1) Count data from Leung et al. (2017) (available from the NCBI Sequence Read Archive (SRA) under accession number SRP074289); (2) Real value 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.

For the simulated data, 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 generation details, see README.txt in the **CopyMix**. We examine different test cases, orthogonal to each other, to show our approach's robustness.

3.1.2 Experimental Protocol for CopyMix with Poisson emission

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).

3.2 Experimental Results for CopyMix with Poisson emission

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.

3.2.1 Simulated data for CopyMix with Poisson emission

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

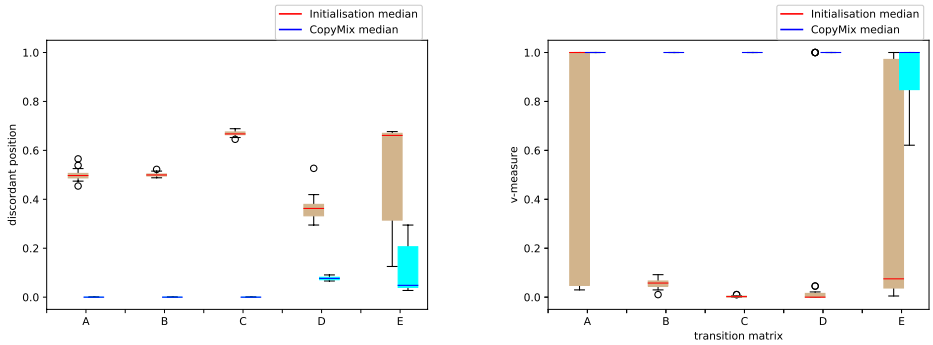

Figure 4: Performance of initialization methods (brown bars with red median values) and CopyMix (cyan bars with blue median values) across different sets of transition matrices. LHS: discordant positions between true and inferred vectors of copy number states. RHS: V-measure between true and inferred cell clustering assignments.

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 4, 5 and 6 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. 4) where V-measure is in an interval of [.85, 1] with the median being 1; proportions of discordant positions 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. 4 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.

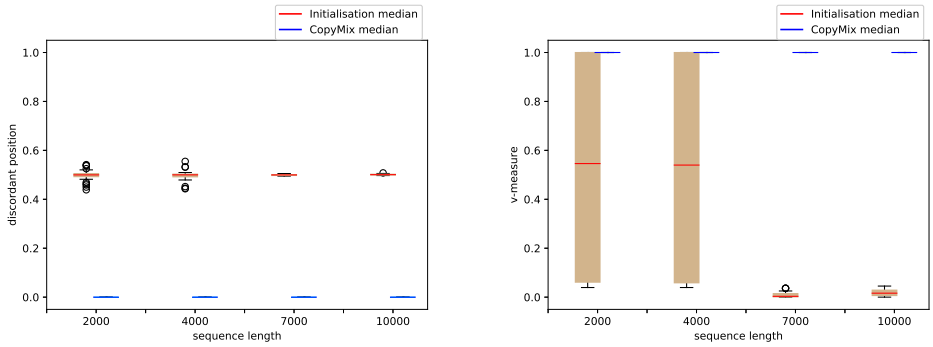

Figure 5: Performance of initialization methods and CopyMix, varying the sequence length. LHS and RHS figures, as well as color codes, are similar to Fig. 4.

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. 4, 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 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. 5. We can observe that our model handles these sequences without any decrease in performance. Finally, Fig. 6 illustrates the perfect performance of CopyMix when varying the number of cells from 15 to 500 when, again, using configuration **A**.

3.2.2 Biological data for CopyMix with Poisson emission

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.

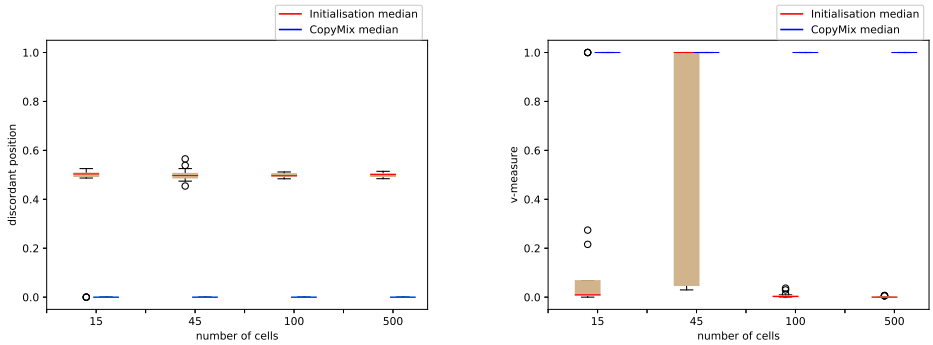

Figure 6: Performance of initialization methods and CopyMix varying the number of cells. LHS and RHS figures, as well as color codes, are similar to Fig. 4.

