

RESEARCH

Open Access



Inferring single-cell copy number profiles through cross-cell segmentation of read counts

Furui Liu¹, Fangyuan Shi^{1,2} and Zhenhua Yu^{1,2*}

Abstract

Background Copy number alteration (CNA) is one of the major genomic variations that frequently occur in cancers, and accurate inference of CNAs is essential for unmasking intra-tumor heterogeneity (ITH) and tumor evolutionary history. Single-cell DNA sequencing (scDNA-seq) makes it convenient to profile CNAs at single-cell resolution, and thus aids in better characterization of ITH. Despite that several computational methods have been proposed to decipher single-cell CNAs, their performance is limited in either breakpoint detection or copy number estimation due to the high dimensionality and noisy nature of read counts data.

Results By treating breakpoint detection as a process to segment high dimensional read count sequence, we develop a novel method called DeepCNA for cross-cell segmentation of read count sequence and per-cell inference of CNAs. To cope with the difficulty of segmentation, an autoencoder (AE) network is employed in DeepCNA to project the original data into a low-dimensional space, where the breakpoints can be efficiently detected along each latent dimension and further merged to obtain the final breakpoints. Unlike the existing methods that manually calculate certain statistics of read counts to find breakpoints, the AE model makes it convenient to automatically learn the representations. Based on the inferred breakpoints, we employ a mixture model to predict copy numbers of segments for each cell, and leverage expectation–maximization algorithm to efficiently estimate cell ploidy by exploring the most abundant copy number state. Benchmarking results on simulated and real data demonstrate our method is able to accurately infer breakpoints as well as absolute copy numbers and surpasses the existing methods under different test conditions. DeepCNA can be accessed at: <https://github.com/zhyu-lab/deepcna>.

Conclusions Profiling single-cell CNAs based on deep learning is becoming a new paradigm of scDNA-seq data analysis, and DeepCNA is an enhancement to the current arsenal of computational methods for investigating cancer genomics.

Keywords Single-cell DNA sequencing, Copy number alteration, Autoencoder, Mixture model

Background

Cancer genomes are often featured with extensive aberrations such as copy number alterations (CNA) and point mutations [1]. Accurate detection of the mutation profiles is essential for deciphering underlying intra-tumor heterogeneity (ITH) and elucidating tumor evolutionary history [2, 3]. Specifically, characterizing CNAs from high-throughput sequencing data has emerged as an important field of cancer related studies.

*Correspondence:

Zhenhua Yu
zhyu@nxu.edu.cn

¹ School of Information Engineering, Ningxia University, Yinchuan 750021, China

² Collaborative Innovation Center for Ningxia Big Data and Artificial Intelligence Co-Founded By Ningxia Municipality and Ministry of Education, Ningxia University, Yinchuan 750021, China



© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

Conventional bulk sequencing has spawned a number of computational methods for CNA calling based on read counts data [4–6]. A bulk sample contains the DNA of thousands or even millions of cells, therefore inferred CNAs are the weighted average of the mutational profiles of present tumor clones. To understand the clonal structure, one needs to deconvolve the mixed profile into distinct clones and simultaneously infer their respective mutations. A main challenge of this deconvolution comes from the ambiguity that the number of tumor clones, their relative sizes, their mutational states and their genealogy are all undetermined [7]. Nowadays single-cell DNA sequencing (scDNA-seq) is becoming a powerful means for profiling single-cell mutations [8], and makes it convenient to elucidate ITH based on single-cell CNAs. scDNA-seq uses whole-genome amplification (WGA) process to generate sufficient DNA material for sequencing from a single cell. The generated nucleotide sequences from WGA are often confounded by amplification bias and errors [9, 10], which results in over-dispersed read counts and makes CNA calling methods originally designed for bulk data less effective to noisy scDNA-seq data.

To date, a number of computational methods [11–19] have been specifically developed to infer single-cell CNAs. These methods adopt a general pipeline that mainly consists of three steps: 1) divide the genome into bins and count reads within each bin (the size of bins can be fixed or variable); 2) preprocess read counts to correct bias caused by GC-content and mappability, remove outlier bins having extremely high or low read counts, and exclude cells that show significantly uneven coverage or are low in sequencing coverage; 3) identify breakpoints that partition the genome into segments (any two adjacent segments have different copy numbers), and call absolute copy number for each segment [20]. Most of the approaches require normal cells or a composite normal sample as negative control. For instance, SCOPE [13] uses Gini coefficient to identify normal cells and remove outlier cells, then jointly segments the cells using a generalized likelihood ratio test as well as a modified Bayes information criterion (BIC). SCNV [12] uses normal cells to construct a composite control, and detects CNAs bases on a bin-free segmentation algorithm. Another interesting method SCICoNE [14] is developed to detect CNAs based on reconstruction of a CNA tree. By using SCOPE to preprocess the read counts, SCYN [15] employs a dynamic programming algorithm to segment the genomes. SeCNV [18] is developed to estimate copy number profiles from large scDNA-seq dataset, it needs to find normal cells from input data and uses them as the negative control. A recently proposed method SCONCE [17] uses hidden Markov model (HMM) to

