bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

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: negars@kth.se

Abstract

Motivation: Single-cell sequencing technologies are becoming increasingly more established, in particular, in the study of tumor heterogeneity, i.e., the cell subpopulations that a cancer tumor typically comprises. Investigating tumor heterogeneity is imperative to better understand how tumors evolve since each of 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 detecting single-cell copy number variations (copy number profiling) as well as clustering; however, these two tasks have up to now been handled sequentially with various ad-hoc preprocessing steps lacking an automated, generalized and fully probabilistic framework.

Results: We propose CopyMix, a novel probabilistic mixture model based method for single-cell clustering and copy number profiling using Variational Inference, to simultaneously cluster cells and infer copy number profiles corresponding to the clusters. CopyMix is evaluated using simulated data as well as published biological data from metastatic colorectal cancer. The results reveal high V-measures for clustering and low errors in copy number inference. These favorable results indicate a considerable potential to obtain clinical impact by using CopyMix in studies of cancer tumor heterogeneity.

Availability: The software is available at: https://github.com/negar7918/CopyMix and the previously published biological dataset is available from the NCBI Sequence Read Archive (https://www.ncbi.nlm.nih.gov/sra) under accession number SRP074289)

Keywords: single-cell sequencing; copy number calling; clustering; mixture models; variational inference.

1 Introduction

A tumor typically consists of a collection of heterogeneous cell populations, each having distinct genetic and phenotypic properties, in particular, concerning 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

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

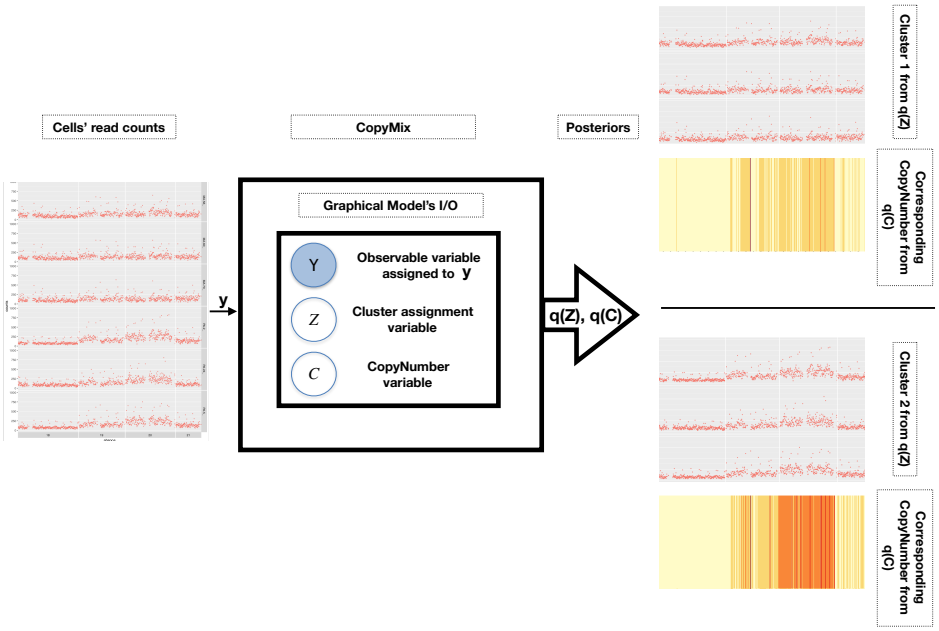

Figure 1: The overview of CopyMix with: input y, cells' reads; outputs $q(Z)$ and $q(C)$, posterior distributions resulting in clusters within y with their corresponding copy number profiles illustrated by heatmaps (this is an example of binary clustering).

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 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 in the near future.

Current single-cell datasets pose a wealth of computational challenges. Computational methods have emerged for detecting single-cell copy number variations (copy number profiling) as well as clustering; however, these two tasks have up to now been handled sequentially with various ad-hoc preprocessing steps lacking an automated, generalized and fully probabilistic framework. Concerning copy number profiling, Hidden Markov Models (HMMs) have been applied to individual cells in different settings (Vitak et al., 2017; Zahn et al., 2017). Leung et al. (2017) used a changepoint method (Nilsen et al., 2012) to infer single-cell copy number profiles. They preprocessed the data before changepoint detection and then scaled the copy number profiles to have a mean equal to the ploidy of the originating tumor. After derivation of copy number profiles, Leung et al. (2017) clustered the cells using hierarchical clustering based on Euclidean distances and Ward-linkage.

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Here, we develop CopyMix, a novel method to analyze single-cell copy number data, using a probabilistic model-based approach, that allows us to simultaneously cluster cells and infer copy number profiles by combining statistical strength from data across all cells and neighbouring sites. Model-based methods are interpretable and measures of uncertainty can be obtained for the inferred quantities. To our best knowledge, no earlier work has simultaneously performed copy number inference and cell clustering; however, similar approaches have been considered for single-cell DNA methylation (Kapourani and Sanguinetti, 2019; de Souza et al., 2018), 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 clustering approach that uses a mixture model with components, corresponding to clusters, each having a specific copy number profile modeled by a sequence of latent variables. 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 distribution of all quantities of interest rather than point estimates, it protects against over-fitting, and renders it possible to perform principled determination of the optimal number of mixture components (Bishop, 2006).