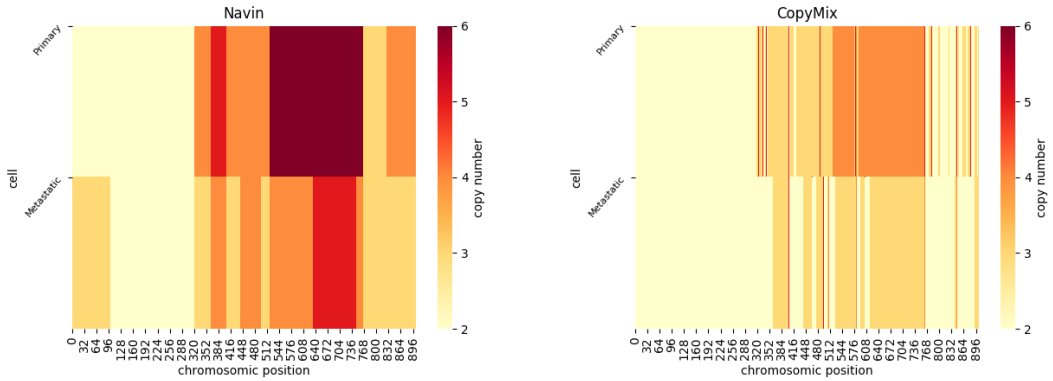

Figure 7: Copy number profiling comparison between the results from Navin Leung et al. (2017) (*LHS*) and CopyMix (*RHS*) for both primary and metastatic clusters.

(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. 7 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).

3.2.3 A sequential CopyMix

Applying a sequential version of CopyMix (1. performing copy number profiling, 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 setup for CopyMix with Gaussian emission

3.3.1 Dataset for CopyMix with Gaussian emission

We conduct experiments on both simulated data and a published biological dataset Laks and McPherson (2019). The biological dataset is obtained from https://zenodo.org/record/3445364.YegRmljMLzW.

The biological data, ovarian cancer single-cells, are generated from DLP+, a scalable single-cell whole-genome sequencing platform, using Direct Library Preparation (DLP) technology. In the rest of empirical investigation we refer to this data as DLP data. The DLP data are decomposed into three cancer cell lines derived from the same patient, sourced from one primary tumor and two relapse specimens. We refer to these three cell lines as *cell lines*. According to the original paper Laks and McPherson (2019), we conduct experiments on 891 cells which are produced after outlier removal.

For the simulated data, we generate 18 datasets, based on various series of copy number duplication and deletion events. We design these datasets such that each cluster corresponds to a unique CNV pattern, i.e., state transition pattern in the Markov chain. In order to adequately evaluate the robustness of CopyMix, we design various CNV patterns for each of the clustering scenarios spanning 2, 3, 4, and 5 total number of clusters. To this end, we generate 18 different configurations of simulated single cell clustering (CONF 1 to CONF 18). For all of the 18 configurations, we use 150 cells and 200 bins. We assume equal amplification on the read counts across bins since the bin size is large enough (500kb Laks and McPherson (2019)), meaning that the effect of unequal amplification is negligible. We modulate the read counts subtlety by introducing variance, representing noise in data, when generating cells from a Gaussian distribution. CONF 1 to CONF 6, illustrated in Fig. 8, are various binary clustering scenarios with different copy number state patterns. CONF 7 to CONF 10, illustrated in Fig. **??**, are various scenarios to differentiate three clusters. Similarly, CONF 10 to CONF 14, illustrated in Fig. **??**, are various scenarios to differentiate four clusters. Finally, CONF 15 to CONF 18 are scenarios to differentiate five clusters. The copy number transition patterns are formed using the following copy number state patterns which are inspired from the CNV along single cells that are obtained using the DLP technology Laks and McPherson (2019). For the details of the generation of these 18 configuration, see Supplementary information 4.

- **single state**: a copy number is inclined to remain at one certain state (all configurations include this pattern except CONF 6, CONF 10, CONF 11, and CONF 13);