call copy number segments for tumor cells, and uses normal cells as controls to attenuate the effects of read counts fluctuation. Aforementioned advanced methods generally require normal cells to normalize read counts for better detection of CNAs, and how to accurately infer single-cell CNAs without negative controls is a hard challenge. To address this problem, rcCAE [19] employs a convolutional autoencoder to distill copy number information from noisy read counts without normal cells as the controls, it can jointly infer tumor clones and single-cell CNAs. However, rcCAE tends to over-segment the genome because it detects breakpoints under a cell-by-cell manner, which may complicate downstream phylogenetic analysis. Cross-cell segmentation tends to deliver more accurate predictions of the breakpoints since breakpoints are shared among single cells and could be treated as a process that identifies consecutive bins having highly similar read counts across cells. Considering current scDNA-seq data could contain thousands of cells, segmentation over such a high-dimensional space may suffer from high computation intensity and degraded accuracy, therefore methods for this purpose is still highly required especially on large datasets.

Deep learning models have shown excellent performance in learning latent representations of high-dimensional single-cell data. For instance, variational autoencoder models enable more accurate clustering of single cells over a latent space [21, 22]. The encoding–decoding process could be treated as a distillation process that fetches effective latent representations from noisy single-cell data by reconstructing the original data. For cross-cell segmentation of the read count sequence, each genomic bin could be encoded into a latent space using an autoencoder model, thus breakpoints could be found by efficient segmentation over the learned low-dimensional sequential data.

In this paper, we introduce a novel computational method called DeepCNA for inferring single-cell CNAs (as shown in Fig. 1). In DeepCNA, we use an autoencoder to learn low-dimensional latent representation of each genomic bin, then apply the circular binary segmentation (CBS) algorithm [23] on each dimension of the latent data to find the breakpoints, and finally employ a mixture model to estimate copy numbers for each cell. We adopt several data filtering and normalization approaches to improve the quality of read counts data. By minimizing reconstruction loss between the recovered and original data, the autoencoder is encouraged to deliver effective representations of genomic bins. Given the inferred segments by the CBS algorithm, we formulate observed data with a mixture model where copy number state of each segment acts as a latent variable and is efficiently optimized by using the expectation–maximization (EM)

