Safinianaini, Negar; De Souza, Camila P.E.; Roth, Andrew; Koptagel, Hazal; Toosi, Hosein; Lagergren, Jens

CopyMix : Mixture model based single-cell clustering and copy number profiling using variational inference

Published in:
Computational Biology and Chemistry

DOI:
10.1016/j.compbiolchem.2024.108257

Published: 01/12/2024

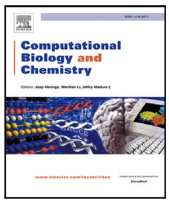
Document Version
Publisher's PDF, also known as Version of record

Computational Biology and Chemistry 113 (2024) 108257

Contents lists available at ScienceDirect

Computational Biology and Chemistry

journal homepage: www.elsevier.com/locate/cbac



Research Article

CopyMix: Mixture model based single-cell clustering and copy number profiling using variational inference

Negar Safinianaini [a,*], Camila P.E. De Souza [b], Andrew Roth [c,d], Hazal Koptagel [e], Hosein Toosi [e], Jens Lagergren [e,f]

[a] Department of Computer Science, Aalto University, Konemiehentie 2, Espoo, 02150, Helsinki, Finland
[b] Department of Statistical and Actuarial Sciences, University of Western Ontario, 1151 Richmond Street, London, N6A 5B7, Ontario, Canada
[c] British Columbia Cancer Agency, 675 West 10th Avenue, Vancouver, V5Z 1L3, BC, Canada
[d] Faculty of Computer Science, University of British Columbia, Building 201-2366 Main Mall, London, V6T 1Z4, BC, Canada
[e] Science for Life Laboratory, Tomtebodavägen 23, Solna, 171 65, Stockholm, Sweden
[f] Department of Computer Science, KTH, Malvinas v 10, Stockholm, 114 28, Stockholm, Sweden

ARTICLE INFO

Keywords:
Copy number profiling
Tumor clonal decomposition
Mixture models
Variational inference
Single-cell
Cancer

ABSTRACT

Investigating tumor heterogeneity using single-cell sequencing technologies is imperative to understand how tumors evolve since each cell subpopulation harbors a unique set of genomic features that yields a unique phenotype, which is bound to have clinical relevance. Clustering of cells based on copy number data obtained from single-cell DNA sequencing provides an opportunity to identify different tumor cell subpopulations. Accordingly, computational methods have emerged for single-cell copy number profiling and clustering; however, these two tasks have been handled sequentially by applying various ad-hoc pre- and post-processing steps; hence, a procedure vulnerable to introducing clustering artifacts. We avoid the clustering artifact issues in our method, CopyMix, a Variational Inference for a novel mixture model, by jointly inferring cell clusters and their underlying copy number profile. Our probabilistic graphical model is an improved version of the mixture of hidden Markov models, which is designed uniquely to infer single-cell copy number profiling and clustering. For the evaluation, we used likelihood-ratio test, CH index, Silhouette, V-measure, total variation scores. CopyMix performs well on both biological and simulated data. Our favorable results indicate a considerable potential to obtain clinical impact by using CopyMix in studies of cancer tumor heterogeneity.

1. Introduction

A tumor typically consists of a collection of heterogeneous cell populations, each having distinct genetic and phenotypic properties, in particular, concerning the capacity to promote cancer progression, metastasis, and therapy resistance (Eirew et al., 2015; Nowell, 1976). Single-cell sequencing technologies (Gawad et al., 2016; Navin et al., 2011; Shapiro et al., 2013; Zahn et al., 2017) provide an opportunity to investigate the genomic profile of individual cells regarding both single nucleotide variation (SNV) and copy number variation (CNV). CNVs and SNVs are essential contributors to phenotypic variation relating to health, and disease (Baslan et al., 2012; Lawson et al., 2018). Although single-cell SNV profiling is hampered by experimental imperfections such as drop-outs, copy number profiling, i.e., detecting single-cell CNVs, is feasible, at least at coarser resolutions. Clustering cells based on their copy number profiles improves understanding of tumor subpopulations and tumor heterogeneity, issues bound to have clinical relevance.

Current single-cell datasets pose a wealth of computational challenges. As answers to some of those, methods have emerged for single-cell copy number profiling and clustering; some methods infer clustering after copy number profiling, e.g., Garvin et al. (2015), Zahn et al. (2017), Leung et al. (2017), Vitak et al. (2017), Zaccaria and Raphael (2021), while a recent method, CONET (Markowska et al., 2022), derives copy number profiles by post-processing in addition to a user-defined sequence of breakpoints per cell. Unfortunately, these methods perform single-cell copy number profiling and clustering sequentially with various ad-hoc processes. The sequential approach is vulnerable to artifacts since preprocessing decisions, typically irreversible, constrain all later analyses. Even if each task performs optimally, the final result may still fall short of the best possible performance (Blocker and Meng, 2013). Another problem is when using HMMcopy (Shah et al., 2006; Vitak et al., 2017; Zahn et al., 2017) for copy number profiling–copy number profiles can naturally be modeled by a sequence of latent variables forming Hidden Markov Models (HMMs)–that has

* Corresponding author.
E-mail address: negar.safinianaini@aalto.fi (N. Safinianaini).

https://doi.org/10.1016/j.compbiolchem.2024.108257
Received 17 June 2024; Received in revised form 15 August 2024; Accepted 15 October 2024
Available online 23 October 2024

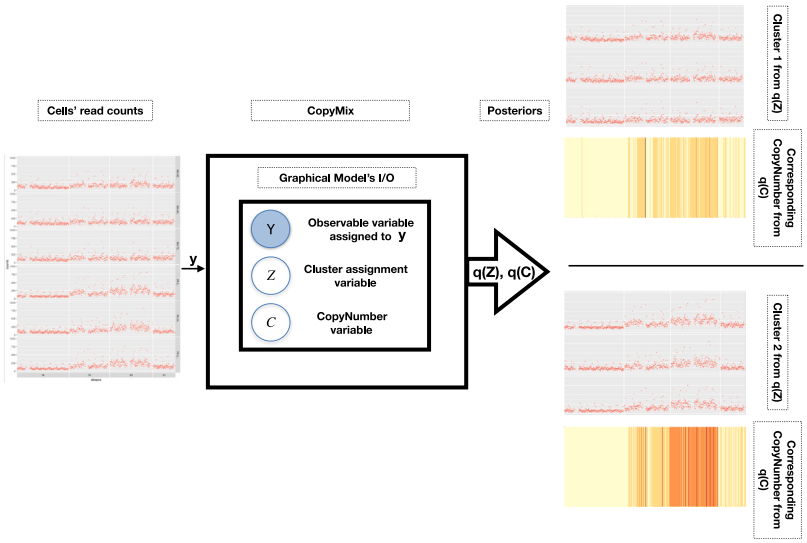


Fig. 1. The overview of CopyMix for binary clustering: input y, cells' reads; outputs $q(Z)$ and $q(C)$, approximated posterior distributions resulting in clusters within y with their corresponding copy number profiles illustrated by heatmaps.

limitations such as requiring manual calibration of more than ten parameters (Mallory et al., 2020). Moreover, as addressed recently by CONET (Markowska et al., 2022), one should account for the fact that copy number profiles are generated by a clonal process; however, mixtures of Hidden Markov Models (MHMMs) (Smyth, 1997), a joint inference alternative for performing the two biological tasks, consider a different copy number profile per cell, thus lacking the clonal copy number profiling.

By a joint inference solution, also motivated by future directions of computational modeling in single-cell cancer genomics (Zhang and Campbell, 2020), we propose a novel framework referred to as CopyMix to alleviate three problems: (1) the sequential treatment of copy number profiling and clustering; (2) the labor of HMMcopy parameter tuning; (3) lack of clonal copy number profiling by MHMMs.

CopyMix performs the joint inference of copy number profiling and clustering, and as to the best of our knowledge, no earlier work has simultaneously performed these two tasks. Similarly, joint inference has been considered for single-cell DNA methylation (Kapourani and Sanguinetti, 2019; de Souza et al., 2020), single-cell SNV data (Roth et al., 2016), and bulk chIP-seq data from several replicates (Zuo et al., 2016). Due to the model-based treatment, CopyMix enjoys the advantages of a fully probabilistic framework, such as transparency, uncertainty measurements, and modeling flexibility. Copy number profiles can naturally be modeled by a sequence of latent variables following 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). However, as addressed recently by CONET (Markowska et al., 2022), one should account for the fact that copy number profiles are generated by a clonal process; as mixtures of Hidden Markov Models (MHMMs) (Smyth, 1997) consider a copy number profile per cell, MHMMs lack the clonal copy number profiling. CopyMix, a biologically meaningful tangent of MHMMs, uses a novel mixture model with components (expressing the clonal process as opposed to cell-specific MHMMs) corresponding to clusters, each having a specific copy number profile, revealing the copy number variation pattern behind each cluster. We deploy a Bayesian treatment and infer all quantities of interest using Variational Inference (VI) (Jordan et al., 1999), which typically yields faster inference methods than methodologies like Markov chain Monte Carlo sampling (Blei et al., 2017). Compared to Expectation-Maximization (EM), VI has multiple advantages; e.g., it estimates the posterior distributions rather than point estimates, protects against over-fitting and allows for principled model selection, i.e., identifying the optimal number of mixture components (Bishop, 2006). Finally, by using graphs, our novel VI simplifies the normalization for the posterior approximation of the HMM part, while earlier methods (McGrory and Titterington, 2009) use complex solutions for the normalization part.

2. Material and methods

2.1. CopyMix

CopyMix[1] 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 discrete time-homogeneous Markov chain—this Markov chain describes a sequence of possible copy numbers in which the probability of each copy number, corresponding to a state, depends only on the previous state in the sequence. Fig. 1 contains an overview of CopyMix; the CopyMix inputs are sequences of read count ratios per genomic bin, and the outputs are clusters of cells with their corresponding copy number sequences (profiles).

Our input data, read count ratios (or read ratios), are assumed to be GC-corrected—a necessary bias correction due to the dropping of read coverage at the regions with extreme GC contents (Yoon et al., 2009). The read counts are integers, and the GC-corrected read counts become positive real numbers (read ratios) through the GC-correction process. We consider read ratios over M fixed, typically equal-sized, genomic bins, as in Leung et al. (2017), Laks et al. (2019). We assume that the read ratios follow a Gaussian distribution; for the details supporting this choice, see Appendix F. Moreover, we presume that the read ratios are emitted from a latent sequence of copy number states. The probabilistic graphical model is illustrated in Fig. 2. The read ratios are modeled as a Gaussian distribution with independent conjugate priors; that is, the mean of the Gaussian follows a Gaussian distribution and the precision of the Gaussian follows a Gamma distribution. Y denotes the observable variables, read ratios per predefined bins, C the latent copy number states forming a Markov chain, and Z the latent cell-specific cluster assignment variables. As shown in Fig. 2, each Y has two levels of dependencies, which are reflections of the assumption that the Gaussian distribution over the read ratios depends on a latent cell-specific cluster assignment, Z, and a corresponding latent copy number

[1] For implementation, see https://github.com/negar7918/CopyMix/.