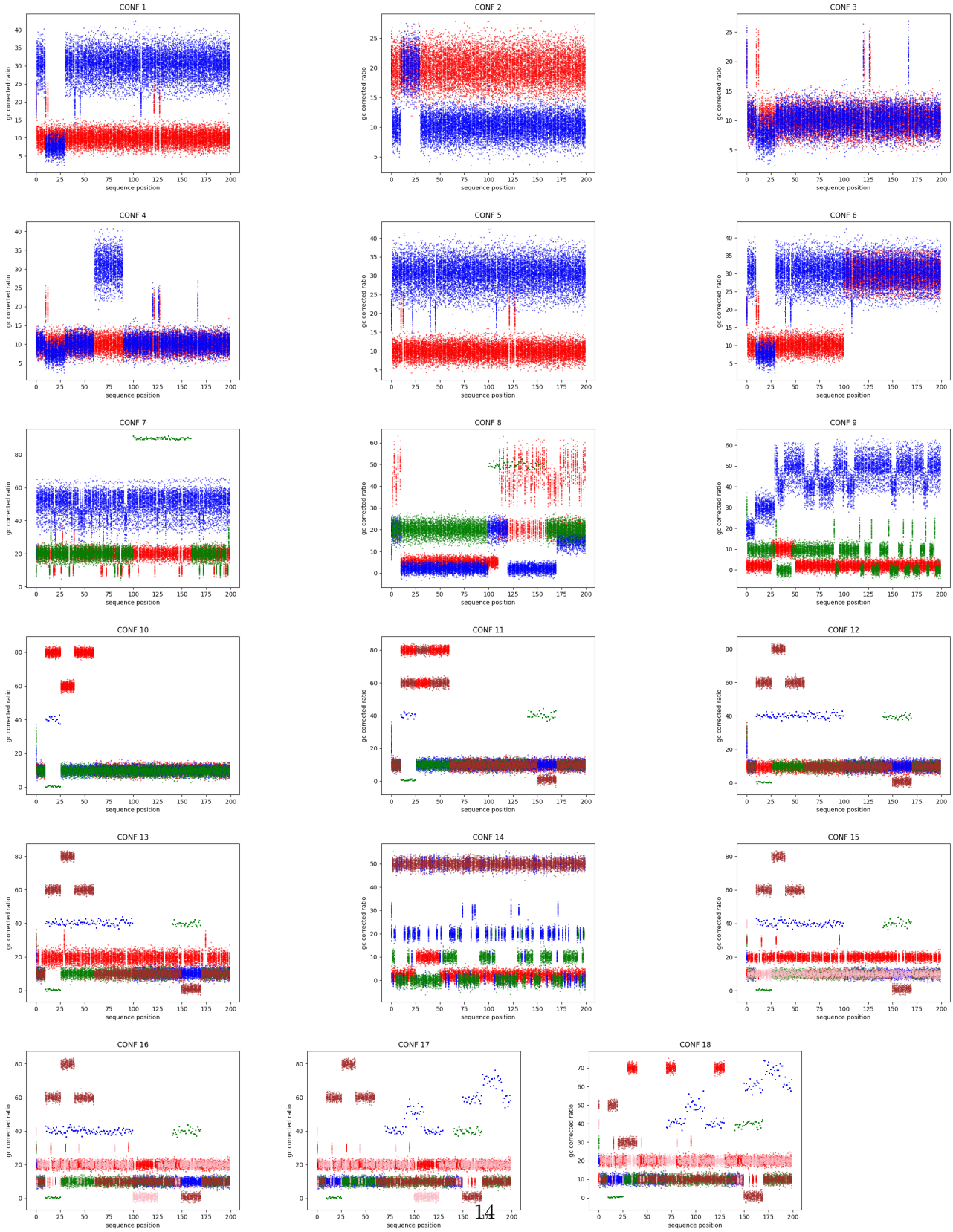

Figure 8: The datasets for CONF 1 to 18 are shown. CONF 1 to 6, CONF 7 to 10, CONF 11 to 14, and CONF 15 to 18 correspond to scenarios where 2, 3, 4, and 5 clusters should be distinguished, respectively.

- **inertial state**: a copy number is inclined to remain unchanged (The blue cluster in CONF 9 includes this pattern);
- **oscillating state**: a copy number fluctuates between two states (CONF 9 and CONF 14 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 10, CONF 11, and CONF 12);
- **whole genome duplication**: all copy numbers across the genome are scaled (CONF 5, CONF 7, CONF 14, and CONF 15 include this pattern, however, each with different scaling).

3.3.2 Experimental protocol for CopyMix with Gaussian emission

During all experiments, we assume we know the number of clusters in advance. 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.

We assess the clustering performance of CopyMix using a well accepted measure, V-measure Rosenberg and Hirschberg (2007). The V-measure quantifies to which extent a cluster only contains data points that are members of a single class; the worst clustering corresponds to V-measure 0% and the best clustering corresponds to V-measure 100%.

For simulated data, we compare CopyMix clustering results with the ground truth using V-measure. For biological data evaluation, V-measure is used to compare our inferred clusters to the ones detected by the original paper Laks and McPherson (2019). In addition to merely comparing to the clustering result in the original paper, we examine if the original paper has inferred meaningful clusters; we obtain this by calculating the CH Index Calinski and Harabasz (1974) and Silhouette Rousseeuw (1987) metrics on the raw data. Finally, we use likelihood ratio test to examine the statistical significance of the winning hypothesis.

Regarding the inference of copy number profiles, since CopyMix does not guarantee the scaling of copy number profiles we convert the them into proportions or probabilities; these are achieved by normalizing the copy number states. Using these normalized values, we can compare the distance between the inferred copy number proportions and the true copy number proportions. We compare these probabilities (proportions) by using total variation, previously shown in equation (7). We compare the copy number profiles between the results from CopyMix and the results from the original paper Laks and McPherson (2019) by plotting heat maps, where the copy numbers are color coded.

3.4 Experimental results for CopyMix with Gaussian emission

3.4.1 Simulated data for CopyMix with Gaussian emission

Here, we evaluated the performance of CopyMix on 18 simulate datasets based on the aforementioned 18 configurations, see Fig. 8. In Table 1, for each of the configurations, we report the V-measure of each

synthetic dataset. We can observe that CopyMix, except for CONF 5, has a good clustering performance, and reach V-measures ranging from 76% to 100%, Table 1. The exception, i.e., CONF 5, consists of two genomes where one is a whole genome duplication of the other, *a whole genome duplication pair*. Also other configurations contain a whole genome duplication pair, see the blue- and red-colored genomes in CONF 7, and the red- and pink-colored genomes in CONF 15 in Fig. 8. No other configuration contains a whole genome duplication, and for all except CONF 8, CopyMix obtained a V-measure ranging from 87% to 100%. In each of these configurations CONF 5, CONF 7, and CONF 15, our method did not achieve a good clustering due to the presence of the whole genome duplication. In each of these three configurations, all of the incorrectly clustered cells were associated with the configuration's whole genome duplication pair. The explanation for the increasing trend of V-measures for the configurations CONF 5, CONF 7, and CONF 15 is the increasing number of other genomes.

Another important observation from Table 1 is that Copymix is robust when 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, shown in equation (**??**), i.e., each bin for each cell has a mean equal to the multiplication of the cell specific rate and the underlying latent copy number at that bin. Namely, despite providing strength in differentiating clusters, the multiplicative nature of mean (copy number $\times$ cell rate) can lead to similar or equal valued means, e.g. $2 \times 4 = 1 \times 8$, where 2 and 1 are two copy numbers and 4 and 8 are two cell rates. It may raise a question on why the cell rates are not aligned with the copy number states. The answer is the *unidentifiability* Murphy (2012)—inability to derive the true labels in unsupervised learning—of transition matrix in a Markov chain. Due to this issue which appears in CONF 8, CONF 11, and CONF 14, some cluster pairs are estimated to be similar, hence the clusters are wrongly merged into one cluster, see Fig. 8.