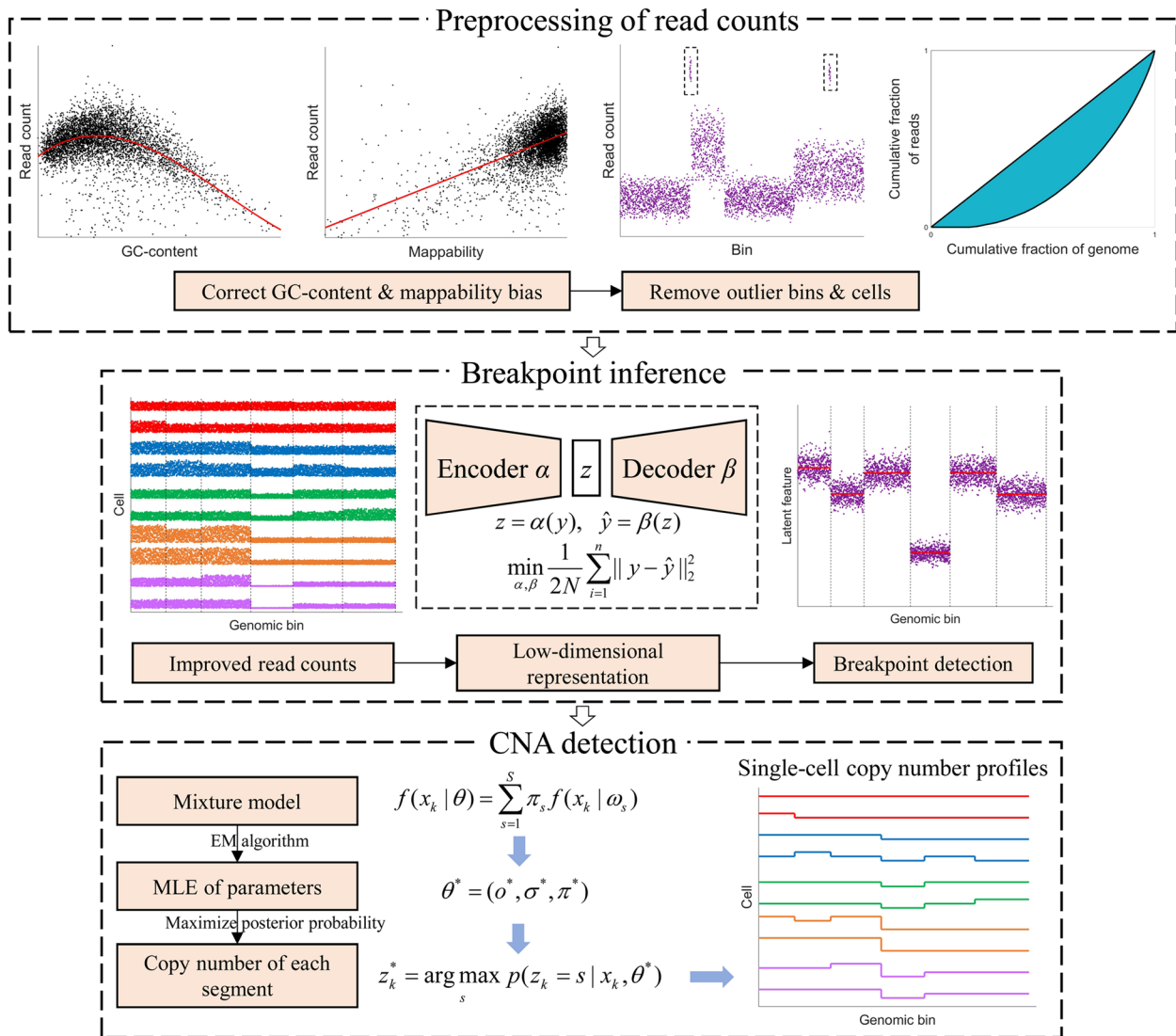


Fig. 1 The workflow of DeepCNA. To improve read counts obtained from scDNA-seq data, DeepCNA takes several data correction and normalization procedures including correcting GC-content and mappability bias, eliminating outlier bins with extremely high or low read counts, and removing outlier cells with low sequencing quality. Given the normalized read counts, DeepCNA employs an autoencoder network to learn low-dimensional latent representation of each genomic bin, such that the breakpoints can be accurately detected along each latent dimension and merged together to form the final breakpoints. Finally, a mixture model is adopted to estimate copy numbers of the inferred segments for each cell

algorithm. Unlike previous methods in this field, DeepCNA performs cross-cell segmentation through encoding each genomic bin into a low-dimensional space, and thus makes it convenient to directly apply conventional segmentation approaches on the data. We comprehensively evaluate DeepCNA on both simulated and real datasets to show its effectiveness

in detecting breakpoints and deciphering single-cell CNAs from complex scDNA-seq data.

Materials and methods

Single-cell DNA sequencing datasets

Simulated datasets

To fully assess the effectiveness of DeepCNA, we generate various datasets by emulating different cell ploidy.

The simulation pipeline is implemented by following our previous study [19]. Specifically, the simulation consists of three steps: 1) construct a clonal tree following the approach adopted in [24]; 2) generate genome sequence for each cell based on simulated CNAs; and 3) produce reads of each cell given biological and technological parameters. When generating the clonal tree, we assume the ancestral cell state is homogeneous with the defined ploidy and then all clones evolve from that state. Nodes of the clonal tree represent tumor clones and edges are labeled with CNAs. The size of simulated CNAs ranges between 3 and 20 Mb, the number of clones is set to 4, and the number of cells is set to 100. Given the CNAs of each cell, SCSSim [25] tool is employed to generate reads under sequencing coverage of 0.02. Read alignments are obtained using BWA [26] tool under default parameters, and further processed with SAMtools [27] to generate BAM files. We generate diploid, triploid and tetraploid datasets to examine the ability of DeepCNA in distinguishing between different tumor ploidy. For each tumor ploidy, the simulation is repeated 10 times, resulting in 30 datasets for benchmarking.

Real datasets

Two real datasets are employed in this study. The first dataset consists of 100 single cells from a breast ductal carcinoma patient [28], and sequencing data can be downloaded from NCBI SRA under accession number SRA018951. The second dataset is a 10X Genomics dataset containing 2053 cells from a triple negative ductal carcinoma [16], and sequencing data are freely available at https://support.10xgenomics.com/single-cell-dna/datasets/1.0.0/breast_tissue_E_2k.