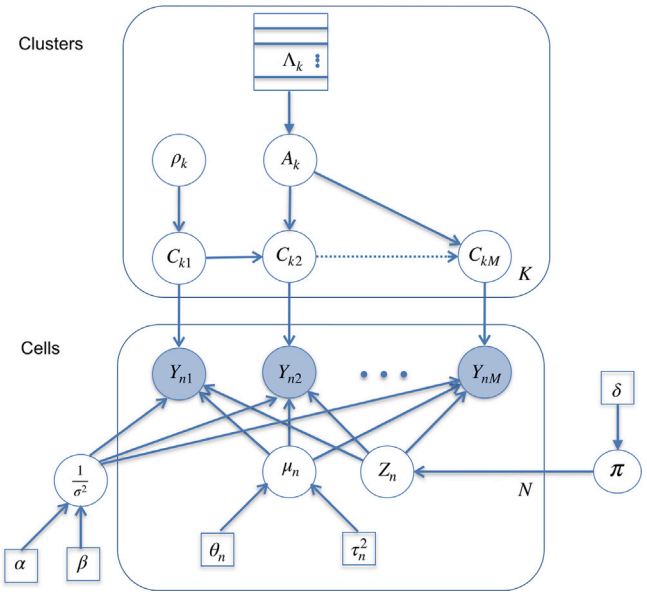


Fig. 2. Probabilistic graphical model behind CopyMix is shown, where shaded nodes are observed values, the unshaded ones are the latent variables, and the squares are the hyperparameters; a posterior distribution over the values of the unshaded nodes is approximated using Variational Inference. Y_{nm} is an observed read count ratio from cell n and bin m; C_{km}, corresponds to a latent copy number state, where C_k forms a Markov chain; μ_n, is a cell-specific rate; Z_n is a latent cell-specific cluster assignment variable. Finally, π and A_k, are the conjugate priors of Z_n and C_{km} respectively, and ρ_k is the initial probability.

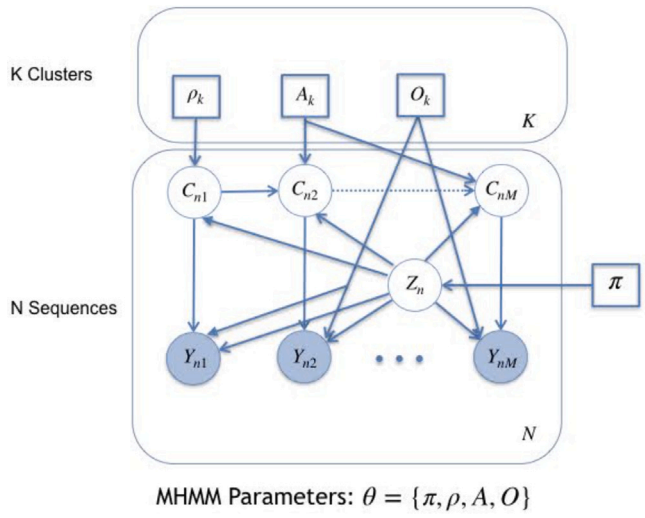


Fig. 3. Probabilistic graphical model of the mixture of hidden Markov models.

state, C. Intuitively, a higher copy number should correspond to a higher read ratio. We incorporate this belief in our model by defining the mean of a Gaussian distribution as the product of copy number state (C_{km}) and cell-specific mean (μ_n). The use of the multiplicative structure can be found in a recent work in which the mean is dependent on copy number events by a multiplication operation (Malekpour et al., 2018). This also implies that μ_n corresponds to the average sequencing coverage for a haploid genome. Due to the multiplicative structure, a copy number of zero produces the lowest mean, implying a copy number deletion event. In what follows, our model is described in more detail.

Let Y_{nm} be the observed GC-corrected reads ratio of bin m for cell n for $n = 1, \dots, N$ and $m = 1, \dots, M$, and $\mathbf{Y}_n = (Y_{n1}, \dots, Y_{nM})$ be a vector of observed data for cell n. The cluster membership of cell n is indicated by the hidden variable Z_n that takes values in $[K] = \{1, \dots, K\}$. We assume there are $K \ll N$ hidden copy number sequences, one for each cluster. The variables $Z_1, \dots, Z_N$ 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 kth copy number sequence, defined as $C_k = (C_{k1}, \dots, C_{kM})$, with each C_{km} taking values in $[J] = \{1, \dots, J\}$. We assume that $C_{k1}, \dots, C_{kM}$ follows a discrete time-homogeneous Markov chain with initial probabilities $\rho_{kj} = P(C_{k1} = j)$ 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 copy number sequence, $Y_{n1}, \dots, Y_{nM}$ are independent with Y_{nm} following a distribution with parameters depending on the hidden true state at bin m for cluster k, that is, $Y_{nm} | Z_n = k, C_{km} = j$. As assumed, the read ratios follow Gaussian distribution, i.e., $F_{j\mu_n,\sigma^2}(Y_{nm}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(Y_{nm} - j \times \mu_n)^2}{2\sigma^2}}$.

Note that one needs B-allele frequencies (BAF) to perform ploidy estimation, and the methods not using BAF data apply restrictive assumptions to select among many equally plausible solutions to estimated ploidy (Zaccaria and Raphael, 2021); this may result in selecting copy numbers that contradict the underlying allelic balance/imbalance. In the Concluding remarks, we refer to allele-based CopyMix as a future work. In the Results section, we account for chromosomal copy numbers by assigning initial probabilities to the beginning of each chromosome.

As aforementioned, MHMMs have another probabilistic graphical model that is not suitable for our biological task here. That is, every cell has a different copy number profile (sequence of C) assigned to it. The MHMM model is illustrated in Fig. 3, and the most critical difference to the probabilistic graphical model of CopyMix is that the sequence of C is in the bottom plate N while in CopyMix, the sequence of C is in the top plate K.

For the detailed derivation of our novel VI, see Appendix A, B, and C.

2.2. Evaluation of clustering and copy number profile estimates

Here, we describe the metrics used in the Results section. We assess the clustering performance of CopyMix using the well-accepted V-measure (Rosenberg and Hirschberg, 2007), CH Index (Calinski and Harabasz, 1974), Silhouette (Rousseeuw, 1987) and likelihood ratio test. Total variation (Sirazdinov and Mamatov, 1962; Alison and Edward Su, 2002) is the metric that we use to evaluate CopyMix copy number inference results.

One standard distance measure for comparing probabilities is *total variation* that can be used to measure the distance between copy number proportions (Zaccaria and Raphael, 2020). It is computed by $TV(C, C') = \frac{1}{2} \sum_{m \in M} |C_m - C'_m|$ with two sequences C, C', where C_m and C'_m are copy number probabilities at position m obtained by normalizing a sequence of copy numbers across the sequence. TV is zero when the similarity between sequence variations is highest.

3. Theory

In this section, we provide the novelty of our inference for the HMMs. Instead of calculating the normalization term done previously (McGrory and Titterington, 2009) requiring complex calculations, we treat the HMM as a graph with weights instead of probabilities. Subsequently, the normalization is easily applied to the resulting graph weights to convert them to probabilities.

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 $\log q(\mathbf{C}_k)$). Finally, we have the log of Gaussian likelihood defined as

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

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

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

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

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 defined as $w_k(C_0C_{1j})$ assuming the source starts at 1. Notice that k shows the cluster to which the graph belongs. We can calculate forward and backward 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: $w_k(C_0C_{1j}) = \exp\{\mathrm{E}_{q(\rho_k)}(\log\rho_{kj})\}$; $w_k(C_{m-1i}C_{mj}) = \exp\{\mathrm{E}_{q(\mathbf{a}_i^k)}(\log a_{ij}^k)\}$ and $w_k(C_{mj}) = \exp\{\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mu_n),q(\frac{1}{\sigma^2})}(D_{n,i,s})\}$. Note that we skip writing i, j, k in the calculations to make them short and more readable:

$$\begin{aligned}
\phi_m &= \sum_{C_1=1}^{J}..\sum_{C_{m-1}=1}^{J}u_k(\mathbf{C}_{1:m-1}, C_m) = \sum_{C_1=1}^{J}..\sum_{C_{m-1}=1}^{J}u_k(\mathbf{C}_{1:m-2}, C_{m-1}, C_m) =\\
&\quad \sum_{C_1=1}^{J}..\sum_{C_{m-1}=1}^{J}u_k(\mathbf{C}_{1:m-2}, C_{m-1})w(C_m)w(C_{m-1}C_m)\\
&= w(C_m)\sum_{C_{m-1}=1}^{J}\phi_{m-1}w(C_{m-1}C_m)\\
\beta_m &= \sum_{C_{m+1}=1}^{J}..\sum_{C_M=1}^{J}v_k(\mathbf{C}_{m+1:M}, C_m)\\
&= \sum_{C_{m+1}=1}^{J}..\sum_{C_M=1}^{J}v_k(\mathbf{C}_{m+2:M}, C_{m+1}, C_m) =\\
&\quad \sum_{C_{m+1}=1}^{J}..\sum_{C_M=1}^{J}v_k(\mathbf{C}_{m+2:M})w(C_mC_{m+1})w(C_{m+1})\\
&= \sum_{C_{m+1}=1}^{J}\beta_{m+1}w(C_mC_{m+1})w(C_{m+1})
\end{aligned}$$

We now calculate the posteriors:

$$\begin{aligned}
q_k(C_m = j) &= \sum_{C_{1:m-1}}^{J}\sum_{C_{m+1:M}}^{J}q_k(C_{1:m-1}, C_m = j, C_{m+1:M})\\
&= \sum_{C_{1:m-1}}^{J}u_k(\mathbf{C}_{1:m-1}, C_m = j)\sum_{C_{m+1:M}}^{J}v_k(\mathbf{C}_{m+1:M}, C_m = j)\\
&= \phi_{mj}\beta_{m,j}
\end{aligned}$$

$$\begin{aligned}
&q_k(C_{m-1} = i, C_m = j)\\
&= \sum_{C_{1:m-2}}^{J}\sum_{C_{m+1:M}}^{J}q_k(C_{1:m-2}, C_{m-1} = i, C_m = j, C_{m+1:M}) =\\
&\quad \sum_{C_{1:m-2}}^{J}u_k(\mathbf{C}_{1:m-2}, C_{m-1} = i)w(C_{m-1i}C_{mj})w(C_{mj})\\
&\quad \times \sum_{C_{m+1:M}}^{J}v_k(\mathbf{C}_{m+1:M}, C_m = j) =\\
&\quad w(C_{m-1i}C_{mj})w(C_{mj})\phi_{m-1j}\beta_{m,j}
\end{aligned}$$

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

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

$$w_k(C_{m-1i}C_{mj}) = \exp\{\mathrm{E}_{q(\mathbf{a}_i^k)}(\log a_{ij}^k)\} \quad (2)$$

$$w_k(C_{mj}) = \exp\Big\{\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)E_{q(\mu_n),q(\frac{1}{\sigma^2})}\big(D_{n,m,j}\big)\Big\} \quad (3)$$

$$q(C_{km} = j) = \frac{\phi_{mj}^k\beta_{mj}^k}{\sum_{j=1}^{J}\phi_{mj}^k\beta_{mj}^k} \quad (4)$$

$$q(C_{km-1} = i, C_{km} = j) = \frac{w_k(C_{m-1i}C_{mj})w_k(C_{mj})\phi_{m-1i}^k\beta_{mj}^k}{\sum_{j=1}^{J}w_k(C_{m-1i}C_{mj})w_k(C_{mj})\phi_{m-1i}^k\beta_{mj}^k} \quad (5)$$

It can be observed that the weights of the vertex and edge of the graph are provided in Eqs. (2) and (3). Finally, the normalizations are simply done in Eqs. (4) and (5) by merely dividing the graph weights terms in the numerators to the summation of those.

4. Results

4.1. Experimental setup

4.1.1. Dataset

To adequately evaluate CopyMix, as opposed to earlier literature lacking the evaluation against ground truth and hence subjective, we conduct experiments on simulated data, where we have access to the ground truth. We also evaluate CopyMix on two biological datasets: ovarian cancer (DLP data obtained from https://zenodo.org/record/3445364#.YegRmljMLzW) (Laks et al., 2019) and colorectal cancer (CRC data) (Leung et al., 2017). The DLP data are ovarian cancer single-cells generated using the DLP technology, a scalable single-cell whole-genome sequencing platform. The DLP data are decomposed into three cancer *cell lines* derived from the same patient, sourced

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

CONF 1	CONF 2	CONF 3	CONF 4	CONF 5	CONF 6
100% – 0.09	100% – 0.05	100% – 0.05	100% – 0.20	0% – 0.11	100% – 2.45
CONF 7	**CONF 8**	**CONF 9**	**CONF 10**	**CONF 11**	**CONF 12**
100% – 2.45	100% – 2.45	100% – 2.36	100% – 2.45	87% – 1.02	100% – .96
CONF 13	**CONF 14**	**CONF 15**	**CONF 16**	**CONF 17**	**CONF 18**
100% – 0.93	88% – 1.20	87% – 0.96	100% – 1.03	100% – 1.04	100% – 1

from one primary tumor, and two relapse specimens. Following the original paper, Laks et al. (2019), we conduct experiments on the 891 cells that are GC-corrected, obtained by HMMcopy. The CRC data contain single-cells belonging to primary and metastatic tumors, and the copy number profiling has been performed by the original paper by using Circular Binary Segmentation (CBS) change point detection method—an alternative way to HMMs.