Finally, the other source of error in our V-measure results is when the clusters are overlapping (CONF 16 and CONF 17), however, the definition of clustering implies distinguishing non-overlapping data; see how the cluster-overlapping in CONF 16 is improved in CONF 17 and became even less overlapped in CONF 18 in Fig. 8. The clustering improvements are obtained by decreasing the overlapping parts of the genome that is performed by addition of CNVs on the overlapping parts of the genomes.

Note that the V-measures of the initialization methods for all of the configurations are small values, revealing that VI results are not already achieved by the initialization methods.

Table 2 provides the total variations between the true copy number proportions and those inferred by CopyMix, previously calculated by equation (7). CopyMix reveals high quality results, i.e., the errors (total variation distance to the ground truth copy numbers) are all small values. However, we can observe that with increasing the number of clusters, slightly larger errors begin to manifest. Also, we notice that when the number of breakpoints in the copy number profiles are many (the red-colored genomes in CONF 8 and the blue-colored genomes in CONF 14) the error slightly increases. These small increase in the errors are expected since the possibility of differing copy numbers increases as the number of copy number breakpoints increases.

To conceptualize the total variation results, we illustrate an example: comparison of the inferred copy numbers for CONF 1 and CONF 4. As shown in Fig. 9, CopyMix copy numbers are estimated better for CONF 1 than for CONF 4. That is, the copy number deletion event in CONF 1 is detected, but the copy number deletion events between positions 10 and 30 in CONF 4 has not been detected. The superity of copy number detection of CONF 1 over CONF 4 can be confirmed by that the total variation of CONF 1 (0.097) is less than the total variation of CONF 4 (0.203).

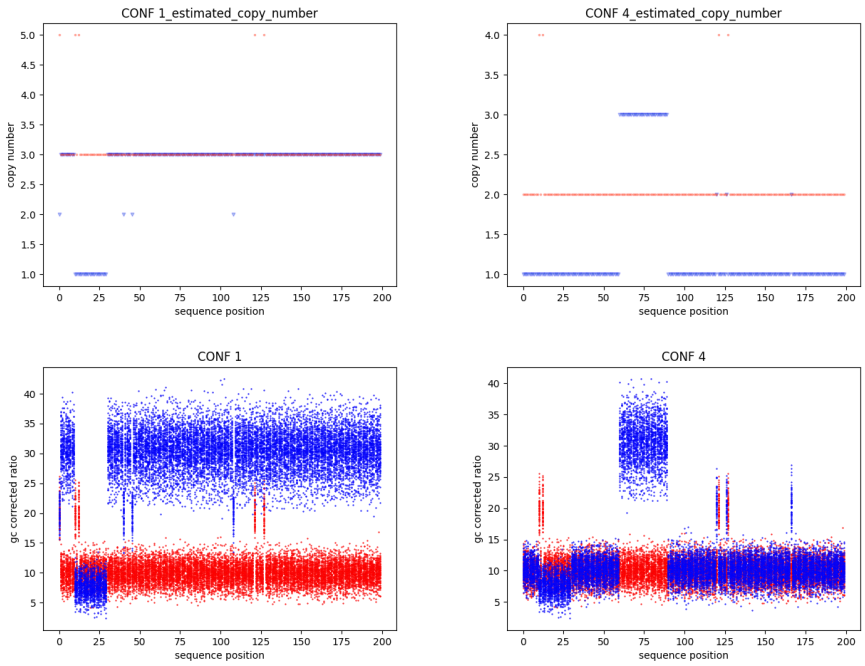

Figure 9: The LHS figures plot the estimated copy number and the corresponding dataset for CONF1, and the RHS figures plot those for CONF 4.

Table 1: V-measure (clustering result) of CopyMix for Conf 1 to Conf 18.

Number of Clusters	**Model**	$CONF1$	$CONF2$	$CONF3$	$CONF4$	$CONF5$	$CONF6$
2	CopyMix V-measure	**100%**	**100%**	**100%**	**100%**	0%	**100%**

Number of Clusters	**Model**	$CONF7$	$CONF8$	$CONF9$	$CONF10$
3	CopyMix V-measure	76%	76%	**100%**	**100%**

Number of Clusters	**Model**	$CONF11$	$CONF12$	$CONF13$	$CONF14$
4	CopyMix V-measure	87%	**100%**	**100%**	87%

Number of Clusters	**Model**	$CONF15$	$CONF16$	$CONF17$	$CONF18$
5	CopyMix V-measure	80%	88%%	91%	**100%**

Table 2: Total variation (TV) measuring the error of copy number inference obtained by CopyMix for Conf 1 to Conf 18.