Fetch and preprocess read counts

Read alignments of cells could be stored in a merged BAM file or per-cell BAM files. When a merged BAM file as used in 10X genomics is provided, we use barcodes of the cells to fetch read alignments of each cell and calculate read counts. When per-cell BAM files are available, read counts of each cell are obtained from the corresponding BAM file. The read counts are measured in consecutive equal-sized bins along the genome. We have implemented a tool that uses the APIs of BamTools [29] software to fetch read alignment information from the BAM files. To improve data quality, we also calculate GC-content and mappability of each bin from reference sequence, which are utilized to exclude outlier bins and correct read counts bias. Bins with <10% GC percent, >90% GC percent, or >0.9 mappability score are marked as outliers [30] and excluded from downstream analysis. We then perform library size normalization for each cell by dividing read counts with the mean read

count, which makes read counts comparable between cells. To further filter “bad” bins that have extremely high or low read counts, we exclude bins whose mean read counts are in the lower or upper 1% quantile. Moreover, cells with extensively fluctuated read counts should be eliminated as these cells may act as confounding outliers. Specifically, Gini coefficient is calculated for each cell based on the read counts, and cells that have ≥ 0.3 Gini coefficient are excluded. Finally, we employ a median normalization approach [5] to correct read counts bias caused by mappability and GC-content for each cell:

$$\hat{r}_k = r_k \cdot \frac{m}{m_k} \quad (1)$$

where r_k is the read count of the k -th bin, m is the median read count of all bins, m_k is the median read count of bins that show same GC-content (mappability) as the k -th bin, and \hat{r}_k is the corrected read count. We then calculate log2 read counts (LRC) for downstream analysis. The resulted LRC data of N cells in M bins are stored in a $N \times M$ matrix X .

Learn low-dimensional representations of genomic bins

Each genomic bin is represented by a feature vector y (one col of X) with size of N , and cross-cell segmentation of the LRC data could be treated as a process that identifies consecutive bins having highly similar features. As current scDNA-seq data frequently contain hundreds even thousands of cells, measuring similarity between bins over a high-dimensional space is challenging and requires appropriate selection of the distance metric. To obviate this inconvenience, we project the feature vector of each bin into a d -dimensional space where segmentation of the bins could be efficiently conducted along each dimension and optimize the latent representation by reconstructing the original features. We use a function α to project y into a d -dimensional latent representation z , i.e. $z = \alpha(y)$, and employ a function β to recover y from z , i.e. $\hat{y} = \beta(z)$. Specifically, the function α and β are implemented as the encoder and decoder of an autoencoder network, respectively. The encoder consists of three fully-connected layers each with 256, 128 and 64 neurons, while the decoder is composed of the mirrored structure. The network is trained by minimizing the following reconstruction loss:

$$L(y, \hat{y}, \alpha, \beta) = \frac{1}{2N} \|y - \hat{y}\|_2^2 \quad (2)$$

we use gradient decent algorithm to update the network weights by minimizing the mean loss of a batch of samples in each iteration. After the model converges, we get a d -dimensional latent representation of each genomic bin

by taking the LRC data of the bin as input of the encoder network, and latent features of M bins are denoted by a $d \times M$ matrix Z .

Perform segmentation on the learned low-dimensional sequential data

Based on the learned representation Z , we employ the CBS algorithm [23] to segment each row of Z , and merge the breakpoints from all rows to get the final breakpoints. To minimize the number of false negative calls for the breakpoints, we set p -value to 0.1 in the CBS algorithm. This may tend to result in over-segmented results containing false positive calls of the breakpoints, while the results could be recalibrated by merging consecutive segments that have same copy number. We use K to denote the number of inferred segments, and L_k to represent the number of bins contained in the k -th segment.

Estimate copy numbers for each cell

We estimate absolute copy numbers of the segments for each cell by maximizing the likelihood of LRC data. Without loss of generality, we use $x = (x_1, x_2, \dots, x_K)$ to denote LRC data of a cell, where x_k indicates LRC data of the bins in the k -th segment. Suppose there are S copy number states, the probability density of LRC in the k -th segment is formulated with a mixture model:

$$f(x_k|\theta) = \sum_{s=1}^S \pi_s f(x_k|z_k = s, \theta) = \sum_{s=1}^S \pi_s f(x_k|\omega_s) \quad (3)$$

where we assume the segments are independent of each other, θ denotes model parameters, s is a copy number state, z_k denotes copy number state of the k -th segment, π_s is the proportion of the s -th component in the mixture model, and ω_s represents parameters associated with the s -th component. The conditional likelihood $f(x_k|\omega_s)$ can be formulated as $\prod_{i=1}^{L_k} f(x_{ki}|\omega_s)$, here x_{ki} is LRC of the i -th bin of the k -th segment. For computational convenience, we assume x_{ki} follows a normal distribution:

$$f(x_{ki}|\omega_s) = f(x_{ki}|\mu_s, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_{ki} - \mu_s)^2}{2\sigma^2}\right) \quad (4)$$

with σ being standard deviation and μ_s being the mean value. Considering the mean value of LRC in copy-neutral regions may deviate from 0 due to aneuploidy, we define μ_s by introducing a parameter o that quantifies baseline shift of LRC as adopted in [19]:

$$\mu_s = \log_2(0.5c_s) + o \quad (5)$$

where c_s is the copy number under state s . The complete likelihood of all segments is then computed as:

$$\log(f(x|\theta)) = \log\left(\prod_{k=1}^K f(x_k|\theta)\right) = \sum_{k=1}^K \log(f(x_k|\theta)) \quad (6)$$