We simulate data under various clustering scenarios, varying the number of clusters and copy number profiles. For details on the simulated data, see Appendix D.

4.1.2. Experimental protocol

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

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

4.2. Experimental results

4.2.1. Simulated data

We evaluated the performance of CopyMix on 18 simulated dataset configurations, see Fig. 4. In Table 1, we report V-measure for each configuration. We can observe that CopyMix, except for CONF 5, has a good clustering performance with V-measures ranging from 87% to 100%, Table 1. The exception, i.e., CONF 5, consists of two simulated genomes where one is a whole-genome duplication (WGD) of the other, *a whole-genome duplication pair*. Typically, when a genome undergoes a whole-genome duplication, the copy numbers across the sequence are scaled to a larger value, i.e., usually, diploid becomes tetraploid. This issue of CopyMix in distinguishing the ploidy can be explained partially by the notion of *unidentifiability* – inability to derive the true labels in unsupervised learning – of the transition matrix in a Markov chain (Murphy, 2012).

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

Regarding the run-time performance of CopyMix, we observed that the run-time increases linearly by increasing the number of clusters. For the 2, 3, 4, and 5 clusters, CopyMix run-time is 9, 11, 14, and 15 s, given 32 CPU cores. The maximum increase is from 2 to 5 clusters, which has a linear increase by a factor that is 1.6. The run-time is constant for different numbers of cells as long as it is less than or equal to the number of CPU cores (due to the parallelism of the Python code); otherwise, if the number of cells is greater than the number of CPU cores, then the run-time increases linearly by the proportion of the cells and cores.

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

4.2.2. Biological data

We evaluated CopyMix on the DLP data (Laks et al., 2019), where each genome is partitioned into 6206 bins (bin size of 500K). [2] The clusters should agree with cell lines; no cluster contains cells from two different cell lines. It turns out that CopyMix succeeds in clustering the main clusters since the V-measure when comparing CopyMix clusters with the cell lines is 98%. The V-measure of overall CopyMix clusters based on those reported by the original clusters is 67%.

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

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

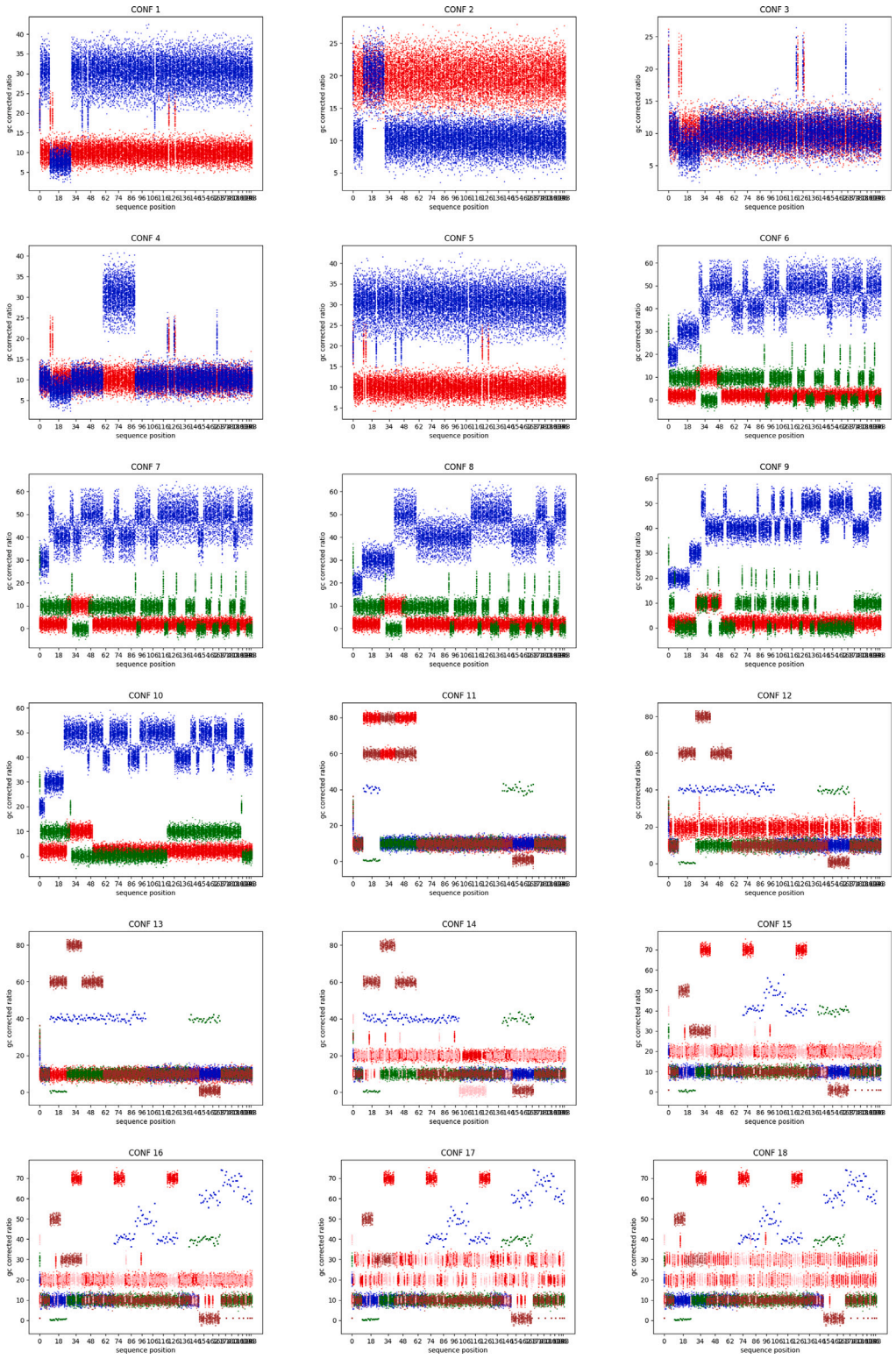


Fig. 4. The datasets for CONF 1 to 18 are shown. CONF 1 to 5, CONF 6 to 10, CONF 11 to 13, and CONF 14 to 18 correspond to scenarios where 2, 3, 4, and 5 clusters are color-coded, respectively.

Table 2 shows that CopyMix outperforms the original paper w.r.t. clustering according to CH Index and Silhouette metrics. We also evaluated Silhouette for fewer clusters reported by the original paper; these clusters are obtained according to the reported phylogenetic tree in the original paper (Laks et al., 2019). Regarding Silhouette, even though the CopyMix score, 0.49, is not the perfect score, it is far better than those obtained by the method in the original paper. These metrics show that the method in the original paper has detected overlapping clusters, which is not sufficiently supported by the data. The SNV-CopyMix did not detect more clusters due to shallow data; for the details, see Appendix G. Despite the unchanging effect of the SNV inclusion for the DLP data, there is generally motivation to combine the two signals; new mutations and, subsequently, complex phenotypes can be caused by combined CNV-SNV signals (Lobo, 2008; Freeman et al., 2006).

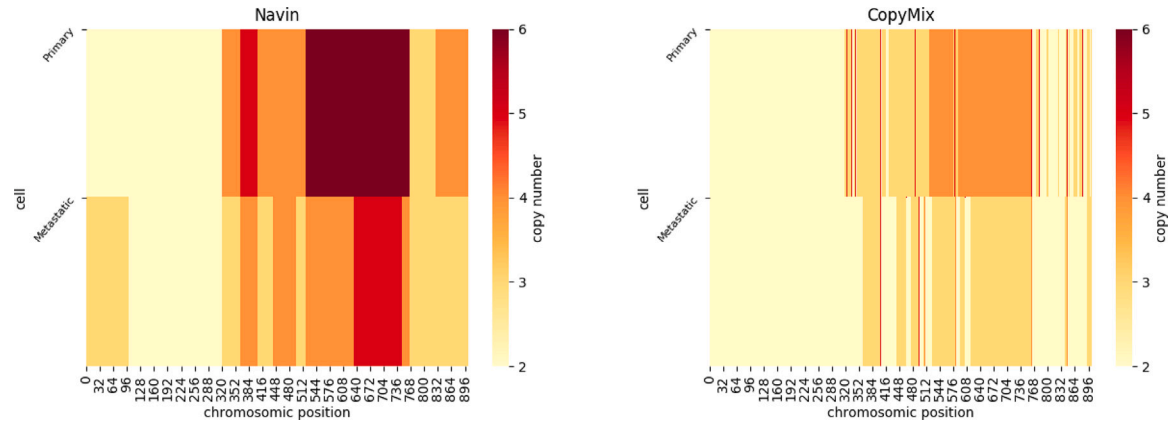


Fig. 5. Copy number profiling comparison between the results from Navin (Leung et al., 2017) (*LHS*) and CopyMix (*RHS*) for both primary and metastatic clusters.

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

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

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

It is important to emphasize that the clusters reported by the original paper are not ground truth; hence, we cannot conclude that any diverging result is poor.

Next, we compared CopyMix's clustering results with those of Ginkgo. See Appendix E for details on how CopyMix outperforms Ginkgo.

To evaluate CopyMix on another biological dataset and against a different method, a CBS-based method, we compared CopyMix to the results of a pipeline (Leung et al., 2017): CBS-based copy number profiling followed by a hierarchical clustering on CRC data. 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). As shown in Fig. 5, 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).

Finally, to further justify our claim on the criticality of joint inference of copy numbers and clusters, we created a pipeline by running an Expectation-Maximization algorithm for HMMs followed by performing K-means clustering over the inferred copy numbers, resembling (Shah et al., 2006; Vitak et al., 2017) but excluding their ad-hoc processes. The result on the CRC data was poor in clustering (V-measure of 0.24), while CopyMix gained a V-measure of 1. This result confirms that the joint inference of copy number profiles and clustering is superior to the sequential approach.

5. Concluding remarks

We introduced CopyMix, a novel mixture model to jointly perform single-cell copy number profiling and clustering. CopyMix, while enjoying the advantages of Bayesian inference, addresses the following issues: (1) the sequential treatment of copy number profiling and single-cell clustering, prone to introduce clustering artifacts; (2) the labor of HMMcopy parameter tuning; (3) the lack of clonal copy number profiling by MHMMs; (4) the complicated posterior approximation when deriving VI for HMMs. We evaluated our approach on both simulated and biological data, which indicated that CopyMix performs well when estimating both single-cell clustering and the corresponding copy number profiles.

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

CRediT authorship contribution statement

Negar Safinianaini: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Camila P.E. De Souza:** Writing – review & editing, Writing – original draft, Supervision, Resources, Methodology, Investigation, Funding acquisition. **Andrew Roth:** Methodology, Investigation. **Hazal Koptagel:** Data curation. **Hosein Toosi:** Formal analysis. **Jens Lagergren:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

In this Appendix, a short overview of VI for CopyMix is described.

As aforementioned, we let $\boldsymbol{\Psi}$ be the set containing all the unknown model parameters, i.e., $\boldsymbol{\Psi} = \{\mathbf{A}, \boldsymbol{\mu}, \sigma^2, \boldsymbol{\pi}\}$, where

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

We, w.l.o.g., consider the initial probabilities, ρ_{kj}'s, to be fixed and known. We use the following prior distributions for the parameters in $\boldsymbol{\Psi}$.