Model	$CONF1$	$CONF2$	$CONF3$	$CONF4$	$CONF5$	$CONF6$	$CONF7$	$CONF8$	$CONF9$	$CONF10$	$CONF11$
CopyMix TV	0.097	0.100	0.057	0.203	0.113	0.414	0.920	0.651	2.409	0.650	0.996

Model	$CONF12$	$CONF13$	$CONF14$	$CONF15$	$CONF16$	$CONF17$	$CONF18$
CopyMix TV	1.104	1.114	3.541	0.691	1.352	1.609	1.022

3.4.2 Biological data for CopyMix with Gaussian emission

We evaluated CopyMix on the DLP data Laks and McPherson (2019), ovarian cancer single-cells each decomposed of 6206 bins. We initialized the number of clusters in CopyMix with 9, i.e., the number of detected clusters by the original paper Laks and McPherson (2019). However, CopyMix detects 4 clusters in total; the cell lines are recovered and one of the cell lines is clustered into two sub-clones. It follows from the cell lines that the DLP data should be clustered into three major clones. Therefore, we conclude that CopyMix succeeds in the clustering of the major clones.

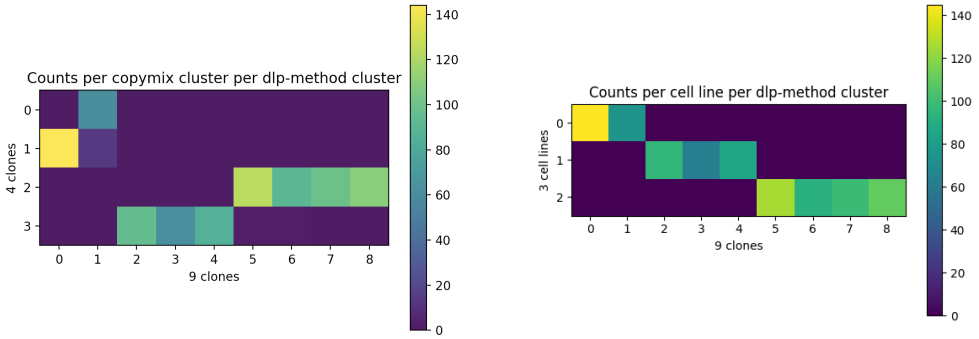

Figure 10: The co-clustering of cell lines and clones are shown, where the RHS figure corresponds to the original clones and the LHS figure corresponds to CopyMix clones.

The co-clustering, clustering concordance, between the cell lines and the 9 clones detected by the original paper, and the co-clustering between 4 CopyMix clones and 9 original clones) are shown in Fig. 10. In the LHS figure in Fig. 10, CopyMix clones 0 and 1 correspond to clones E and F, the RHS figure, that are detected by the original paper, respectively. That is, 145 cells from one cell line belong to clone E and 75 cells from the same cell line belong to clone F. The rest of the cell lines are also matched correctly, as it can be seen by comparing the plots. The V-measure when comparing CopyMix clusters with original clusters is 67%. The initialization method provides V-measure 1%. Note that the V-measure when comparing CopyMix clusters with the cell lines is 98%, and the V-measure when comparing original clusters E and F with CopyMix clusters 0 and 1 is 87%.

Next, we show the results of three metrics which confirm that 9 original clones are statistically insignificant. Table 3 shows that CopyMix outperforms the original paper w.r.t. clustering according to CH Index and Silhouette metrics. However, regarding Silhouette, the score .49 is not the perfect score, i.e., 1. Note that 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). These metrics show that the method in the original paper has detected overlapping clusters which is not in consensus with the definition of clustering in machine learning literature. Finally, performing a likelihood ratio test confirms that CopyMix as a null hypothesis is favored over the method in the original paper. Assuming each clone follows an empirical Gaussian distribution—the mean and variance can be calculated by maximum likelihood estimate, the original clones result in a log likelihood of -8980284.42 and the CopyMix log likelihood is -9023905.94. The test results in p-value 1, which means that the original likelihood is not significantly higher that CopyMix likelihood.

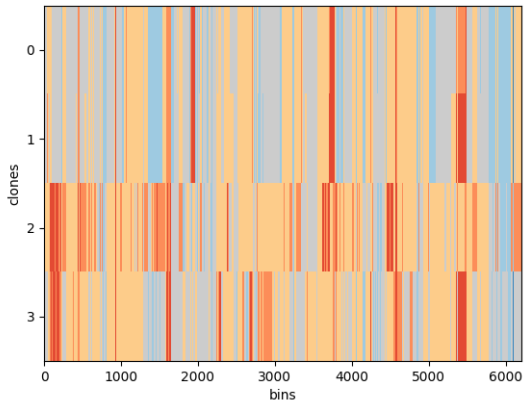

Figure 11: The copy numbers detected for 4 clones by CopyMix are shown, where the darker the color the higher they copy number is.

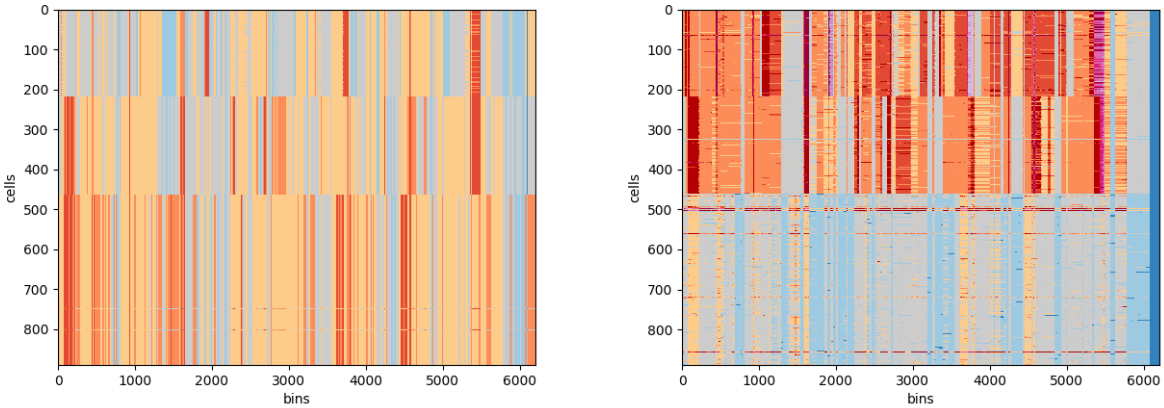

Figure 12: The cell-specific copy number profiles detected by CopyMix are shown in the LHS plot and those detected by the original paper are shown in the RHS plot. The original paper has detected copy number 0 for all clones at chromosome X which corresponds to the ending of the sequence.