To infer the hidden states $z = (z_1, z_2, \dots, z_K)$ and model parameters, we employ EM algorithm to train the model. During the n -th iteration of the EM algorithm, the posterior probability of z_K is measured as:

$$\gamma_{ks}^{(n)} = p(z_k = s | x_k, \theta^{(n-1)}) = \frac{\pi_s^{(n-1)} f(x_k | \omega_s^{(n-1)})}{f(x_k | \theta^{(n-1)})} \quad (7)$$

where $\theta^{(n-1)}$ denotes the estimated parameters in the previous iteration. We then maximize the following objective function to update model parameters $\theta = (o, \sigma, \pi)$:

$$J(\theta) = \sum_{k=1}^K \sum_{s=1}^S \gamma_{ks}^{(n)} (\log(\pi_s) + \sum_{i=1}^{L_k} \log(f(x_{ki}|\omega_s))) \quad (8)$$

By solving the optimization problem, we get following rule to update parameter o :

$$o^{(n)} = \frac{\sum_{k=1}^K \sum_{s=1}^S \gamma_{ks}^{(n)} \sum_{i=1}^{L_k} (x_{ki} - \log_2(0.5c_s))}{\sum_{k=1}^K \sum_{s=1}^S \gamma_{ks}^{(n)} L_k} \quad (9)$$

and then update σ with

$$\sigma^{(n)} = \sqrt{\frac{\sum_{k=1}^K \sum_{s=1}^S \gamma_{ks}^{(n)} \sum_{i=1}^{L_k} (x_{ki} - \mu_s^{(n)})^2}{\sum_{k=1}^K \sum_{s=1}^S \gamma_{ks}^{(n)} L_k}} \quad (10)$$

We update parameter π under the constraint $\sum_{s=1}^S \pi_s = 1$:

$$\pi_s^{(n)} = \frac{\sum_{k=1}^K \gamma_{ks}^{(n)}}{K} \quad (11)$$

By initializing σ to the standard deviation of LRC data and performing a linear search of initial values of o , we iteratively update the model parameters until the algorithm converges and the optimal parameters are denoted by θ^* . Copy number of each segment is then inferred from the state that has the largest posterior probability:

$$z_k^* = \arg \max_s p(z_k = s | x_k, \theta^*) \quad (12)$$

As the number of copy number states considered in the mixture model is limited, copy numbers of highly amplified segments are underestimated. To address this issue, we use the post-processing approach adopted in rcCAE [19] to recalculate copy number of each segment given the learned model parameters and mean of LRC data of the segment.

Performance evaluation

We compare DeepCNA to six advanced methods including SCOPE [13], SCYN [15], SCICoNE [14], SeCNV [18], SCONCE [17] and rcCAE [19] on the simulated datasets. CHISEL [16] is excluded from performance evaluation since it additionally exploits allele frequency to call allele-specific CNAs, and phasing based on single nucleotide variations is critically challenging due to low coverage of scDNA-seq data. As data preprocessing strategy varies across different methods and size of the resulted data may be inconsistent, we extract common cells and common bins from the results of all methods for performance evaluation. To evaluate the accuracy of breakpoint detection, we adopt two performance metrics that are similar to the ones adopted in SCONCE: 1) on each simulated dataset, we fetch the set of ground truth breakpoints, and measure the distance to the nearest predicted breakpoint for each real breakpoint, then sum the distances across the genome to indicate the breakpoint detection accuracy; 2) as inferring large number of false positive breakpoints tends to decrease the breakpoint distance, we calculate w as the ratio between the number of inferred breakpoints and the number of real breakpoints, such that lower breakpoint distance and w closer to 1 indicate higher detection accuracy. To evaluate copy number estimation performance of each method, we calculate mean of absolute differences (MAD) between real copy number and estimated copy number for each cell across the common bins [17]. To assess ploidy estimation performance, average copy number (ACN) is calculated for each cell based on the inferred copy number profiles and compared to the ground truth [19]. We follow same strategy as used in rcCAE to run each method, and specific

parameters used to run each method are given in Supplementary Table 1 (Additional file 1).

Results

Evaluation of DeepCNA on simulated datasets

We run DeepCNA on the simulated datasets to examine its ability of detecting breakpoints and single-cell CNAs. The read counts data are obtained from every 200 kb bins (~ 3450 bins), and ~ 2955 bins are remained after applying the preprocessing steps. For model training, we set the latent dimension d to 1, and train the autoencoder network 500 epochs with learning rate of 0.0001 and batch size of 256. The maximum copy number is set to 10 when calling single-cell CNAs. The following sections give the detailed performance evaluation results.