Fig. 1 illustrates an overview of CopyMix where 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.

2 Our novel framework: CopyMix

CopyMix is a probabilistic clustering method based on a mixture model with components, corresponding to clusters, each having a specific copy number profile modeled by a sequence of latent variables. Similarly as in Shah et al. (2006); Vitak et al. (2017); Zahn et al. (2017), we assume that each latent copy number sequence is governed by a Markov chain (describing a sequence of possible copy numbers in which the probability of each copy number value, defined as state, depends only on the previous state in the sequence). In CopyMix, 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. Note that our inference framework can easily be adapted to other emission distributions aside from the Poisson distribution.

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 and we incorporate this assumption in our model by defining the rate of Poisson distribution as the product of copy number state and cell-specific rate, θ_n, see Eq. (1); this also implies that θ_n corresponds to the average sequencing coverage for a haploid genome. Due to this multiplicative structure for the Poisson rate, a copy number state of zero results in the lowest rate implying copy number deletion events; however, since the rate of Poisson cannot be zero our implementation of CopyMix handles this by assigning a very small number instead of zero. When it comes to the priors, we use conjugacy. Our model is described in more detail in what follows.

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

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,

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

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

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

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

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

Next, we describe the main steps of the proposed VI approach for inferring $\mathbf{Z}, \mathbf{C}$ and $\mathbf{\Psi}$.

Step 1. VD factorization

We assume the following factorization of the VD:

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

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

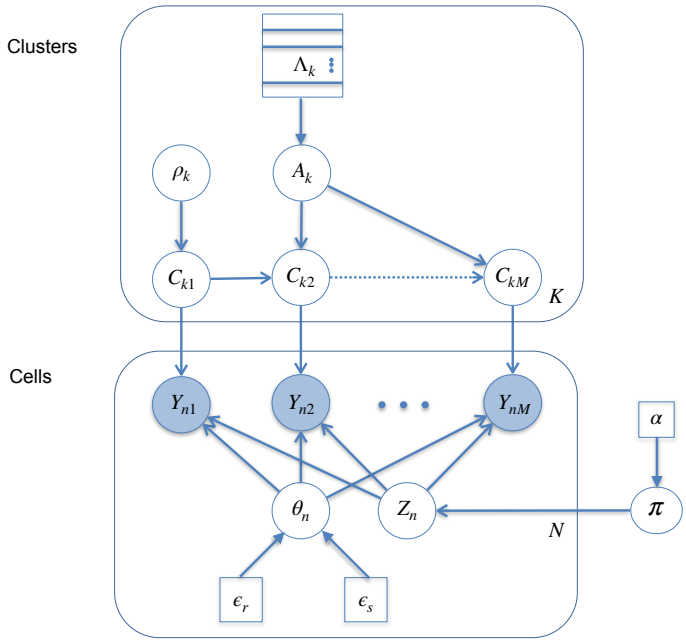

Figure 2: Probabilistic graphical model representing our proposed model. Shaded nodes represent observed values, the unshaded ones are the latent variables and the squares are the hyperparameters of the model; a posterior distribution over the values of the unshaded nodes is approximated using Variational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

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

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 calculations see Appendix A.1.

Step 3. VD computation

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

Update equation for $\boldsymbol{\pi}$: Let $\mathrm{I}(\cdot)$ be the indicator function. $q(\boldsymbol{\pi})$ is a Dirichlet distribution with parameters $\alpha = (\alpha_1, \ldots, \alpha_K)$, where

$$\alpha_k = \alpha_k^0 + \sum_{n=1}^{N} \mathrm{E}_{q(Z_n)}(\mathrm{I}(Z_n = k)). \tag{4}$$

Update equation for Z_n: 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 $D_{n,m,j} \equiv y_{nm} \log \theta_n + y_{nm} \log j - j\theta_n - \log(y_{nm}!)$.*Thedistribution* $q(Z_n)$ is a Categorical distribution with parameters $\boldsymbol{\pi}_n = (\pi_{n1}, \ldots, \pi_{nK})$, where

$$\pi_{nk} = \frac{\exp(\tilde{\pi}_{nk})}{\sum_{k'=1}^{K} \exp(\tilde{\pi}_{nk'})} \tag{5}$$

with

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

Update equation for θ_n: $q(\theta_n)$ is a Gamma distribution with shape and rate parameters:

$$\epsilon_{sn} = \epsilon_{sn}^0 + \sum_{m=1}^{M} y_{nm} \tag{6}$$

$$\epsilon_{rn} = \epsilon_{rn}^0 + \Big\{ \sum_{m=1}^{M} \sum_{k=1}^{K} \mathrm{E}_{q(Z_n)}(\mathrm{I}(Z_n = k)) \sum_{j=1}^{J} \mathrm{E}_{q(\mathbf{C}_k)}(\mathrm{I}(C_{km} = j)) j \Big\}. \tag{7}$$

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

$$\Lambda_{ij}^k = \Lambda_{ij}^{k0} + \sum_{m=2}^{M} \mathrm{E}_{q(\mathbf{C}_k)}(\mathrm{I}(C_{km-1} = i, C_{km} = j)). \tag{8}$$