- $\mathbf{a}_i^k = (a_{i1}^k, \dots, a_{iJ}^k) \sim \text{Dirichlet}(\boldsymbol{\Lambda})$

- $\mu_n \sim \text{Normal}(\theta_n, \tau_n^2)$. The conjugate prior concerning the mean of a Gaussian distribution is Gaussian distribution.
- $\frac{1}{\sigma^2} \sim \text{Gamma}(\alpha, \beta)$. The conjugate prior concerning the precision of a Gaussian distribution is Gamma distribution.
- $\boldsymbol{\pi} \sim \text{Dirichlet}(\boldsymbol{\delta})$

In order to infer $\boldsymbol{\Psi}$, the hidden states $\mathbf{C} = \{\mathbf{C}_1, \dots, \mathbf{C}_K\}$, and $\mathbf{Z} = (Z_1, \dots, Z_n)$ 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) given by

$$\text{ELBO}(q) = \text{E}\big[\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi})\big] - \text{E}\big[\log q(\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi})\big]. \tag{A.1}$$

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

Step 1. VD factorization

We assume the following factorization of the VD:

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

Note that Z and C are dependent in the model, which is meaningful because the clustering is based on the copy number sequence C. In the mean-field approximation, they are independent due to the mean-field assumption for the factorization of the variational distribution. However, the update equation for Z_n (explained later in the Appendix) depends on the VD-based expected values of copy number states C_k and the update equation for C_k depends on the VD-based expected values of the clustering assignments.

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 B.

Step 3. VD computation

We now derive a coordinate ascent algorithm for the VD; we derive an update equation for each term in the factorization, Eq. (A.2), by calculating the expectation of $\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi})$ over the VD of all random variables except the one currently being updated (Bishop, 2006). See Appendix B for the update equation of each term in Eq. (A.2).

Step 4. Summary of updates

We update the parameters of each variational distribution presented in Step 3; for the details of the update equations and the CopyMix algorithm, see Appendix B. A shorter version of the CopyMix algorithm is shown in Alg. 1. As shown in lines 8 and 9 of Alg. 1, we emphasize that the main output of CopyMix is the approximated posterior distributions of cluster assignments and copy number states. Note that the update equations in line 5 of Alg. 1 are performed sequentially motivated by coordinate ascent (Tseng, 2001).

Algorithm 1 CopyMix: a Variational Inference Algorithm

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

Appendix B

In this Appendix, the detailed VI derivations for CopyMix are provided. The logarithm of the joint distribution of $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{C}$ and $\boldsymbol{\Psi}$ satisfies

$$\log P(\mathbf{Y}, \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) = \log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) + \log P(\mathbf{C}|\boldsymbol{\Psi}) + \log P(\mathbf{Z}|\boldsymbol{\Psi}) + \log P(\boldsymbol{\Psi}), \tag{B.3}$$

where

$$\begin{aligned}\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) &= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \text{I}(Z_n = k)\text{I}(C_{km} = j)\log F_{\mu_{nj},\sigma^2}(r_{nm}) \\ &= \sum_{n,m,k,j} \text{I}(Z_n = k)\text{I}(C_{km} = j)(-\frac{1}{2}) \\ &\quad \times \Big[\log(2\pi\sigma^2) + \frac{(r_{nm} - j\mu_n)^2}{\sigma^2}\Big], \end{aligned} \tag{B.4}$$

$$\begin{aligned}\log P(\mathbf{C}|\boldsymbol{\Psi}) = \sum_{k=1}^{K}\Bigg[&\sum_{j=1}^{J} \text{I}(C_{k1} = j)\log \rho_{kj} \\ &+ \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J} \text{I}(C_{km-1} = i, C_{km} = j)\log a_{ij}^k\Bigg], \end{aligned} \tag{B.5}$$

and

$$\log P(\mathbf{Z}|\boldsymbol{\Psi}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \text{I}(Z_n = k)\log \pi_k \tag{B.6}$$

Moreover,

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

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

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

$$\log P(\boldsymbol{\mu}) = \sum_{n=1}^{N}\Big[(-\frac{1}{2})\Big[\log(2\pi\tau^2) + \frac{(\mu_n - \theta)^2}{\tau^2}\Big]\Big], \tag{B.7}$$

$$\log P(\frac{1}{\sigma^2}) = \Big[-\frac{\beta}{\sigma^2} + (\alpha - 1)\log(\frac{1}{\sigma^2})\Big] + C, \tag{B.8}$$

$$\log P(\mathbf{A}) = \sum_{k=1}^{K}\sum_{i=1}^{J}\Big[\sum_{j=1}^{J}(\Lambda_j - 1)\log a_{ij}^k - \log B(\boldsymbol{\Lambda})\Big], \tag{B.9}$$

and

$$\log P(\boldsymbol{\pi}) = \sum_{k=1}^{K}(\delta_k - 1)\log \pi_k - \log B(\boldsymbol{\delta}). \tag{B.10}$$

Update equation for π: Since (B.6) and (B.10) are the only terms in (B.3) that depend on $\boldsymbol{\pi}$, the update equation $q(\boldsymbol{\pi})$ can be derived as follows.

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

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

$$\delta_k = \delta_k^0 + \sum_{n=1}^{N}\text{E}_{q(Z_n)}\big(\text{I}(Z_n = k)\big). \tag{B.11}$$

Update equation for Z_n: Since (B.4) and (B.6) are the only terms in (B.3) 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.,

$$\begin{aligned}\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\boldsymbol{\Psi}) &= \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{I}(Z_n=k)\mathrm{I}(C_{km}=j)(-\frac{1}{2})\\&\quad\times\Big[\log(2\pi\sigma^2)+\frac{(r_{nm}-j\mu_n)^2}{\sigma^2}\Big]\\&+\sum_{l\neq n}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{I}(Z_l=k)\mathrm{I}(C_{km}=j)(-\frac{1}{2})\Big[\log(2\pi\sigma^2)+\frac{(r_{lm}-j\mu_l)^2}{\sigma^2}\Big]\end{aligned}\tag{B.12}$$

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,

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

We conclude that $q(Z_n)\sim$ 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

$$p_{nk}=\frac{\exp(\tilde{p}_{nk})}{\sum_{k'=1}^{K}\exp(\tilde{p}_{nk'})}.\tag{B.13}$$

Update equation for μ: Since (B.4) and (B.7) are the only terms in (B.3) that depend on μ_n, $q(\mu_n)$ can be obtained as follows.

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

We can write $\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\boldsymbol{\Psi})$ as in Eq. (B.4), and $\mathrm{E}_{-\mu_n}\big(\log P(\boldsymbol{\mu})\big)=\log P(\boldsymbol{\mu})$, which is:

$$\begin{aligned}\log P(\boldsymbol{\mu})&=(-\frac{1}{2})\Big[\log(2\pi\tau^{02})+\frac{(\mu_n-\theta^0)^2}{\tau^{02}}\Big]\\&\quad+\sum_{l\neq n}(-\frac{1}{2})\Big[\log(2\pi\tau^{02})+\frac{(\mu_l-\theta^0)^2}{\tau^{02}}\Big]\end{aligned}$$

Therefore,

$$\begin{aligned}&\log q(\mu_n)\\&\overset{+}{\approx}-\frac{1}{2}\Bigg((\frac{\mu_n^2-2\theta^0\mu_n}{\tau^{02}})+\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\\&\quad\times\mathrm{E}_{q(\frac{1}{\sigma^2})}[\frac{j^2\mu_n^2-2j\mu_n r_{nm}}{\sigma^2}]\Bigg)\\&\overset{+}{\approx}-\frac{1}{2}\Bigg((\frac{\mu_n^2-2\theta^0\mu_n}{\tau^{02}})+\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\\&\quad\times\frac{\alpha}{\beta}(j^2\mu_n^2-2j\mu_n r_{nm})\Bigg)\\&\overset{+}{\approx}-\frac{1}{2}\Bigg(\mu_n^2\Big[\frac{1}{\tau^{02}}+\frac{\alpha}{\beta}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)j^2\Big]\\&\quad-2\mu_n\Big[\frac{\theta^0}{\tau^{02}}+\frac{\alpha}{\beta}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)jr_{nm}\Big]\Bigg)\end{aligned}$$

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

$$\theta_n=\frac{\frac{\theta^0}{\tau^{02}}+\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)jr_{nm}\frac{\alpha}{\beta}}{\frac{1}{\tau^{02}}+\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\frac{\alpha}{\beta}j^2}\tag{B.14}$$

$$\tau_n^2=\frac{1}{\frac{1}{\tau^{02}}+\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\frac{\alpha}{\beta}j^2}\tag{B.15}$$

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

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

We can write $\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\boldsymbol{\Psi})$ and $\log P(\frac{1}{\sigma^2})$ as they are in Eqs. (B.4) and (B.8). Putting them together, we achieve the following.

$$\begin{aligned}&\log q(\frac{1}{\sigma^2})\\&\overset{+}{\approx}\big[-\frac{\beta^0}{\sigma^2}+(\alpha^0-1)\log(\frac{1}{\sigma^2})\big]+\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\\&\quad\times\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\mathrm{E}_{q(\mu_n)}[D_{nmj}]\\&\overset{+}{\approx}-\frac{\beta^0}{\sigma^2}+(\alpha^0-1)\log(\frac{1}{\sigma^2})\\&\quad-\frac{1}{2}\sum_{n,m,k,j}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\Big[-\log(\frac{1}{\sigma^2})\\&\quad+\frac{r_{nm}^2-2jr_{nm}\theta_n+j^2(\tau_n^2+\theta_n^2)}{\sigma^2}\Big]\\&=\log(\frac{1}{\sigma^2})\Big\{\alpha-1\Big\}-\frac{1}{\sigma^2}\Big\{\beta\Big\}\end{aligned}$$

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

$$\alpha=\alpha^0+\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)=\alpha^0+\frac{NM}{2}\tag{B.16}$$

$$\begin{aligned}\beta&=\beta^0+\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\\&\quad\times\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km}=j)\big)\Big(r_{nm}^2-2jr_{nm}\theta_n+j^2(\tau_n^2+\theta_n^2)\Big)\end{aligned}\tag{B.17}$$

Update equation for $\mathbf{a}_i^k$: Because (B.5) and (B.9) are the only terms in (B.3) that depend on $\mathbf{a}_i^k$, we calculate $q(\mathbf{a}_i^k)$ as follows.

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

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:

$$\log q(\mathbf{a}_i^k)\overset{+}{\approx}\sum_{j=1}^{J}\sum_{m=2}^{M}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i,C_{km}=j)\big)\log a_{ij}^k$$

$$+\sum_{j=1}^{J}(\Lambda_j^0-1)\log a_{ij}^k$$
$$=\sum_{j=1}^{J}\log a_{ij}\Big\{\Big[\Lambda_j^0+\sum_{m=2}^{M}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i,C_{km}=j)\big)\Big]-1\Big\}.$$

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

$$\Lambda_{ij}^k=\Lambda_j^0+\sum_{m=2}^{M}\mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km-1}=i,C_{km}=j)\big) \tag{B.18}$$

Update equation for $\mathbf{C}_k$: Since only (B.4) and (B.5) depend on $\mathbf{C}_k$, $\log q(\mathbf{C}_k)$ can be calculated as follows.

$$\log q(\mathbf{C}_k)$$
$$=\mathrm{E}_{-\mathbf{C}_k}\big(\log P(\mathbf{Y},\mathbf{Z},\mathbf{C},\boldsymbol{\Psi})\big)$$
$$\overset{+}{\approx}\mathrm{E}_{-\mathbf{C}_k}\big(\log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\boldsymbol{\Psi})\big)+\mathrm{E}_{-\mathbf{C}_k}\big(\log P(\mathbf{C}|\boldsymbol{\Psi})\big).$$

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

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

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

$$\log q(\mathbf{C}_k)$$
$$\overset{+}{\approx}\sum_{m=1}^{M}\sum_{j=1}^{J}\mathrm{I}(C_{km}=j)\Big\{\sum_{n=1}^{N}\mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n=k)\big)\mathrm{E}_{q(\mu_n),q(\frac{1}{\sigma^2})}(D_{n,m,j})\Big\}$$
$$+\sum_{j=1}^{J}\mathrm{I}\big(C_{k1}=j\big)\mathrm{E}_{q(\rho_k)}\big(\log\rho_{kj}\big)$$
$$+\sum_{m=2}^{M}\sum_{j=1}^{J}\sum_{i=1}^{J}\mathrm{I}\big(C_{km-1}=i,C_{km}=j\big)\mathrm{E}_{q(\mathbf{a}_i^k)}\big(\log a_{ij}^k\big).$$

Calculations regarding directed graph: The details of this section are in the main manuscript.