DeepCNA shows better performance in inferring breakpoints

DeepCNA uses an autoencoder network to project the high-dimensional read counts into a low-dimensional latent sequence, then employs the CBS algorithm to segment the chromosomes into distinct regions. An example of inferred latent representation and breakpoints on the simulated datasets is depicted in Supplementary Fig. S1 (Additional file 1). It is observed that breakpoints associated with some small CNAs (located on chromosome 2) are also effectively characterized in the latent space and thus accurately detected by the CBS algorithm. The autoencoder network is trained to automatically learn the representation of read count sequence, obviating the need for design and calculation of specific statistics for each bin as adopted in the existing methods. This distinctive feature makes DeepCNA well identify the changes along the read count sequence.

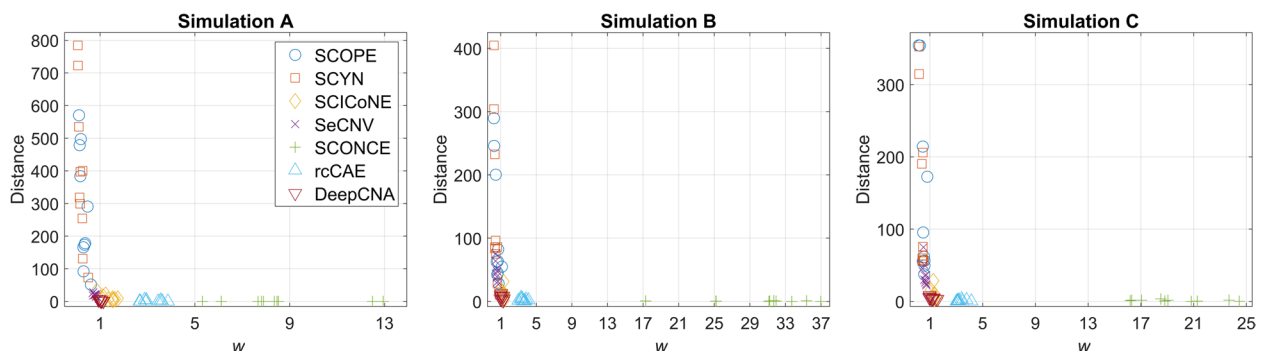


Fig. 2 Breakpoint inference accuracy of the investigated methods on simulated datasets. Two performance metrics are calculated for evaluation: 1) we measure the distance to the nearest predicted breakpoint for each real breakpoint, then sum the distances for all real breakpoints; 2) w is defined as the ratio between the number of inferred breakpoints and the number of real breakpoints. A lower breakpoint distance coupled with w closer to 1 indicate higher inference accuracy. Performance of each method on simulated diploid (Simulation A), triploid (Simulation B) and tetraploid (Simulation C) datasets are evaluated

We make comparison between DeepCNA and other methods in inferring breakpoints, and the results are shown in Fig. 2 (lower breakpoint distance and w closer to 1 indicate better results). On datasets from Simulation A, SCOPE and SCYN exhibit higher breakpoint distance than other methods, and the mean distances are 288.2 and 391.5, respectively. They may fail to find breakpoints that are only present in small proportion of cells. In addition, SCONCE shows the lowest breakpoint distance but yields many false positives (the mean of w is 8.6). Similarly, rcCAE achieves good performance (the mean distance is 1.5) in finding true breakpoints, nevertheless it produces high proportion of false positives (the mean of w is 3.2). As SCONCE and rcCAE detect breakpoints under a cell-by-cell manner, they are sensitive to read count fluctuation occurring in single cells, and thus tend to over-segment the genome. Giving large number of false positive breakpoints complicates downstream ITH analysis such as inferring tumor subclones and phylogeny. Compared to the existing methods, DeepCNA achieves a good tradeoff between precision and recall, it

yields the mean breakpoint distance of 3.2 and mean w of 1.0, which suggests DeepCNA successfully disentangles real breakpoints from technically confounded factors, and effectively represents the read count sequences over the low-dimensional latent space. On datasets from Simulation B and Simulation C, we find each method shows similar performance as observed on Simulation A datasets, and DeepCNA generally surpasses other methods in either finding true breakpoints or suppressing false positives.

DeepCNA achieves competitive performance in estimating single-cell copy numbers

Given the inferred breakpoints, we employ a mixture model to estimate copy numbers of segments for each cell. An example of copy number estimation results on the simulated datasets is shown in Fig. 3, which shows DeepCNA correctly identifies cell ploidy and precisely predicts copy number of each segment. As our method explicitly formulates the relationship between LRC and cell ploidy as well as copy number, it is able to find the