Table 3: The result of CH Index and Silhouette for different numbers of clones detected by CopyMix and the original paper.

Method	Number of clusters	CH index
The method from the original paper	9	462
The three cell lines from the original paper	3	623
CopyMix	4	**1082**

Method	Number of clusters	Silhouette	Description
The method from the original paper	9	0.06	Overlapping clusters
The original paper with removal of 2 minor clones in the phylogenetic tree	7	0.21	
The original paper with removal of 4 minor clones in the phylogenetic tree	5	0.25	
CopyMix	4	**0.49**	Nicely separated clusters

Regarding the copy number profiling, Fig. 11 illustrates the copy number profiles inferred by CopyMix, where clones 0 and 1 (covering 220 cells) corresponds to clones E and F in the original paper, i.e., the two sub-clone in one cell line. We compared the copy number obtained from CopyMix and the original paper, depicted in Fig. 12. Note that since CopyMix does not infer cell-specific copy numbers we replicated a common copy number sequence for the cells belonging to common clone. The three cell lines (cells 0 to 220; cells 221 to 643; and cells 643 to 891) are clearly separated, shown in the heat maps obtained by CopyMix and the original paper results. In the heat map by CopyMix (LHS plot in Fig. 12), the sub clones for the first 220 cells (one cell line) are two groups: 1) 158 cells forming the majority of cells—these cells are in dark orange color in the heat map around bin 5500; 2) 62 cells—these cells are in yellow color and they are vertically distributed around bin 5500. In the heat map obtained by the results from the original paper (RHS plot in Fig. 12), differentiation of the two clusters in the first 220 cells can be observed by looking at the magenta-color copy numbers around bin 5500.

Note that there are two cells which, mistakenly by CopyMix, are clustered in a wrong cell line; see two horizontal lines around cell 740 and 800; this observation explains the cell lines comparisons to three main clusters detected by CopyMix, revealing V-measure 98%, as previously reported.

Aside from the visual comparison, we calculated the total variation between the copy number proportions achieved by CopyMix and the original paper, which is 91. Interestingly, the division of this total variation over the sequence length ($\frac{91}{6206} = 0.014$) is slightly less than that division for the simulated CONF 14 ($\frac{3.5}{200} = 0.017$). Due to the similar and relatively small total variations, and the fact that the clustering performance of CONF 14 (V-measure 87%) is high, we can conclude that the total variation between for the DLP data is relatively good.

We remind that CopyMix does not account for the scaling of the ploidy, which explains why the color-coded copy numbers in Fig. 12 differ between CopyMix and the original paper.

4 Conclusion

The tasks of single-cell copy number profiling and clustering have so far been sequential, requiring multiple ad-hoc processing of data; hence prone to introduce clustering artifacts. We introduce a mixture model-

based framework, CopyMix, to allow for a fully probabilistic and simultaneous single-cell copy number profiling and clustering using VI. Our approach is evaluated on both simulated and biological data resulting in good performance w.r.t. both of the tasks.

A future direction is to apply a biological indicator to detect clustering artifacts, where the contribution of non-sequential methods can be additionally illustrated. Overall, CopyMix provides various research opportunities concerning clustering and copy number profiling, such as improving clustering, computational efficiency, augmenting and refining the model, and modeling biallelic deletions and aneuploidy inspired by a recently proposed CopyKat Gao et al. (2021).

Acknowledgments

We thank Dr. Andrew Roth (University of British Columbia and British Columbia Cancer Agency) for providing useful suggestions for the evaluation of CopyMix.

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.

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.

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.

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.

Mallory, X., Edrisi, M., Navin, N., and Nakhleh, L. (2020), "Methods for copy number aberration detection from single-cell DNA-sequencing data," *Genome Biology*.

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

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.

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.

Sirazdinov, S. and Mamatov, M. (1962), "On mean convergence for densities," *Teor. Veroyatn. Primen.*, 7, 433–437.

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.

Zaccaria, S. and Raphael, B. (2020), "Accurate quantification of copy-number aberrations and whole-genome duplications in multi-sample tumor sequencing data," *Nature Communications*.