Algorithm 2 CopyMix: a Variational Inference Algorithm

procedure ESTIMATE-POSTERIORS(Y): ▷ Y input is N sequences of length M
Initialization and assigning priors
  repeat
    Update $\delta_k^{(c)}$ using $p_{nk}^{(c-1)}$ and δ_k^0 in equation (B.11)
    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
      $w_k(C_{mj})^{(c-1)}$ (equation (2) to (5))
    Update $\Lambda_{ij}^{k(c)}$ using Λ_{ij}^{k0}, $q(C_{km-1}=i,C_{km}=j)^{(c)}$ in equation (B.18)
    Update $\alpha^{(c)}$ using α^0 and y in equation (B.16)
    Update $\beta^{(c)}$ using β^0, $p_{kn}^{(c)}$, $q(C_{km}=j)^{(c)}$, θ_n, τ_n, and y in equation (B.17)
    Update $p_{nk}^{(c)}$ using $\delta_k^{(c)}$, θ_n, $\tau_n^{(c)}$, $\alpha^{(c)}$, $\beta^{(c)}$, $q(C_{km}=j)^{(c)}$ and y in equation (B.13)
    Update $w_k(C_{mj})^{(c)}$ using $\Lambda_{ij}^{k(c)}$, $p_{kn}^{(c)}$, θ_n, $\tau_n^{(c)}$, $\alpha^{(c)}$, $\beta^{(c)}$, y in equation (3)
    Update $w_k(C_{m-1i}C_{mj})^{(c)}$ using $\Lambda_{ij}^{k(c)}$ and $p_{nk}^{(c)}$, in equation (2)

    $c\leftarrow c+1$
  until convergence of the ELBO
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

Calculating expectations and ELBO for CopyMix

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

$$\boldsymbol{\Psi}(x)=\frac{d}{dx}\log\Gamma(x), \tag{B.19}$$

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.

$$\mathrm{E}_{q(Z_n)}(\mathrm{I}(Z_n=k))=p_{nk}$$

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

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

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

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

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

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

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

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

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

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. (A.1) which is calculated as below. An assumption in the ELBO calculation is that

we ignore all the constants contributing in the ELBO value.

$$
\begin{aligned}
ELBO(q) = \mathrm{E}\Big[\ & \log P(\mathbf{Y}|\mathbf{Z},\mathbf{C},\boldsymbol{\Psi}) + \log P(\mathbf{Z}|\boldsymbol{\Psi}) + \log P(\mathbf{C}|\boldsymbol{\Psi}) \\
& + \log P(\boldsymbol{A}) + \log P(\boldsymbol{\mu}) + \log P(\frac{1}{\sigma^2}) \\
& + \log P(\boldsymbol{\pi})\ \Big] - \mathrm{E}\Big[\ \log q(\mathbf{Z}) + \log q(\mathbf{C}) + \log q(\boldsymbol{A}) \\
& + \log q(\boldsymbol{\mu}) + \log q(\frac{1}{\sigma^2}) + \log q(\boldsymbol{\pi})\ \Big] = \\
\mathrm{E}\Big[& \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{I}(Z_n = k)\mathrm{I}(C_{km} = j) D_{nmj} + \sum_{n=1}^{N}\sum_{k=1}^{K} \mathrm{I}(Z_n = k)\log \pi_k \\
+ & \sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{I}(C_{k1} = j)\log \rho_{kj} + \sum_{k=1}^{K}\sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J} \mathrm{I}(C_{km-1} = i, C_{km} = j)\log a_{ij}^{k} \\
+ & \sum_{k=1}^{K}\sum_{i=1}^{J}\sum_{j=1}^{J} (\Lambda_j - 1)\log a_{ij}^{k} + \sum_{n=1}^{N}(\frac{-1}{2})\Big(\log(2\pi\tau^2) + \frac{\mu_n^2 - 2\theta\mu_n + \theta^2}{\tau^2}\Big) \\
+ & (\frac{-\beta}{\sigma^2}) + (\alpha - 1)\log\frac{1}{\sigma^2} + \log P(\boldsymbol{\pi})\ \Big] \\
- & \ \mathrm{E}\Big[\sum_{n=1}^{N}\log q(Z_n) + \sum_{k=1}^{K}\log q(\mathbf{C}_k) + \sum_{k=1}^{K}\sum_{i=1}^{J}\log q(\mathbf{a}_i^k) \\
+ & \sum_{n=1}^{N}\log q(\mu_n) + \log q(\frac{1}{\sigma^2}) + \log q(\boldsymbol{\pi})\ \Big]
\end{aligned}
\tag{B.20}
$$

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:

$$
\begin{aligned}
& \mathrm{E}_{q(\mathbf{C}_k)}[\log q(\mathbf{C}_k)] \\
& = \mathrm{E}_{q(\mathbf{C}_k)}[\log \prod_{m=2}^{M}\prod_{i=1}^{J}\prod_{j=1}^{J} q(C_{km} = j|C_{km-1} = i)^{\mathrm{I}(C_{km-1}=i, C_{km}=j)}] = \\
& \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J} \mathrm{E}_{q(C_{km-1}, C_{km})}\Big[\mathrm{I}(C_{km-1} = i, C_{km} = j)\Big]\log q(C_{km} = j|C_{km-1} = i) \\
& \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J} \mathrm{E}_{q(C_{km-1}, C_{km})}\Big[\mathrm{I}(C_{km-1} = i, C_{km} = j)\Big] \\
& \times \log\frac{\mathrm{E}_{q(C_{km-1}, C_{km})}\Big[\mathrm{I}(C_{km-1} = i, C_{km} = j)\Big]}{\mathrm{E}_{q(C_{km-1})}\Big[\mathrm{I}(C_{km-1} = i)\Big]}
\end{aligned}
\tag{B.21}
$$

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

$$
\begin{aligned}
\mathrm{E}_{(q(Z_n), q(\boldsymbol{\pi}))}\Big[\log p(Z_n|\Psi)\Big] &= \mathrm{E}_{(q(Z_n), q(\boldsymbol{\pi}))}\Big[\sum_{k=1}^{K}\mathrm{I}(Z_n = k)\log \pi_k\Big] \\
&= \sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n = k)\Big]\mathrm{E}_{q(\boldsymbol{\pi})}\Big[\log \pi_k\Big]
\end{aligned}
\tag{B.22}
$$

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

$$
\begin{aligned}
\mathrm{E}_{q(Z_n)}\Big[\log q(Z_n)\Big] &= \mathrm{E}_{q(Z_n)}\Big[\sum_{k=1}^{K}\mathrm{I}(Z_n = k)\log q(Z_n = k)\Big] \\
&= \sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n = k)\Big]\log q(Z_n = k)
\end{aligned}
\tag{B.23}
$$

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

$$
\mathrm{E}_{q(\mu_n)}[\log p(\mu_n)] = (\frac{-1}{2})\Big(\log(2\pi\tau^2) + \frac{\mathrm{E}_{q(\mu_n)}\Big[\mu_n^2\Big] - 2\theta\mathrm{E}_{q(\mu_n)}\Big[\mu_n\Big] + \theta^2}{\tau^2}\Big)
\tag{B.24}
$$

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

$$
\begin{aligned}
\mathrm{E}_{q(\mu_n)}[\log q(\mu_n)] &= (\frac{-1}{2})\Big(\log(2\pi\tau_n^{*2}) + \frac{\mathrm{E}_{q(\mu_n)}\Big[\mu_n^2\Big] - 2\theta_n^{*}\mathrm{E}_{q(\mu_n)}\Big[\mu_n\Big] + \theta_n^{*2}}{\tau_n^{*2}}\Big) \\
&= (\frac{-1}{2})\Big(\log(2\pi\tau_n^{*2}) + 1\Big)
\end{aligned}
\tag{B.25}
$$

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

$$
\begin{aligned}
& \mathrm{E}_{q(\boldsymbol{\pi})}(\log P(\boldsymbol{\pi})) \\
& = \sum_{k=1}^{K}(\delta_k - 1)\mathrm{E}_{q(\boldsymbol{\pi})}(\log \boldsymbol{\pi}) - \log B(\boldsymbol{\delta}) \\
& = \sum_{k=1}^{K}(\delta_k - 1)\Big[\boldsymbol{\Psi}(\delta_k) - \boldsymbol{\Psi}\big(\sum_{k=1}^{K}\delta_k\big)\Big] - \log B(\boldsymbol{\delta}) \\
& = \Big(\sum_{k=1}^{K}\delta_k - K\Big)\boldsymbol{\Psi}\Big(\sum_{k=1}^{K}\delta_k\Big) - \sum_{k=1}^{K}\boldsymbol{\Psi}(\delta_k)(\delta_k - 1) - \log B(\boldsymbol{\delta})
\end{aligned}
\tag{B.26}
$$

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

$$
\mathrm{E}_{q(\boldsymbol{\pi})}(\log q(\boldsymbol{\pi})) = \Big(\sum_{k=1}^{K}\delta_k - K\Big)\boldsymbol{\Psi}\Big(\sum_{k=1}^{K}\delta_k\Big) - \sum_{k=1}^{K}\boldsymbol{\Psi}(\delta_k)(\delta_k - 1) - \log B(\boldsymbol{\delta})
\tag{B.27}
$$

$$
ELBO(q) = \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n = k)\Big]\mathrm{E}_{q(C_{km})}\Big[\mathrm{I}(C_{km} = j)\Big]
\tag{B.28}
$$

$$
\times\ \mathrm{E}_{q(\mu_n)q(\frac{1}{\sigma^2})}\Big[D_{nmj}\Big]
$$

$$
+ \sum_{n=1}^{N}\sum_{k=1}^{K}\mathrm{E}_{q(Z_n)}\Big[\mathrm{I}(Z_n = k)\Big]\mathrm{E}_{q(\boldsymbol{\pi})}\Big[\log \pi_k\Big]
\tag{B.29}
$$

$$
+ \cancel{\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(C_{k1})}\Big[\mathrm{I}(C_{k1} = j)\Big]\log \rho_{kj}}
$$

$$
+ \sum_{k=1}^{K}\sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J}\mathrm{E}_{q(C_{km-1}, C_{km})}\Big[\mathrm{I}(C_{km-1} = i, C_{km} = j)\Big]\mathrm{E}_{q(\mathbf{a}_i^k)}\Big[\log a_{ij}^{k}\Big]
\tag{B.30}
$$

$$
+ \sum_{k=1}^{K}\sum_{i=1}^{J}\sum_{j=1}^{J}(\Lambda_j - 1)\mathrm{E}_{q(\mathbf{a}_i^k)}\Big[\log a_{ij}^{k}\Big]
$$

$$
+ \sum_{n=1}^{N}(\frac{-1}{2})\Big(\log(2\pi\tau^2) + \frac{\mathrm{E}_{q(\mu_n)}\Big[\mu_n^2\Big] - 2\theta\mathrm{E}_{q(\mu_n)}\Big[\mu_n\Big] + \theta^2}{\tau^2}\Big)
\tag{B.31}
$$