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

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^k_{mj}) and backward (β^k_{mj}) 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. The update equations are as follows.

$$w_k(C_{m-1i}C_{mj}) = \exp\{\mathrm{E}_{q(\mathbf{a}^k_i)}(\log a^k_{ij})\} \tag{9}$$

$$w_k(C_{mj}) = \exp\{\sum_{n=1}^{N} \mathrm{E}_{q(Z_n)}(\mathrm{I}(Z_n = k))\mathrm{E}_{q(\theta_n)}(D_{n,m,j})\} \tag{10}$$

$$q(C_{km} = j) = \frac{\phi^k_{mj}\beta^k_{mj}}{\sum_{j=1}^{J}\phi^k_{mj}\beta^k_{mj}} \tag{11}$$

$$q(C_{km-1} = i, C_{km} = j) = \frac{w_k(C_{m-1i}C_{mj})w_k(C_{mj})\phi^k_{m-1i}\beta^k_{mj}}{\sum_{j=1}^{J} w_k(C_{m-1i}C_{mj})w_k(C_{mj})\phi^k_{m-1i}\beta^k_{mj}} \tag{12}$$

Step 4. Summary of updates

We update the parameters of each the posterior distribution presented in Step 3 using Algorithm 1. As we aim to find the posterior distributions of cluster assignments and copy number states, we only return $q(Z_n)$ and $q(C_{km} = j)$ for all the values of n, k, m and j.

3 Empirical investigation

3.1 Experimental setup

3.1.1 Dataset

We perform experiments on both simulated data and a published biological dataset (Leung et al., 2017) (available from the NCBI Sequence Read Archive (https://www.ncbi.nlm.nih.gov/sraSRA) under accession number SRP074289). The biological data stem from single-cell whole-genome sequencing from 18

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Algorithm 1 Variational Inference for clustering and hidden state inference

procedure ESTIMATE-POSTERIORS(y): y, the input of N sequences of length M
Initialise 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. 6
Update $w_k(C_{m-1i}C_{mj})$ using $\mathbf{\Lambda}^{k0}_i$ and π_{nk} in Eq. 9
Update $w_k(C_{mj})$ using $\mathbf{\Lambda}^{k0}_i$, π_{nk}, ϵ^0_{rn}, ϵ_{sn} and y in Eq. 10
    repeat
        Update $\alpha^{(c)}_k$ using $\pi^{(c-1)}_{nk}$ and α^0_k in Eq. 4
        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. 9 to 12)
        Update $\mathbf{\Lambda}^{k(c)}_i$ using $\mathbf{\Lambda}^{k0}_i$ and $q(C_{km-1}=i, C_{km}=j)^{(c)}$ in Eq. 8
        Update $\epsilon^{(c)}_{rn}$ using ϵ^0_{rn}, $\pi^{(c-1)}_{nk}$ and $q(C_{km}=j)^{(c)}$ in Eq. 7
        Update $\pi^{(c)}_{nk}$ using $\alpha^{(c)}_k$, ϵ_{sn}, $\epsilon^{(c)}_{rn}$, $q(C_{km}=j)^{(c)}$ and y in Eq. 5
        Update $w_k(C_{mj})^{(c)}$ using $\mathbf{\Lambda}^{k(c)}_i$, $\pi^{(c)}_{kn}$, $\epsilon^{(c)}_{rn}$, ϵ_{sn} and y in Eq. 10
        Update $w_k(C_{m-1i}C_{mj})^{(c)}$ using $\mathbf{\Lambda}^{k(c)}_i$ and $\pi^{(c)}_{nk}$ in Eq. 9
        $c \leftarrow c+1$
    until convergence of the ELBO in Eq. 2
Output: $q(Z_n) \sim Cat(\pi_{n1}, .., \pi_{nK})$, cluster probabilities
Output: $q(C_{km}=j)$, hidden state probabilities
return $q(Z_n)$ and $q(C_{km}=j)$

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 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 in the dataset corresponds to the total number of reads aligned per bin per cell after GC correction (Leung et al., 2017). These counts reflect the hidden copy numbers—copy number integers are conventionally ranged from 0 to 6 as in Leung et al. (2017)—in the tumor cell's genome. 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 colon and the metastatic cells from 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 generate 30 datasets for each test case, following our graphical model and inspired by biological datasets from Leung et al. (2017) (details in section 3.2.1); for implementation details, see README.txt in https://github.com/negar7918/CopyMix. We examine different test cases that are orthogonal to each other, in order to show the robustness of our approach; each scenario, for both clustering and copy number state inference, is reproducible.