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: $\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \mathbf{\Psi}) =$
$\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{I}(Z_n = k)\mathrm{I}(C_{km} = j) \log F_{j\theta_n}(y_{nm}) =$
$\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{I}(Z_n = k)\mathrm{I}(C_{km} = j)\Big[y_{nm}\log(j\theta_n) - j\theta_n - \log(y_{nm}!))\Big]$;
$\log P(\mathbf{C}|\mathbf{\Psi}) = \sum_{k=1}^{K}\big[\sum_{j=1}^{J}\mathrm{I}(C_{k1} = j)\log\rho_{kj} + \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{I}(C_{km-1} = i, C_{km} = j)\log a_{ij}^{k}\big]$;

$$\begin{aligned}
\log P(\mathbf{Z}|\mathbf{\Psi}) &= \sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{I}(Z_n = k)\log\pi_k; \\
\log P(\mathbf{\Psi}) &= \log P(\mathbf{A}) + \log P(\boldsymbol{\theta}) + \log P(\boldsymbol{\pi}); \qquad (8)
\end{aligned}$$

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

$$\alpha_k = \alpha_k^0 + \sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big). \qquad (9)$$

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

$$\pi_{nk} = \frac{\exp(\tilde{\pi}_{nk})}{\sum_{k'=1}^{K}\exp(\tilde{\pi}_{nk'})} \tag{10}$$

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. 23 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:

$$\epsilon_{sn} = \epsilon^0_{sn} + \sum_{m=1}^{M}y_{nm} \tag{11}$$

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

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

$$\Lambda_{ij}^k = \Lambda_{ij}^{k0} + \sum_{m=2}^{M}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i, C_{km}=j)\big). \tag{12}$$

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 (2009) 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: $\mathrm{u}_k(\mathbf{C}_{1:m-1}, C_m = j) = \prod_{i=1}^{m-1}\prod_{s=1}^{J}\exp\Big\{\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big)\mathrm{E}_{q(\theta_n)}\big(D_{n,i,s}\big)\Big\}^{\mathrm{I}(C_{ki}=s)}$
$\exp\Big\{\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big)\mathrm{E}_{q(\theta_n)}\big(D_{n,m,j}\big)\Big\}$
$\prod_{j=1}^{J}\exp\Big\{\log\rho_{kj}\Big\}^{\mathrm{I}\left(C_{k1}=j\right)}$
$\prod_{t=2}^{m-1}\prod_{s=1}^{J}\prod_{i=1}^{J}\exp\Big\{\mathrm{E}_{q(\mathbf{a}_i^k)}\big(\log a_{is}^k\big)\Big\}^{\mathrm{I}\left(C_{kt-1}=i, C_{kt}=s\right)}$

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

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. 11
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. 9
    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. 12
    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. 10
    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

$$\log P(\mathbf{Y},\mathbf{Z},\mathbf{C},\mathbf{\Psi}) = \log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi}) + \log P(\mathbf{C}|\mathbf{\Psi}) + \log P(\mathbf{Z}|\mathbf{\Psi}) + \log P(\mathbf{\Psi}), \tag{13}$$

where

$$\begin{aligned}\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi}) &= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{I}(Z_n=k)\mathrm{I}(C_{km}=j)\log F_{\mu_{nj},\sigma^2}(r_{nm}) \\ &= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{I}(Z_n=k)\mathrm{I}(C_{km}=j)(-\frac{1}{2})\Big[\log(2\pi\sigma^2)+\frac{(r_{nm}-j\mu_n)^2}{\sigma^2}\Big], \end{aligned} \tag{14}$$

$$\log P(\mathbf{C}|\mathbf{\Psi}) = \sum_{k=1}^{K}\left[\sum_{j=1}^{J}\mathrm{I}(C_{k1}=j)\log\rho_{kj} + \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{I}(C_{km-1}=i,C_{km}=j)\log a_{ij}^{k}\right], \tag{15}$$

and

$$\log P(\mathbf{Z}|\mathbf{\Psi}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{I}(Z_n=k)\log\pi_k \tag{16}$$

Moreover,

$$\log P(\mathbf{\Psi}) = \log P(\mathbf{A}) + \log P(\boldsymbol{\mu}) + \log P(\frac{1}{\sigma^2}) + \log P(\boldsymbol{\pi}), \tag{17}$$

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

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

$$\log P(\boldsymbol{\mu}) = \sum_{n=1}^{N}\big[(-\frac{1}{2})\Big[\log(2\pi\tau^2)+\frac{(\mu_n-\theta)^2}{\tau^2}\Big]\big], \tag{18}$$

$$\log P(\frac{1}{\sigma^2}) = \big[-\frac{\beta}{\sigma^2}+(\alpha-1)\log(\frac{1}{\sigma^2})\big]+C, \tag{19}$$

$$\log P(\mathbf{A}) = \sum_{k=1}^{K}\sum_{i=1}^{J}\big[\sum_{j=1}^{J}(\Lambda_j-1)\log a_{ij}^{k} - \log B(\mathbf{\Lambda})\big], \tag{20}$$

and

$$\log P(\boldsymbol{\pi}) = \sum_{k=1}^{K}(\delta_k-1)\log\pi_k - \log B(\boldsymbol{\delta}). \tag{21}$$

A.5 Step 3 and 4 of Section 2.2

Update equation for π: Since (16) and (21) are the only terms in (13) 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

$$
\delta_k^* = \delta_k + \sum_{n=1}^{N} \mathrm{E}_{q(Z_n)} \big(\mathrm{I}(Z_n = k) \big). \tag{22}
$$

Update equation for Z_n: Since (14) and (16) are the only terms in (13) 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.,

$$
\begin{aligned}
\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) &= \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{j=1}^{J} \mathrm{I}(Z_n = k) \mathrm{I}(C_{km} = j) (-\frac{1}{2}) \Big[\log(2\pi\sigma^2) + \frac{(r_{nm} - j\mu_n)^2}{\sigma^2} \Big] \\
&+ \sum_{l \neq n} \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{j=1}^{J} \mathrm{I}(Z_l = k) \mathrm{I}(C_{km} = j) (-\frac{1}{2}) \Big[\log(2\pi\sigma^2) + \frac{(r_{lm} - j\mu_l)^2}{\sigma^2} \Big]
\end{aligned} \tag{23}
$$

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:

$$
D_{nmj} = -\frac{1}{2} \Big[\log(2\pi\sigma^2) + \frac{(r_{nm} - j\mu_n)^2}{\sigma^2} \Big] \tag{24}
$$

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

$$
p_{nk}^* = \frac{\exp(\tilde{p}_{nk})}{\sum_{k'=1}^{K} \exp(\tilde{p}_{nk'})}. \tag{25}
$$

Update equation for μ: Since (14) and (18) are the only terms in (13) 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. 23, 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, $\log q^*(\mu_n)$

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

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

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

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

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

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

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. 14, 19. Putting them together, we achieve the following.