$$
+ (-\beta)\mathrm{E}_{q(\frac{1}{\sigma^2})}\Big[\frac{1}{\sigma^2}\Big] + (\alpha - 1)\mathrm{E}_{q(\frac{1}{\sigma^2})}\Big[\log\frac{1}{\sigma^2}\Big] + \mathrm{E}_{q(\boldsymbol{\pi})}\Big[\log P(\boldsymbol{\pi})\Big]
\tag{B.32}
$$

$$
- \sum_{n=1}^{N}\Big[\sum_{k=1}^{K}p_{nk}\log p_{nk}\Big] - \sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{j=1}^{J}\mathrm{E}_{q(C_{km})}\Big[\mathrm{I}(C_{km} = j)\Big]\mathrm{E}_{q(\mu_n)q(\frac{1}{\sigma^2})}\Big[D_{nmj}\Big]
\tag{B.33}
$$

$$
- \cancel{\sum_{k=1}^{K}\sum_{j=1}^{J}\mathrm{E}_{q(C_{k1})}\Big[\mathrm{I}(C_{k1} = j)\Big]\log \rho_{kj}}
\tag{B.34}
$$

$$
- \sum_{k=1}^{K}\sum_{m=2}^{M}\sum_{j=1}^{J}\sum_{i=1}^{J}\mathrm{E}_{q(C_{km-1}, C_{km})}\Big[\mathrm{I}(C_{km-1} = i, C_{km} = j)\Big]
$$

$$
\times \log\frac{\mathrm{E}_{q(C_{km-1}, C_{km})}\Big[\mathrm{I}(C_{km-1} = i, C_{km} = j)\Big]}{\mathrm{E}_{q(C_{km-1})}\Big[\mathrm{I}(C_{km-1} = i)\Big]}
\tag{B.35}
$$

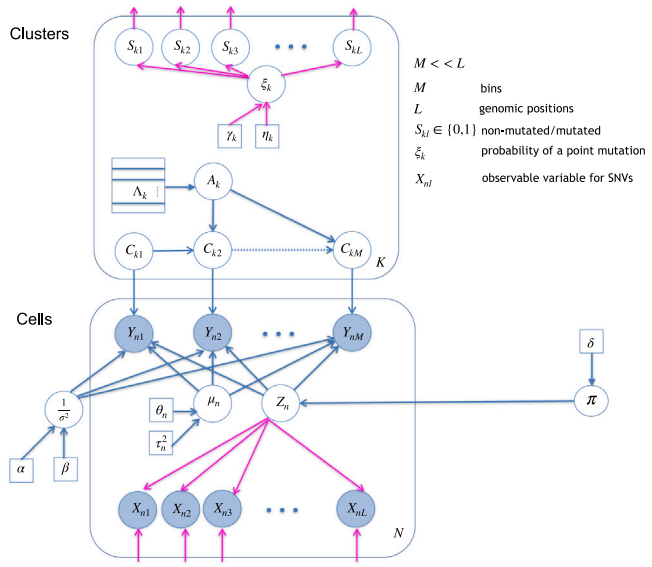


Fig. 6. The graphical model with SNV process included.

$$-\sum_{k=1}^{K}\sum_{i=1}^{J}\Big[\sum_{j=1}^{J}\mathrm{E}_{q(\mathbf{a}_i^k)}\Big[\log a_{ij}^k\Big]\Big\{\Lambda_j - 1\Big\}\Big] \tag{B.36}$$

$$-\sum_{n=1}^{N}(\frac{-1}{2})\Big(\log(2\pi\tau_n^2)+1\Big) \tag{B.37}$$

$$-\Big(\alpha - \ln\beta + \ln\Gamma(\alpha) + (1-\alpha)\boldsymbol{\Psi}(\alpha)\Big) \tag{B.38}$$

$$-\mathrm{E}_{q(\boldsymbol{\pi})}\Big[\log q(\boldsymbol{\pi})\Big] \tag{B.39}$$

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:

$$\alpha - \ln\beta + \alpha\ln(\alpha) - \alpha - \ln(\alpha) + (\ln(\alpha) - \frac{1}{2\alpha}) - \alpha\ln(\alpha) + \frac{1}{2}$$
$$= \frac{1}{2}(1 - \frac{1}{\alpha}) - \ln\beta$$

Appendix C

SNV-CopyMix

This section incorporates SNV data into the model (see Fig. 6). We augment CopyMix graphical model by a copy-number-independent SNV process; we refer to this version of CopyMix as SNV-CopyMix. In Fig. 6, the new components are colored in magenta. X_{nl} denotes the observable variable, corresponding to a nucleotide in the genome, that we assume to be dependent on cluster-specific latent point mutation S_{kl}. Naturally, each S_{kl} is distributed as Bernoulli, where mutation corresponds to the value of zero, and non-mutation corresponds to the value of one. The prior distribution of S_{kl} is ξ, a Beta distribution. Note that the SNV data contains L sites such that $L \gg M$. In the following subsection, the calculation of the probability of X_{nl} is described; for more detail, see the work by Koptagel et al. (2018). The VI update equations are calculated analogously to those in CopyMix; see the derivations in the rest of this Appendix A.

The likelihood model for SNVs

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

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

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

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

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

The details of SNV-CopyMix

Let X_{nl} be an observation that is dependent on genotype. In this case we deal with two genotypes, i.e., 0 for non mutation and 1 for mutation. We assume to have access to these probabilities from another graphical model (Koptagel et al., 2018) which deals with SNVs. In the following section, we provide details on SNV probabilities.

Dataset Details There are 891 cells, from three cell lines. There is no bulk data from healthy tissue.

1. Create a txt file which includes all bam filenames.
2. Extract SNV sites information from the CSV file provided by the authors
3. Extract reads from BAM files.
4. Process reads of each cell
   - Remove first/last tokens from nucleotides, discard N.
   - Convert Phred quality scores to error probabilities using the formula $\epsilon = 10^{-0.1\times q}$
5. Compute loglikelihood tensor for various scenarios
   - heterozygous SNV
   - homozygous SNV
   - samtoools mpileup -B

For $c \in [C]$ cells for $s \in S$ sites, the goal is to report a tensor of shape $L = (C \times S \times 2)$ where L_{csm} is the log-likelihood of observing the reads of cell c at site s if the cell is mutated $m = 1$ or not $m = 0$.

More formally, if the cell has $R_{cs}^{1:|R|}$ reads and corresponding error probabilities $E_{cs}^{1:|R|}$, we have:

$$\begin{aligned} L_{csm} &= \log p(R_{cs}^{1:|R|} \mid E_{cs}^{1:|R|}, M = m) \\ &= \sum_{r=1}^{|R|} \log p(R_{cs}^{r} \mid E_{cs}^{r}, M = m) \end{aligned} \tag{C.40}$$

Moreover, due to our pipeline, we will know what the mutated and healthy true genotype G is. So, essentially we will compute

$$p(R_{cs}^{r} \mid E_{cs}^{r}, G_s) \tag{C.41}$$

where G_s is the healthy reference if $m = 0$ and SNV if $m = 1$.

A Toy example: Assume the reference genotype at site s is $G_s = \{A, A\}$ and mutated genotype is $G_s = \{A, C\}$. Assume the cell c has three reads $R_{cs}^{1:3} = \{A, A, G\}$ and the corresponding error probabilities $E_{cs}^{1:3} = \{0.9, 0.8, 0.5\}$.

$$\begin{aligned} L_{cs0} &= \log p(R_{cs}^{1:3} \mid E_{cs}^{1:3}, G_s = \{A, A\}) \\ &= \sum_{r=1}^{3} \log p(R_{cs}^{r} \mid E_{cs}^{r}, G_s = \{A, A\}) \end{aligned} \tag{C.42}$$

VI for SNV-CopyMix

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

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

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

$$q(\mathbf{Z}, \mathbf{C}, \mathbf{S}, \boldsymbol{\Psi}) \tag{C.43}$$

with smallest Kullback–Leibler divergence to the posterior distribution

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

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

$$\text{ELBO}(q) = \text{E}\big[\log P(\mathbf{Y}, \mathbf{X}, \mathbf{Z}, \mathbf{C}, \mathbf{S}, \boldsymbol{\Psi})\big] - \text{E}\big[\log q(\mathbf{Z}, \mathbf{C}, \mathbf{S}, \boldsymbol{\Psi})\big]. \tag{C.44}$$

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

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

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

Step 1. VD factorization

We assume the following factorization of the VD:

$$\begin{aligned} &q(\mathbf{Z}, \mathbf{C}, \mathbf{S}, \boldsymbol{\Psi}) = \\ &\Big[\prod_{n=1}^{N} q(Z_n)\Big]\Big[\prod_{k=1}^{K} q(\mathbf{C}_k)\Big]\Big[\prod_{k=1}^{K}\prod_{i=1}^{J} q(\mathbf{A}_i^k)\Big]\Big[\prod_{n=1}^{N} q(\mu_n)\Big] \\ &\Big[\prod_{k=1}^{K} q(\boldsymbol{S}_k)\Big]\Big[\prod_{k=1}^{K} q(\xi_k)\Big] q(\frac{1}{\sigma^2}) q(\boldsymbol{\pi}). \end{aligned} \tag{C.45}$$

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

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

$$\begin{aligned} &\log P(\mathbf{Y}, \mathbf{X}, \mathbf{Z}, \mathbf{C}, \mathbf{S}, \boldsymbol{\Psi}) = \\ &\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) + \log P(\mathbf{X}|\mathbf{Z}, \mathbf{S}, \boldsymbol{\Psi}) + \log P(\mathbf{C}|\boldsymbol{\Psi}) \\ &+ \log P(\mathbf{Z}|\boldsymbol{\Psi}) + \log P(\mathbf{S}|\boldsymbol{\Psi}) + \log P(\boldsymbol{\Psi}), \end{aligned} \tag{C.46}$$

where

$$\log P(\mathbf{X}|\mathbf{Z}, \mathbf{S}, \boldsymbol{\Psi}) = \sum_{n=1}^{N}\sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{s=0}^{1} \text{I}(Z_n = k)\text{I}(S_{kl} = s) log P(X_{nl}|g = s) \tag{C.47}$$

We assume we have these probabilities given the genotype g being either mutation or non mutation.

$$\begin{aligned} \log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) &= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \text{I}(Z_n = k)\text{I}(C_{km} = j) \log F_{\mu_{nj},\sigma^2}(r_{nm}) \\ &= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \text{I}(Z_n = k)\text{I}(C_{km} = j)(-\frac{1}{2}) \\ &\quad \times \Big[\log(2\pi\sigma^2) + \frac{(r_{nm} - j\mu_n)^2}{\sigma^2}\Big], \end{aligned} \tag{C.48}$$

$$\begin{aligned} \log P(\mathbf{C}|\boldsymbol{\Psi}) = \sum_{k=1}^{K}\Bigg[&\sum_{j=1}^{J} \text{I}(C_{k1} = j) \log \rho_{kj} \\ &+ \sum_{m=2}^{M}\sum_{i=1}^{J}\sum_{j=1}^{J} \text{I}(C_{km-1} = i, C_{km} = j) \log A_{ij}^k\Bigg], \end{aligned} \tag{C.49}$$

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

$$\log P(\mathbf{Z}|\boldsymbol{\Psi}) = \sum_{n=1}^{N}\sum_{k=1}^{K} \text{I}(Z_n = k) \log \pi_k, \tag{C.50}$$

and

$$\log P(\mathbf{S}|\boldsymbol{\Psi}) = \sum_{l=1}^{L}\sum_{k=1}^{K} \text{I}(S_{kl} = 0) \log \xi_k + \text{I}(S_{kl} = 1) \log(1 - \xi_k), \tag{C.51}$$

Moreover,

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

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

$$\log P(\boldsymbol{\mu}) = \sum_{n=1}^{N}\Big[(-\frac{1}{2})\Big[\log(2\pi\tau^2) + \frac{(\mu_n - \theta)^2}{\tau^2}\Big]\Big], \tag{C.52}$$

$$\log P(\frac{1}{\sigma^2}) = \Big[-\frac{\beta}{\sigma^2} + (\alpha - 1)\log(\frac{1}{\sigma^2})\Big] + C, \tag{C.53}$$

$$\log P(\mathbf{A}) = \sum_{k=1}^{K}\sum_{i=1}^{J}\Big[\sum_{j=1}^{J}(\Lambda_j - 1)\log A_{ij}^k - \log \mathcal{B}(\boldsymbol{\Lambda})\Big], \tag{C.54}$$

$$\log P(\boldsymbol{\xi}) = \sum_{k=1}^{K}(\eta_k - 1)\log \xi_k + (\gamma_k - 1)\log(1 - \xi_k) - C(\eta_k, \gamma_k), \tag{C.55}$$

and

$$\log P(\boldsymbol{\pi}) = \sum_{k=1}^{K}(\delta_k - 1)\log \pi_k - \log \mathcal{B}(\boldsymbol{\delta}). \tag{C.56}$$

Step 3. VD computation by coordinate ascent We now derive a coordinate ascent algorithm for the VD. That is, we derive an update equation for each term in the factorization (in step 1) by calculating the expectation of $\log P(\mathbf{Y}, \mathbf{X}, \mathbf{Z}, \mathbf{C}, \mathbf{S}, \boldsymbol{\Psi})$ over the VD of all random variables except the one of interest. Below, the update equation is derived for each random variable. For convenience, we use $\overset{+}{\approx}$ to denote equality up to a constant additive factor.