3.1.2 Experimental Protocol

As maximizing the ELBO, given in Eq. 2, is a non-convex optimization problem, it can lead to a local optimum; to avoid this problem, it is crucial to initialize the proposed VI algorithm properly (Blei et al., 2017). We develop the following initialization framework to tackle this challenge. We run VI for different

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

numbers of clusters considering different initialization methods; below is the list of initialization methods:

- **k-means+bw**: run k-means to find clusters then perform Baum-Welch (Baum, 1972) on HMM with Poisson distributed observations (Paroli et al., 2000) to infer the copy number sequences.
- **k-means+bw-2**: run k-means to find clusters then perform a modified Baum-Welch (similar to the usual Baum-Welch except that the Poisson distribution has the same form as in Eq. 1) to infer the copy number sequences.
- **k-means+rand**: run k-means to find clusters then randomly assign the copy number sequences
- **non-Markovian EM**: run EM over a graphical model similar to the one in Fig 2 but without dependencies between C's, to simultaneously cluster and infer the copy number sequences.
- **rand**: randomly assign clusters and copy number sequences.
- **rand+bw**: randomly assign clusters then perform Baum-Welch on Poisson HMM to infer the copy number sequences.

For each given number of clusters, we choose the best VI run based on the highest expected value of the 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). V-measure is a score between zero and one, where one stands for perfect clustering. V-measure captures the homogeneity (only those cells that are members of a single group are assigned to a single cluster) and completeness (all of those cells that are members of a single group are assigned to a single cluster) properties. 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 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; regarding state inference, we compare the copy number breakpoints between the results from CopyMix and the original paper Leung et al. (2017).

3.2 Experimental Results

3.2.1 Simulated data

Here, we compare CopyMix to the VI initialization methods, listed in section 3.1.2, on 30 simulated datasets for each test scenario; 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 3, 4 and 5 (in the boxplots, the median values are highlighted in red for initialisation method and blue for CopyMix to facilitate comparisons) provide summaries of V-measures and proportions of discordant positions for different settings of: copy number transition matrices and

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Table 1: This table shows the initialization methods chosen by CopyMix framework concerning configuration A, C, D and E in Fig 3 across 30 different datasets.

A	C	D	E
k-means+bw (15)	k-means+rand (5)	non-Markovian EM (19)	k-means+rand (4)
k-means+bw-2 (1)	rand (25)	k-means+bw (4)	k-means+bw (10)
rand (14)		rand+bw (7)	rand (16)

The number in the parentheses is number of datasets using the initialization method

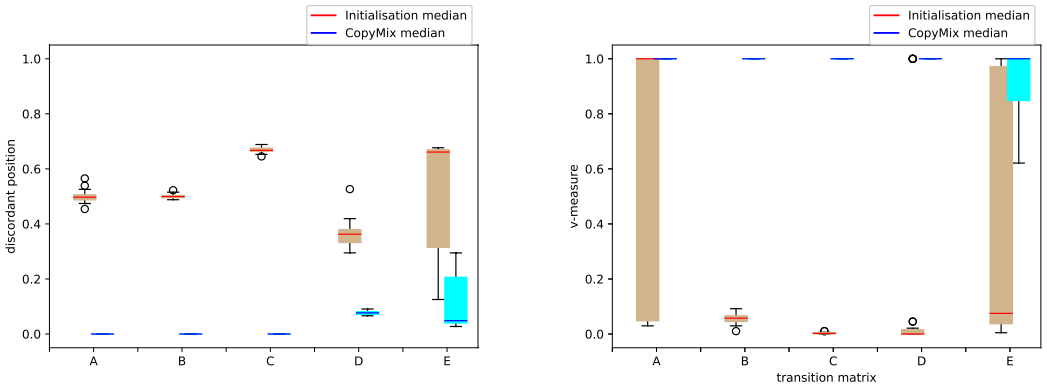

Figure 3: Performance of initialization methods (brown bars with red median values) and CopyMix (cyan bars with blue median values) across different sets of transition matrices *Left:* discordant positions between true and inferred vectors of copy number states. *Right:* V-measure between true and inferred cell clustering assignments.

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. 3) 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]. Note that CopyMix benefits from initializations but interestingly also recovers from poor ones; moreover, CopyMix outperforms non-Markovian EM, k-means and two versions of Baum-Welch included in the initialization methods. Table 1, lists the initialization methods chosen by our framework as the best ones based on the highest likelihood, across 30 datasets for the configurations A, C, D and E in Fig. 3; the rest of the figures in this section have similar results concerning the chosen initialization methods by our framework.

The results in Fig. 3 are obtained considering 45 cells—each with a sequence length of 800 bins—clustered into two (configurations A to D) and three (configuration E) clusters varying the copy number variation (transition) patterns in each cluster. The mean values of θ_n are around 100. 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; The configurations are based on the following definitions: **single state**, a copy number state is inclined to stay at one certain state; **self-transition**, a copy number state is inclined to stay at its current state for awhile; **changing state**, a copy number state changes shortly after being at a state; **non-homogeneous transition**, a sudden change of copy number state arises for