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

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

$$\alpha^* = \alpha + \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big) = \alpha + \frac{NM}{2} \tag{28}$$

$$\beta^* = \beta + \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\Big(r_{nm}^2 - 2jr_{nm}\theta_n^* + j^2(\tau_n^{*2}+\theta_n^{*2})\Big) \tag{29}$$

Update equation for $\mathbf{a}_i^k$: Because (15) and (20) are the only terms in (13) 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

$$\Lambda_{ij}^{k*} = \Lambda_j + \sum_{m=2}^{M}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i, C_{km}=j)\big) \tag{30}$$

Update equation for $\mathbf{C}_k$: Since only (14) and (15) 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: $\phi_m = \sum_{C_1=1}^{J} .. \sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-1}, C$
$\sum_{C_1=1}^{J} .. \sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-2}, C_{m-1}, C_m) =$
$\sum_{C_1=1}^{J} .. \sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-2}, C_{m-1})w(C_m)w(C_{m-1}C_m) = w(C_m)\sum_{C_{m-1}=1}^{J} \phi_{m-1}w(C_{m-1}C_m)$
$\beta_m = \sum_{C_{m+1}=1}^{J} .. \sum_{C_M=1}^{J} v_k(\mathbf{C}_{m+1:M}, C_m) = \sum_{C_{m+1}=1}^{J} .. \sum_{C_M=1}^{J} v_k(\mathbf{C}_{m+2:M}, C_{m+1}, C_m) =$
$\sum_{C_{m+1}=1}^{J} .. \sum_{C_M=1}^{J} v_k(\mathbf{C}_{m+2:M})w(C_mC_{m+1})w(C_{m+1}) = \sum_{C_{m+1}=1}^{J} \beta_{m+1}w(C_mC_{m+1})w(C_{m+1})$ 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

$$\boldsymbol{\Psi}(x) = \frac{d}{dx}\log\Gamma(x), \tag{31}$$

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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\mathrm{E}_{q(\boldsymbol{\pi})}(\log \boldsymbol{\pi}) = \mathbf{\Psi}(\delta_k^*) - \mathbf{\Psi}\Big(\sum_{k=1}^{K}\delta_k^*\Big) \quad \forall k \in K \tag{32}$$

$$\mathrm{E}_{q(\mathbf{a}_i^k)}(\log \mathbf{a}_i^k) = \mathbf{\Psi}(\Lambda_{ij}^{k*}) - \mathbf{\Psi}\Big(\sum_{j=1}^{J}\Lambda_{ij}^{k*}\Big) \quad \forall j \in J \tag{33}$$

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:

$$\mathrm{E}_{(q(Z_n),q(\boldsymbol{\pi}))}\Big[\log p(Z_n|\Psi)\Big]=\mathrm{E}_{(q(Z_n),q(\boldsymbol{\pi}))}\Big[\sum_{k=1}^{K}\mathrm{I}(Z_n=k)\log\pi_k\Big]=\sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n=k)\Big]\mathrm{E}_{q(\boldsymbol{\pi})}\Big[\log\pi_k\Big] \tag{34}$$

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 $\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n=k)\Big]=q(Z_n=k)$.

$$\mathrm{E}_{q(Z_n)}\Big[\log q(Z_n)\Big]=\mathrm{E}_{q(Z_n)}\Big[\sum_{k=1}^{K}\mathrm{I}(Z_n=k)\log q(Z_n=k)\Big]=\sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n=k)\Big]\log q(Z_n=k) \tag{35}$$

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

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

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

$$\mathrm{E}_{q(\mu_n)}[\log q(\mu_n)]=(\frac{-1}{2})\Big(\log(2\pi\tau_n^{*2})+\frac{\mathrm{E}_{q(\mu_n)}\Big[\mu_n^2\Big]-2\theta_n^*\mathrm{E}_{q(\mu_n)}\Big[\mu_n\Big]+\theta_n^{*2}}{\tau_n^{*2}}\Big)=(\frac{-1}{2})\Big(\log(2\pi\tau_n^{*2})+1\Big) \tag{37}$$

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:

$$\mathrm{E}_{q(\boldsymbol{\pi})}(\log q(\boldsymbol{\pi})) = \Big(\sum_{k=1}^{K}\delta_k^* - K\Big)\boldsymbol{\Psi}\Big(\sum_{k=1}^{K}\delta_k^*\Big) - \sum_{k=1}^{K}\boldsymbol{\Psi}(\delta_k^*)(\delta_k^*-1) - \log B(\boldsymbol{\delta}^*) \tag{38}$$

$$\begin{aligned}
&\text{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, $\Big(\alpha^*-\ln\beta^*+\ln\Gamma(\alpha^*)+(1-\alpha^*)\boldsymbol{\Psi}(\alpha^*)\Big)$ 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:

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