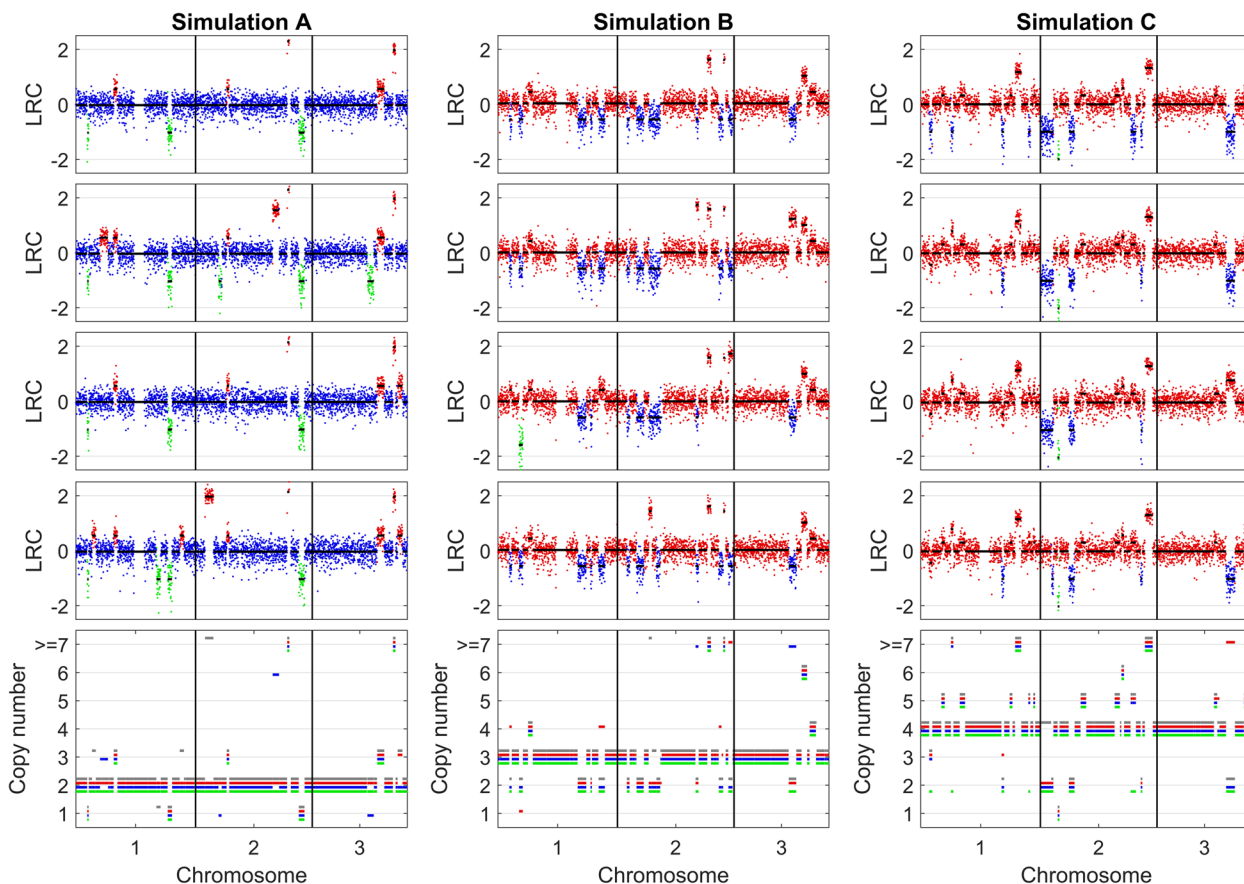


Fig. 3 An illustration of copy number estimation results on simulated datasets. The top four sub-figures depict LRC data of four tumor clones (copy number deletion, neutral and amplification are marked by green, blue and red, respectively), and the bottom subfigure shows the inferred absolute copy numbers of each tumor clone

ground truth ploidy through a line search for the parameter σ as shown in Supplementary Fig. S2.

We evaluate the copy number and cell ploidy estimation accuracy of all methods by calculating the MAD as well as difference between the ground truth and inferred ACNs (Δ ACN). Lower MAD and Δ ACN closer to 0 imply higher estimation accuracy. The copy number and cell ploidy estimation results are shown in Fig. 4 and Supplementary Fig. S3, respectively. On diploid datasets (Simulation A), SCOPE, SCYN, SeCNV and SCONCE exhibit good performance with mean MADs of 0.06, 0.03, 0.02 and 0.09 respectively, while SCICoNE is less accurate in calling absolute copy numbers (mean MAD is 0.72) because it tends to overestimate cell ploidy (median Δ ACN is -0.99). rcCAE accurately estimates copy numbers for most of the cells, but overestimates cell ploidy for 65 cells of which 39 cells are recognized to be tetraploid and 21 cells are predicted to be pentaploid, therefore it produces mean MAD of 0.16. By comparison, our method presents high accuracy in identifying both cell ploidy (median Δ ACN is -0.002) and absolute copy numbers across different datasets, thus reaches the lowest mean MAD of 0.01. On triploid datasets (Simulation B), the mean MADs of SCOPE, SCYN, SCICoNE, SeCNV, SCONCE, rcCAE and DeepCNA are 0.59, 0.43, 0.23, 0.15, 0.81, 0.20 and 0.02, respectively. The lower accuracy of existing methods results from their less effectiveness in estimating cell ploidy. For instance, given 915 common cells covered by the results of all methods, SCOPE erroneously predicts 360 cells as diploid, SCYN tends to underestimate cell ploidy and erroneously identifies 378 cells as diploid, SCICoNE marks all normal cells as triploid, and SeCNV incorrectly labels 107 triploid cells either diploidy or tetraploidy. Despite that rcCAE explicitly takes cell ploidy into consideration in the copy number computation model, it still overestimates ploidy of 74