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

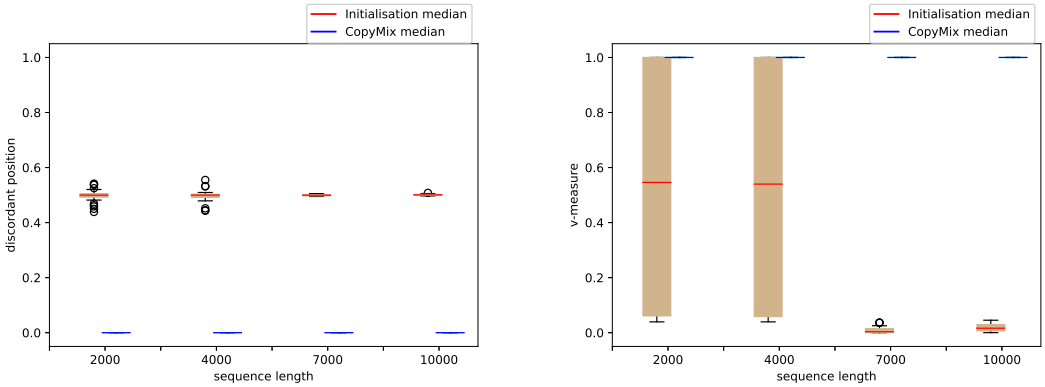

Figure 4: Performance of initialization methods and CopyMix varying the sequence length. Left and right as well as color codes are similar to Fig. 3.

a specific short range of positions in the copy number sequence; **shift state**, copy number state patterns of different clusters differ by a multiplicative or additive constant. We now describe the configurations of copy number transition patterns: **A**, similar to chromosomes 11 in CRC1 and 4 in CRC2 (shift state pattern); **B**, similar to chromosome 11 in CRC1 (one cluster has a self-transition pattern and the other one single state pattern); **C**, similar to chromosome 3 in CRC1 (two clusters have identical transition matrices but one follows the non-homogeneous transition pattern); **D**, similar to chromosomes 18 to 21 in CRC2 which is the dataset used in section 3.2.2 (self-transition, changing state and shift state patterns); **E**, two clusters have the transitions in case C and the third cluster has a combination of changing state and single state pattern . As we can see, 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. Moreover, we investigae 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.

Next, we test configuration A by increasing 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. 4. We can observe that our model handles these sequences without any decrease in performance.

Finally, Fig. 5 illustrates the perfect performance of CopyMix when varying the number of cells from 15 to 500 in configuration A.

3.2.2 Biological data

Next, we evaluate CopyMix on the biological published data from Leung et al. (2017). CopyMix clusters the cells into primary and metastatic groups which compared to the clustering results of Leung et al. (2017) leads to a perfect V-measure of one. Metastatic tumor cells were clustered into two major subclusters in Leung et al. (2017) by amplifications on chromosomes 3 and 8 where some known genes are present; hence, considering chromosomes 18 to 21 (our biological dataset) is reasonable for the purpose of only detecting the two major clusters. Note that CopyMix detects the three clusters if chromosomes 3 and

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

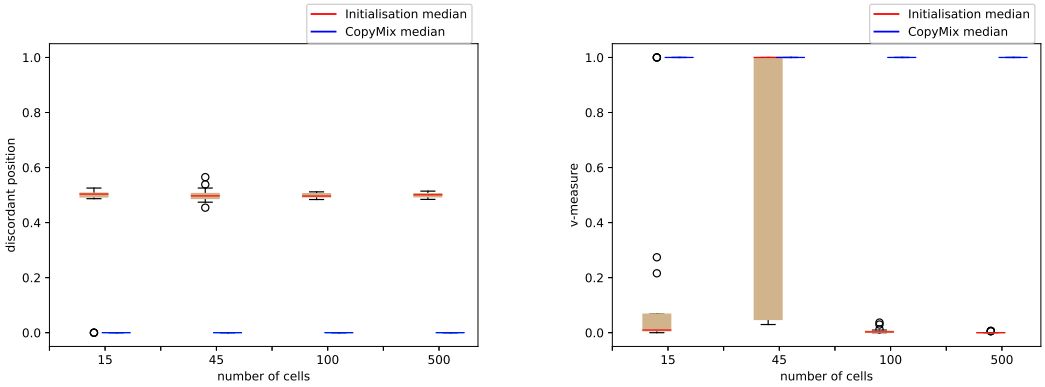

Figure 5: Performance of initialization methods and CopyMix varying the number of cells. Left and right as well as color codes are similar to Fig. 3.

8 are added. However, Leung et al. (2017) does not present a method to decide over the threshold in hierarchical clustering which leads to the chosen the level of subclusters in the hierarchy; moreover, there is no ground truth that we can judge their results. Applying other state-of-the-art clustering methods on the bin counts (e.g., k-means, hierarchical clustering (Bar-Joseph et al., 2001)—this is used by Leung et al. (2017) after copy number profiling—and density-based spatial clustering (Ester et al., 1996)) leads to poor results with V-measure between 0 and .1.

Regarding the copy number profiling, Fig. 6 illustrates the copy number profiles inferred by CopyMix and the original work by (Leung et al., 2017). The heatmaps indicate higher copy number state by a darker color. We can see that CopyMix detects the breakpoints (copy number changes) reported in (Leung et al., 2017) with an accuracy of 88%, considering a margin (of a size corresponding to 1.5% of the sequence length) around the breakpoints; moreover, we can observe that CopyMix further detects finer breakpoints than the ones from (Leung et al., 2017). Note that Leung et al., 2017 perform ploidy scaling which contributes to some copy number scale differences in comparison to the results of CopyMix.