Note that $\mathbf{X}$ can have missing values at certain sites. Therefore, including this into the model, results in splitting $\mathbf{X}$ into $\mathbf{X_{miss}}$ and $\mathbf{X_{obs}}$. Moreover, we would need a Bernoulli variable, ω, which shows which of the two group is the case in the joint distribution. The joint distribution then takes the following form.

$$P(\mathbf{Y}, \mathbf{X_{obs}}, \mathbf{X_{miss}}, \omega, \mathbf{S}, \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) =$$
$$P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) P(\mathbf{X_{obs}}|\mathbf{S}, \mathbf{Z}, \boldsymbol{\Psi}) P(\mathbf{X_{miss}}|\mathbf{S}, \mathbf{Z}, \boldsymbol{\Psi}) P(\mathbf{C}|\boldsymbol{\Psi}) P(\mathbf{Z}|\boldsymbol{\Psi}) P(\mathbf{S}|\boldsymbol{\Psi}) P(\boldsymbol{\Psi})$$
$$P(\omega|\mathbf{Y}, \mathbf{X_{obs}}, \mathbf{X_{miss}}, \mathbf{S}, \mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) \xrightarrow{MCAR}$$

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

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

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

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

$$\delta_k = \delta_k^0 + \sum_{n=1}^{N} \mathrm{E}_{q(Z_n)}\big(\mathrm{I}(Z_n = k)\big). \tag{C.57}$$

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

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

Note that, each of the logarithms above can be written as the sum of two terms, one that depends on Z_n and one that does not; since we want to form a function of Z_n, we discard all other terms w.r.t. $Z_c : c \in [N] \setminus n$.

$$\begin{aligned}
\log P(\mathbf{Y}|\mathbf{Z}, \mathbf{C}, \boldsymbol{\Psi}) \implies & \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{I}(Z_n = k)\mathrm{I}(C_{km} = j)(-\frac{1}{2}) \\
& \times \Big[\log(2\pi\sigma^2) + \frac{(r_{nm} - j\mu_n)^2}{\sigma^2}\Big] \\
\log P(\mathbf{X}|\mathbf{Z}, \mathbf{S}, \boldsymbol{\Psi}) \implies & \sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{s=0}^{1} \mathrm{I}(Z_n = k)\mathrm{I}(S_{kl} = s) \log P(X_{nl}|g = s)
\end{aligned} \tag{C.58}$$

Taking the expectation, we have:

$$\mathrm{E}_{-Z_n}(\log P(\mathbf{X}|\mathbf{Z}, \mathbf{S}, \boldsymbol{\Psi})) = \sum_{l=1}^{L}\sum_{k=1}^{K} \mathrm{I}(Z_n = k) \times \Big(\sum_{s=0}^{1} \mathrm{E}\Big\{\mathrm{I}(S_{kl} = s)\Big\} \log P(X_{nl}|g = s) \; \Big\}\Big) \tag{C.59}$$

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

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

$$\begin{aligned}
Q_Y = \sum_{k=1}^{K} \mathrm{I}(Z_n = k)\Big\{ & \mathrm{E}_{q(\boldsymbol{\pi})}(\log \pi_k) \\
& + \sum_{m=1}^{M}\sum_{j=1}^{J} \mathrm{E}_{q(\mathbf{C}_k)}\big(\mathrm{I}(C_{km} = j)\big)\mathrm{E}_{q(\frac{1}{\sigma^2})q(\mu_n)}[D_{nmj}]\Big\}
\end{aligned} \tag{C.60}$$

Where:

$$D_{nmj} = -\frac{1}{2}\Big[\log(2\pi\sigma^2) + \frac{(r_{nm} - j\mu_n)^2}{\sigma^2}\Big]. \tag{C.61}$$

$$\log q(Z_n) \overset{+}{\approx} Q_Y + \sum_{l=1}^{L}\sum_{k=1}^{K} \mathrm{I}(Z_n = k)\Big(\sum_{s=0}^{1} \mathrm{E}\Big\{\mathrm{I}(S_{kl} = s)\Big\} \log P(X_{nl}|g = s)\Big) \tag{C.62}$$

Reformulating $\log q(Z_n)$, we achieve the below. Note that when processing the input data, for the missing values we merely calculate Q_Y.

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

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

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

and

$$p_{nk} = \frac{\exp(\tilde{p}_{nk})}{\sum_{k'=1}^{K} \exp(\tilde{p}_{nk'})}. \tag{C.63}$$

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

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

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

$$\begin{aligned}
&\log q(\xi_k) \overset{+}{\approx} (\eta^0 - 1)\log \xi_k + (\gamma^0 - 1)\log(1 - \xi_k) \\
&\quad + \sum_{l=1}^{L} \mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl} = 0)) \log \xi_k + \\
&\mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl} = 1)) \log(1 - \xi_k) \\
&= \log \xi_k \big\{\eta^0 - 1 + \sum_{l=1}^{L} \mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl} = 0))\big\} + \log(1 - \xi_k) \; \{ \; \gamma^0 - 1 \\
&\quad + \sum_{l=1}^{L} \mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl} = 1)) \; \}
\end{aligned} \tag{C.64}$$

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

$$\eta_k = \eta^0 - 1 + \sum_{l=1}^{L} \mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl} = 0)) \tag{C.65}$$

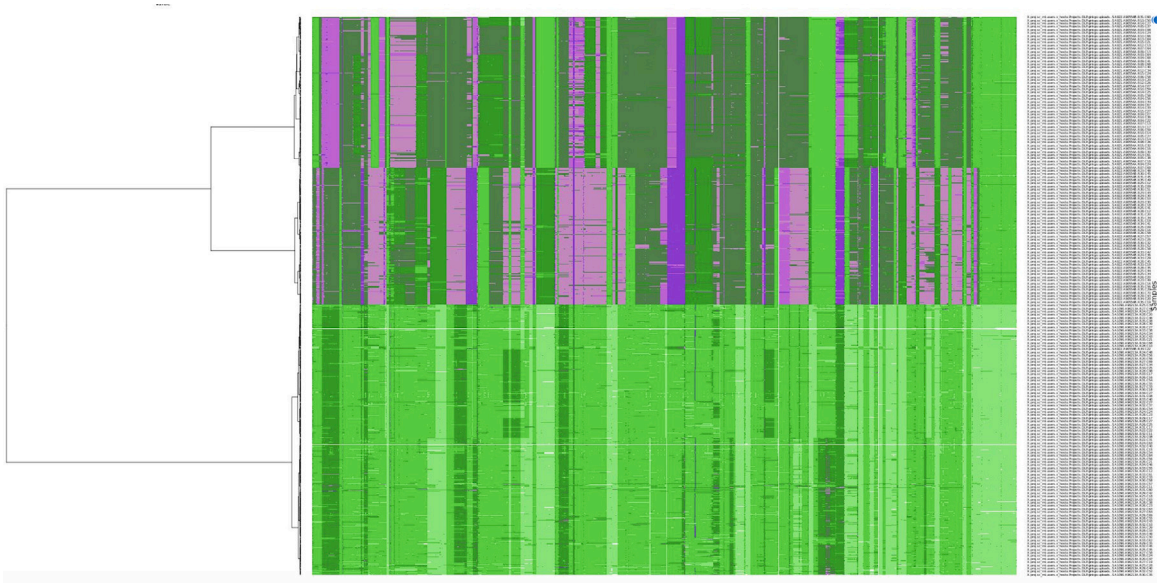


Fig. 7. The heat map and the dendrogram show three major clusters detected by Ginkgo. The copy number is color-coded by a scale from green (small) to magenta (large).

$$\gamma_k = \gamma^0 - 1 + \sum_{l=1}^{L} \mathrm{E}_{q(S_k)}(\mathrm{I}(S_{kl} = 1)) \tag{C.66}$$

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

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

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

$$\begin{aligned}
\log q(S_{kl}) &\overset{+}{\approx} \mathrm{I}(S_{kl} = 0)\mathrm{E}_{q(\xi_k)}(\log \xi_k) + \mathrm{I}(S_{kl} = 1)\mathrm{E}_{q(\xi_k)}(\log(1 - \xi_k)) + \\
&\sum_{n=1}^{N} \sum_{s=0}^{1} \mathrm{E}_{q(Z_n)}(\mathrm{I}(Z_n = k))\mathrm{I}(S_{kl} = s) \log P(X_{nl}|g = s) \\
&\overset{+}{\approx} \mathcal{A} + \sum_{n=1}^{N} \mathcal{B}
\end{aligned} \tag{C.67}$$

where $\mathcal{A} = \mathrm{I}(S_{kl} = 0)\mathrm{E}_{q(\xi_k)}(\log \xi_k) + \mathrm{I}(S_{kl} = 1)\mathrm{E}_{q(\xi_k)}(\log(1 - \xi_k))$, and $\mathcal{B} = \sum_{s=0}^{1} \mathrm{E}_{q(Z_n)}(\mathrm{I}(Z_n = k))\mathrm{I}(S_{kl} = s) \log P(X_{nl}|g = s)$.

Therefore, $q(S_{kl})$ is distributed by Bernoulli up to an additive constant. The Bernoulli parameter, corresponding to $S_{kl} = 1$ is as the following.

$$\hat{\Sigma_{kl}} = \mathrm{E}_{q(\xi_k)}(\log(1 - \xi_k)) + \sum_{n=1}^{N} \mathrm{E}_{q(Z_n)}(\mathrm{I}(Z_n = k)) \log P(X_{nl}|g = 1) \tag{C.68}$$

$$\Sigma_{kl} = \frac{\exp(\hat{\Sigma_{kl}})}{\sum_{k=1}^{K} \exp(\hat{\Sigma_{kl}})} \tag{C.69}$$

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

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

$$\mathrm{E}_{q(\xi_k)}(\log \xi_k) = \boldsymbol{\Psi}(\eta_k) - \boldsymbol{\Psi}(\eta_k + \gamma_k) \tag{C.70}$$

$$\mathrm{E}_{q(\xi_k)}(\log(1 - \xi_k)) = \boldsymbol{\Psi}(\gamma_k) - \boldsymbol{\Psi}(\gamma_k + \eta) \tag{C.71}$$

Appendix D

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

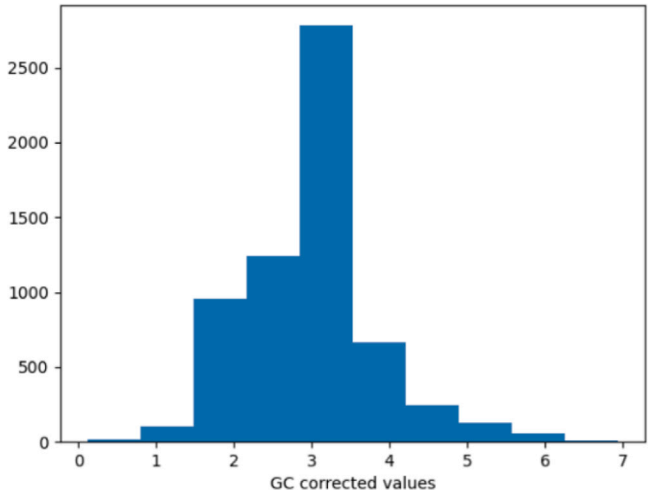


Fig. 8. DLP data histogram plot.

The generated configurations (CONF 1 to 18) are shown in the main paper, see Fig. 3. We modulate the read ratios by introducing subtle noise. Regarding the number of clusters, each of CONF 1 to CONF 5 contains two clusters, CONF 6 to CONF 10 three clusters, CONF 11 to CONF 13 four clusters, and CONF 14 to CONF 18 five clusters. The copy number transition patterns are formed using the following copy number state patterns, which are inspired by the CNV, along with single-cells that are obtained using the DLP technology (Laks et al., 2019).

- **single state**: a copy number is inclined to remain at one certain state (CONF 1, CONF 2, CONF 3, CONF 4, CONF 5, CONF 12, and CONF 13 include this pattern);
- **inertial state**: a copy number is inclined to remain unchanged (The blue cluster in CONF 6 to CONF 10 includes this pattern);

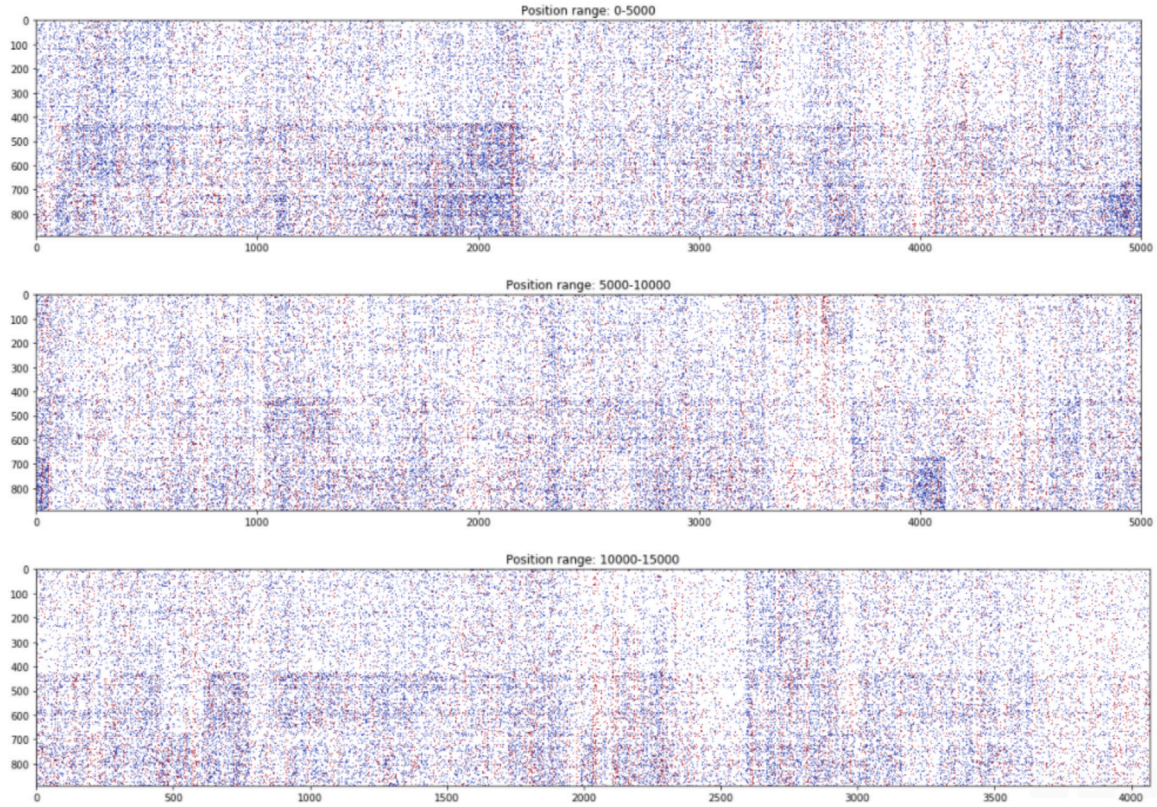


Fig. 9. SNV counts are illustrated as colors in the heatmap for 891 cells (Y axis) across the genomic positions (X axis).

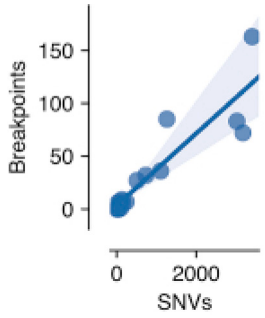


Fig. 10. Correlation between CNVs and SNVs on the DLP data.

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

For each simulation, the following steps are performed.

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

Appendix E

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

Ginkgo's hierarchical clustering, after deleting 80% of the cells, detects three main clusters; firstly, the question is how the clustering would be if they did not exclude so many cells. Secondly, we cannot state that hierarchical clustering detects more than those three clusters. That is because the branch lengths of the smaller clusters, in the hierarchical clustering, are negligible in size compared to the branch length of the main three clusters. To include these branches and, consequently, the smaller clusters, one must choose a very shallow threshold when cutting the dendrogram produced by the hierarchical clustering. If we, based on shared events in dark green in Fig. 7, assign a threshold such that we include the two subgroups depicted in the bottom and green part of Fig. 7, then Ginkgo detects four clusters. Regarding the four clusters, CopyMix outperforms Ginkgo with V-measure 67% compared to V-measure 55%. The remaining Ginkgo subgroups containing

magenta-colored copy numbers require a lower threshold than those for the four clusters. Finally, it is essential to mention that Ginkgo's web application, one advantage of using Ginkgo (Mallory et al., 2020), could not handle the large DLP data. To run Ginkgo, we developed a bash tool that assembled the back-end parts of the Ginkgo into a new script.

Appendix F

This section discusses why Gaussian is chosen as a distribution for the read ratios. Following the principle of maximum entropy, we choose Gaussian because it has maximum entropy for a specified mean and variance (Jaynes, 1957). Also, the histogram plot of DLP data (Fig. 8) confirms a skew-normal distribution. Finally, it is common that the data is assumed to follow Gaussian distribution; see HMMcopy and another HMM-based approach (Shah et al., 2006; Guha et al., 2008).

Appendix G

In this section, we illustrate the SNV data, confirming too shallow data signals for further cluster detection. As shown in Fig. 9, one can observe that the cells, which are on the Y axis, can be clustered into three main groups. Note that the three plots show in total 15000 sites where SNVs are detected. Performing K-means and hierarchical clustering resulted in three clusters; hence, it is reasonable that this data has been too shallow to reveal more clusters.

Finally, Fig. 10 that is taken from the analysis of the DLP data (Laks et al., 2019) shows a fairly equal contribution of CNVs (breakpoints) and SNVs supported by the high correlation between them.

References

Alison, G., Edward Su, F., 2002. On choosing and bounding probability metrics. Int. Stat. Rev..

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

Bishop, C., 2006. Pattern recognition and machine learning. Inf. Sci. Stat..

Blei, D., Kucukelbir, A., Mcauliffe, J., 2017. Variational inference: A review for statisticians. J. Amer. Statist. Assoc. 112 (518), 859–877.

Blocker, A.W., Meng, X.-L., 2013. The potential and perils of preprocessing: Building new foundations. Bernoulli 19 (4), 1176–1211.

Calinski, T., Harabasz, J., 1974. A dendrite method for cluster analysis. Commun. Stat. 1–27.

de Souza, C.P.E., 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 Comput. Biol..

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 (7539), 422.

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

Freeman, J., et al., 2006. Copy number variation: New insights in genome diversity. Genome Res..

Gao, R., Bai, S., Henderson, Y.C., Navin, N.E., 2021. Delineating copy number and clonal substructure in human tumors from single-cell transcriptomes. Nature Biotechnol..

Garvin, T., Aboukhalil, R., Kendall, s., 2015. Interactive analysis and assessment of single-cell copy-number variations. Nature Methods 1058–11060.

Gawad, C., Koh, W., Quake, S.R., 2016. Single-cell genome sequencing: current state of the science. Nature Rev. Genet. 17 (3), 175.

Guha, S., Li, Y., Neuberg, D., 2008. Bayesian hidden Markov modeling of array CGH data. J. Amer. Statist. Assoc..

Jaynes, E.T., 1957. Information theory and statistical mechanics. Phys. Rev..

Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K., 1999. An introduction to variational methods for graphical models. Mach. Learn. 37 (2), 183–233.

Kapourani, C.-A., Sanguinetti, G., 2019. Melissa: Bayesian clustering and imputation of single-cell methylomes. Genome Biology 20 (1), 61.

Koptagel, H., Jun, S., Lagergren, J., 2018. Scuphr: A probabilistic framework for cell lineage tree reconstruction. https://www.biorxiv.org/content/10.1101/357442v1.

Laks, E., McPherson, A., et al., 2019. Clonal decomposition and DNA replication states defined by scaled single-cell genome sequencing. Cell.

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

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

Lobo, I., 2008. Copy number variation and genetic disease. Nature Educ..

Malekpour, S., Pezeshk, H., Sadeghi, M., 2018. Mseq-CNV: accurate detection of copy number variation from sequencing of multiple samples. Nat. Sci. Rep..

Mallory, X., Edrisi, M., Navin, N., Nakhleh, L., 2020. Methods for copy number aberration detection from single-cell DNA-sequencing data. Genome Biol..

Markowska, M., Cakala, T., Miasojedow, B., et al., 2022. CONET: copy number event tree model of evolutionary tumor history for single-cell data. Genome Biol..

McGrory, C.A., Titterington, D.M., 2009. Variational Bayesian analysis for hidden Markov models. Aust. N. Z. J. Stat..

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 (7341), 90.

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

Rosenberg, A., 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 (7), 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., Murphy, K.P., 2006. Integrating copy number polymorphisms into array CGH analysis using a robust HMM. Bioinformatics 22 (14), e431–e439.

Shapiro, E., Biezuner, T., Linnarsson, S., 2013. Single-cell sequencing-based technologies will revolutionize whole-organism science. Nature Rev. Genet. 14 (9), 618.

Sirazdinov, S., Mamatov, M., 1962. On mean convergence for densities. Teor. Veroyatn. Primen. 7, 433–437.

Smyth, P., 1997. Clustering sequences with hidden markov models. Adv. Neural Inf. Process. Syst..

Tseng, P., 2001. Convergence of a block coordinate descent method for nondifferentiable minimization. J. Optim. Theory Appl. 109.

Vitak, S., Torkenczy, K., Rosenkrantz, J., Fields, A., Christiansen, L., Adey, A., 2017. Sequencing thousands of single-cell genomes with combinatorial indexing. Nature Methods 14 (3).

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

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

Zaccaria, S., Raphael, B., 2021. Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL. Nature Biotechnol..

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

Zahn, H., Steif, A., Laks, E., Eirew, P., VanInsberghe, M., Shah, S.P., Aparicio, S., Hansen, C.L., 2017. Scalable whole-genome single-cell library preparation without preamplification. Nature Methods 14 (2), 167.

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

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