4 Conclusion

The tasks of single-cell copy number profiling and clustering have so far been sequential, requiring different and ad-hoc preprocessing of data lacking an automated, generalized and fully probabilistic framework. 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—the biological dataset concerns the clustering and copy number profiling of cancer primary and metastatic cells. Regarding clustering, the V-measure of 1 is achieved for all of the test cases involved except one case with values in the interval of [.85, 1]. Concerning the copy number profiling, results from simulated data lead mostly to zero false detected states; for biological data, CopyMix detects 88% of the breakpoints from the original paper Leung et al. (2017). In addition, CopyMix discovers refined copy number changes between major breakpoints; this allows for potential future investigation on the substructure of copy number variations, with possible clinical relevance.

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

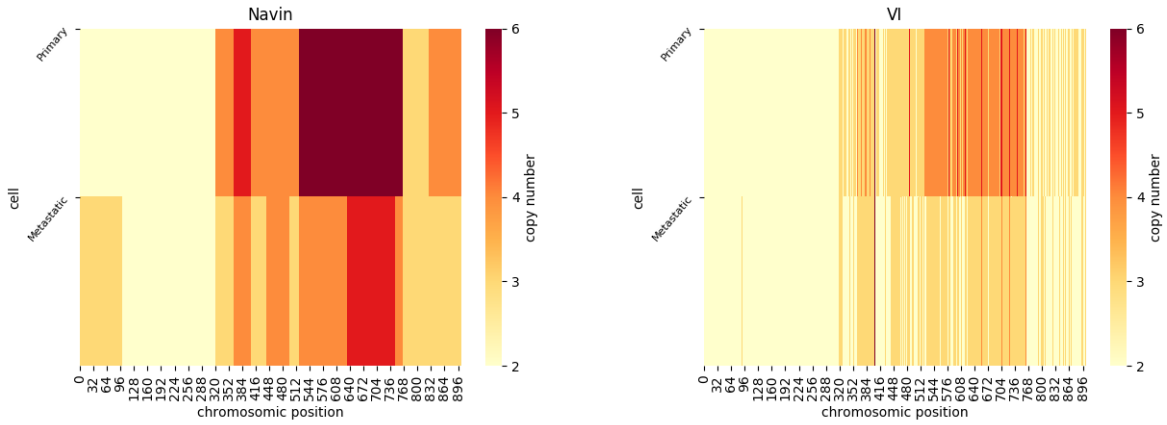

Figure 6: Copy number inference comparison between the results of (Leung et al., 2017) (*left*) and CopyMix (*right*) for both primary and metastatic clusters.

Having good results on shallow simulated data, the proposed approach can be applied and further tested for shallow data as this is a challenge described in Zahn et al. (2017). CopyMix provides for several different research opportunities concerning clustering and state inference, for example: reducing sensitivity in VI initialization; improving computational efficiency; and augmenting and refining the model.

References

Z. Bar-Joseph, D. Gifford, and T. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17(1):S22–9, 2001.

T. Baslan, J. Kendall, L. Rodgers, H. Cox, M. Riggs, A. Stepansky, . . ., N. Navin, and J. Hicks. Genome-wide copy number analysis of single cells. *Nature Protocols*, 7(6):1024–10241, 2012.

L. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3:1–8, 1972.

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

D. Blei, A. Kucukelbir, and J. Mcauliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Camila PE de Souza, Mirela Andronescu, Tehmina Masud, Farhia Kabeer, Justina Biele, Emma Laks, Daniel Lai, Jazmine Brimhall, Beixi Wang, Edmund Su, et al. Epiclomal: probabilistic clustering of sparse single-cell dna methylation data. *bioRxiv*, page 414482, 2018.

Peter Eirew, Adi Steif, Jaswinder Khattra, Gavin Ha, Damian Yap, Hossein Farahani, Karen Gelmon, Stephen Chia, Colin Mar, Adrian Wan, E. Laks, J. Biele, K. Shumansky, J. Rosner, A. McPherson, C. Nielsen, A. Roth, C. Lefebvre, A. Bashashati, C. de Souza, C. Siu, R. Aniba, et al. Dynamics of genomic clones in breast cancer patient xenografts at single cell resolution. *Nature*, 518(7539):422, 2015.

M. Ester, HP. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. *International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

Charles Gawad, Winston Koh, and Stephen R Quake. Single-cell genome sequencing: current state of the science. *Nature Reviews Genetics*, 17(3):175, 2016.

Richard Arnold Johnson and Dean W Wichern. *Applied multivariate statistical analysis, 6th Ed.* Pearson Prentice Hall, 2007.

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Chantriolnt-Andreas Kapourani and Guido Sanguinetti. Melissa: Bayesian clustering and imputation of single-cell methylomes. *Genome biology*, 20(1):61, 2019.

D. Lawson, K. Kessenbrock, R. Davis, N. Pervolarakis, and Z. Werb. Tumour heterogeneity and metastasis at single-cell resolution. *Nature Cell Biology*, 20(12):1349–1360, 2018.

M. Leung, A. Davis, R. Gao, A. Casasent, Y. Wang, E. Sei, . . ., and N. Navin. Single-cell dna sequencing reveals a late-dissemination model in metastatic colorectal cancer. *Genome Research*, 27(8):1287–1299, 2017.

David J.C. MacKay. Ensemble learning for hidden markov models. Technical report, 1997.

Clare A McGrory and DM Titterington. Variational bayesian analysis for hidden markov models. *Australian & New Zealand Journal of Statistics*, 51(2):227–244, 2009.

Nicholas Navin, Jude Kendall, Jennifer Troge, Peter Andrews, Linda Rodgers, Jeanne McIndoo, Kerry Cook, Asya Stepansky, Dan Levy, Diane Esposito, et al. Tumour evolution inferred by single-cell sequencing. *Nature*, 472(7341):90, 2011.

G. Nilsen, K. Liestøl, P. Loo, HK. Moen Vollan, MB. Eide, . . ., and OC. Lingjærde. Copynumber: Efficient algorithms for single- and multi-track copy number segmentation. *BMC Genomics*, 13(1), 2012.

Peter C Nowell. The clonal evolution of tumor cell populations. *Science*, 194(4260):23–28, 1976.

Paroli, Redaelli, and Spezia. Poisson hidden Markov models for time series of overdispersed insurance counts. *ASTIN Colloquium International Actuarial Association*, pages 461–474, 2000.

Andrew Rosenberg and Julia Hirschberg. 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)*, 2007.

Andrew Roth, Andrew McPherson, Emma Laks, Justina Biele, Damian Yap, Adrian Wan, Maia A Smith, Cydney B Nielsen, Jessica N McAlpine, Samuel Aparicio, et al. Clonal genotype and population structure inference from single-cell tumor sequencing. *Nature methods*, 13(7):573, 2016.

Sohrab P Shah, Xiang Xuan, Ron J DeLeeuw, Mehrnoush Khojasteh, Wan L Lam, Raymond Ng, and Kevin P Murphy. Integrating copy number polymorphisms into array cgh analysis using a robust hmm. *Bioinformatics*, 22(14):e431–e439, 2006.

Ehud Shapiro, Tamir Biezuner, and Sten Linnarsson. Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nature Reviews Genetics*, 14(9):618, 2013.

D. Spiegelhalter, N. Best, B. Carlin, and A. Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639, 2002.

S. Vitak, K. Torkenczy, J. Rosenkrantz, A. Fields, L. Christiansen, ..., and A. Adey. Sequencing thousands of single-cell genomes with combinatorial indexing. 2017.

D. Witten. Classification and clustering of sequencing data using a poisson model. *Ann. Appl. Stat.*, 5(4):2493–2518, 2011.

Hans Zahn, Adi Steif, Emma Laks, Peter Eirew, Michael VanInsberghe, Sohrab P Shah, Samuel Aparicio, and Carl L Hansen. Scalable whole-genome single-cell library preparation without preamplification. *Nature methods*, 14(2):167, 2017.

C. Zuo, K. Chen, KJ. Hewitt, EH. Bresnick, and S. Keleş. A hierarchical framework for state-space matrix inference and clustering. *Ann. Appl. Stat.*, 10(3):1348–1372, 2016.

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

A Appendix: Derivation details

A.1 Step 2 of Section 2

Each term of the logarithm of the joint distribution presented in Step 2 of Section 2 is calculated as follows:

$$\begin{aligned}\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi}) &= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{I}(Z_n=k)\mathrm{I}(C_{km}=j)\log F_{j\theta_n}(y_{nm})\\ &= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{I}(Z_n=k)\mathrm{I}(C_{km}=j)\Big[y_{nm}\log(j\theta_n)-j\theta_n-\log(y_{nm}!))\Big];\end{aligned}$$

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

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

$$\log P(\mathbf{\Psi}) = \log P(\mathbf{A})+\log P(\boldsymbol{\theta})+\log P(\boldsymbol{\pi})$$

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 of Section 2

In step 3 of section 2, In what follows we present the derivation of each posterior update equation. For convenience, we use $\overset{+}{\approx}$ to denote equality up to a constant additive factor.
Update equation for π:

$$\begin{aligned}\log q(\boldsymbol{\pi}) &\overset{+}{\approx} \mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\mathbf{Y},\mathbf{Z},\mathbf{C},\mathbf{\Psi})\big)\\ &\overset{+}{\approx} \mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\mathbf{Z}|\mathbf{\Psi})\big)+\mathrm{E}_{-\boldsymbol{\pi}}\big(\log P(\boldsymbol{\pi})\big)\\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\log\pi_k+\log P(\boldsymbol{\pi})\\ &= \sum_{k=1}^{K}\log\pi_k\Big[\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\Big]+\sum_{k=1}^{K}\log\pi_k(\alpha_k^0-1)\\ &= \sum_{k=1}^{K}\log\pi_k\Big[\Big(\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)+\alpha_k^0\Big)-1\Big].\end{aligned}$$

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Update equation for Z_n:

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

Let $D_{n,m,j}\equiv y_{nm}\log\theta_n+y_{nm}\log j-j\theta_n-\log(y_{nm}!)$. Note that $\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\boldsymbol{\Psi})$ 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.,

$$\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\boldsymbol{\Psi})=\textstyle\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{I}(Z_n=k)\mathrm{I}(C_{km}=j)D_{n,m,j}\ +\\ \textstyle\sum_{l\neq n}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{I}(Z_l=k)\mathrm{I}(C_{km}=j)D_{n,m,l}$$

and

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

Consequently,

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

Update equation for θ_n:

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

We can write $\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\boldsymbol{\Psi})$ as in Eq. 13 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,

$$\begin{aligned}\log q(\theta_n) \overset{+}{\approx}\ & \log\theta_n\Big[\sum_{m=1}^{M}y_{nm}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\Big]\\ &-\theta_n\Big[\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)j\Big]\\ &+\ \log\theta_n(\epsilon^0_{sn}-1)-\theta_n\epsilon^0_{rn}.\end{aligned}$$

Note that $\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)=1$.
Update equation for $\mathbf{a}_i^k$:

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

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

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

$$\begin{aligned}\log q(\mathbf{a}_i^k) \quad &\overset{+}{\approx} \quad \sum_{j=1}^{J}\sum_{m=2}^{M} \mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i, C_{km}=j)\big)\log a_{ij}^k + \sum_{j=1}^{J}(\Lambda_{ij}^{k0}-1)\log a_{ij}^k \\ &= \quad \sum_{j=1}^{J}\log a_{ij}\Big\{\Big[\Lambda_{ij}^{k0} + \sum_{m=2}^{M}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i, C_{km}=j)\big)\Big]-1\Big\}.\end{aligned}$$

Update equation for $\mathbf{C}_k$*:*

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

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,

$$\begin{aligned}\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\mathbf{\Psi}) \quad &= \quad \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{j=1}^{J}\mathrm{I}(Z_n=k)\mathrm{I}(C_{km}=j)D_{n,m,j} \\ &\quad + \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k'\neq k}\sum_{j=1}^{J}\mathrm{I}(Z_n=k)\mathrm{I}(C_{k'm}=j)D_{n,m,j}.\end{aligned}$$

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:

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

Calculations regarding directed graph are as follows:

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

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

and

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

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 $w_k(C_0C_{1j})$ 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:

$$\phi_m^k = \textstyle\sum_{C_1=1}^{J}..\sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-1}, C_m) =$$
$$\textstyle\sum_{C_1=1}^{J}..\sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-2}, C_{m-1}, C_m) =$$
$$\textstyle\sum_{C_1=1}^{J}..\sum_{C_{m-1}=1}^{J} u_k(\mathbf{C}_{1:m-2}, C_{m-1})w_k(C_m)w_k(C_{m-1}C_m) =$$
$$w_k(C_m)\textstyle\sum_{C_{m-1}=1}^{J}\phi_{m-1}^k w_k(C_{m-1}C_m).$$

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

We now calculate the posteriors:

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

bioRxiv preprint doi: https://doi.org/10.1101/2020.01.29.926022; this version posted January 31, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

$$
\begin{aligned}
q(C_{km-1}=i, C_{km}=j) \quad = \quad & \sum_{C_{k,1:m-2}} \sum_{C_{k,m+1:M}}^{J} q(C_{k,1:m-2}, C_{k,m-1}=i, C_{km}=j, C_{k,m+1:M}) = \\
& \sum_{C_{1:m-2}}^{J} u_k(\mathbf{C}_{1:m-2}, C_{m-1}=i) w_k(C_{m-1i}C_{mj}) w_k(C_{mj}) \\
& \sum_{C_{m+1:M}}^{J} v_k(\mathbf{C}_{m+1:M}, C_m=j) = w_k(C_{m-1i}C_{mj}) w_k(C_{mj}) \phi^k_{m-1i} \beta^k_{mj}.
\end{aligned}
$$

A.3 Calculating expectations

Let $\boldsymbol{\Phi}$ be the digamma function defined as $\boldsymbol{\Phi}(x) = \frac{d}{dx}\log\Gamma(x)$ 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 $\boldsymbol{\Phi}$ 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
  $\mathrm{E}_{q(\theta_n)}\big(\log\theta_n\big) = \boldsymbol{\Phi}(\epsilon_{sn}) + \log E_{q(\theta_n)}(\theta_n) - \log\epsilon_{sn}$.
  Therefore, we obtain $\mathrm{E}_{q(\theta_n)}\big(D_{n,m,j}\big) = \log F_{\tilde{\theta}}(y_{nm}) + y_{nm} \times \big[\boldsymbol{\Phi}(\epsilon_{sn}) - \log\epsilon_{sn}\big]$, with $F_{\tilde{\theta}}$ being a Poisson p.m.f. with rate parameter $\tilde{\theta} = j \times (\epsilon_{sn}/\epsilon_{rn})$.
- $\mathrm{E}_{q(\boldsymbol{\pi})}(\log\pi_k) = \boldsymbol{\Phi}(\alpha_k) - \boldsymbol{\Phi}\Big(\sum_{k=1}^{K}\alpha_k\Big)$
- $\mathrm{E}_{q(\mathbf{a}^k_i)}(\log a^k_{ij}) = \boldsymbol{\Phi}(\Lambda^k_{ij}) - \boldsymbol{\Phi}\Big(\sum_{j=1}^{J}\Lambda^k_{ij}\Big)$
- $\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big) = q(C_{km}=j)$ 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. 11 and 12.