normal cells. On tetraploid datasets (Simulation C), our method also gains better or comparable performance compared to existing methods. It achieves 0.02 mean MAD and accurately predicts cell ploidy on all datasets. SCOPE, SCYN, SCICoNE, SeCNV and SCONCE tend to underestimate cell ploidy, while rcCAE overestimates ploidy of 54 cells. These results demonstrate DeepCNA is robust to the change of cell ploidy and read count fluctuation.

DeepCNA performs robustly against the change of hyperparameters

There are several hyperparameters, such as bin size and latent dimension, that may affect the detection results of DeepCNA. To fully investigate DeepCNA' robustness against the changes of these hyperparameters, we simulate various datasets for performance evaluation. First, we compare the results on Simulation A datasets by using different bin sizes in {100 kb, 200 kb, 500 kb, 1000 kb}. The results in Supplementary Fig. S4 suggest DeepCNA performs robustly in identifying cell ploidy and copy numbers of segments. Increasing the bin size tends to yield less breakpoints since more real breakpoints will locate inside the bins and thus are undetectable. Second, to assess the effects of latent dimension d on breakpoint inference, we compare the breakpoint distance and metric w under different values of d , and the results in Supplementary Fig. S5 imply the larger the value of d , the smaller the breakpoint distance and the larger the w . This is in concordance with the intuitive conviction that larger latent dimension will result in more breakpoints and thus probably decrease the breakpoint distance. In addition, Supplementary Fig. S6 shows example segmentation results on three simulated datasets when the latent dimension is set to 2. The results suggest the first dimension well captures most of the breakpoints, while

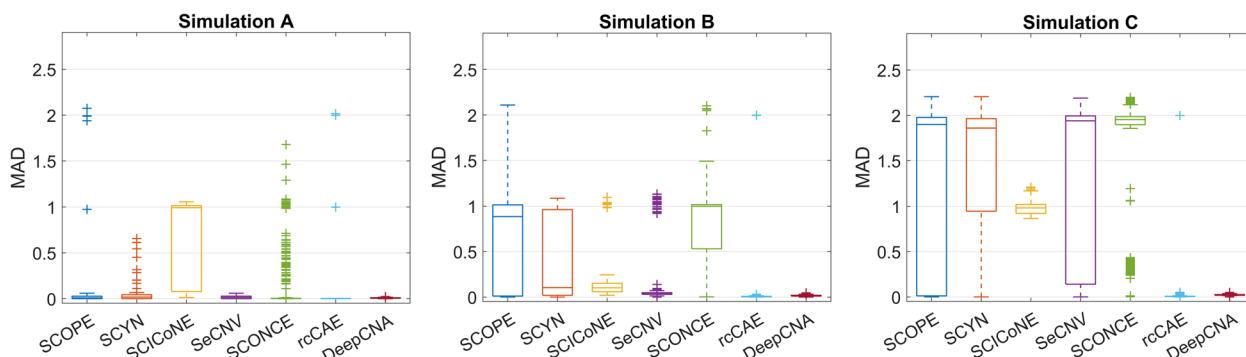


Fig. 4 Copy number estimation accuracy of the investigated methods on simulated datasets. We first extract common bins covered by the results of all methods, then calculate mean of absolute differences (MAD) between real copy number and estimated copy number for each cell across the common bins. Performance of each method on simulated diploid (Simulation A), triploid (Simulation B) and tetraploid (Simulation C) datasets are evaluated

sometimes underrepresents the breakpoints that may be shared by only a few cells, and these breakpoints are detectable from the second dimension. It is also observed that some breakpoints can be simultaneously detected from both dimensions due to the fact that we do not introduce any constraints when learning the different latent dimensions with the autoencoder network. Our results indicate setting latent dimension to 1 or 2 is appropriate to capture local changes of LRC data, accurately detect breakpoints and simultaneously suppress false positives.

DeepCNA shows good robustness against the change of data size and heterogeneity

We also assess DeepCNA's ability of coping with varying data size and heterogeneity. To check if DeepCNA generalizes well on different-sized datasets, we generate simulation datasets by sampling the number of cells from {100, 200, 300} and fixing the number of clones to 8. The evaluation results (Supplementary Fig. S7) suggest our method scales well to larger datasets and performs consistently well in identifying cell ploidy as well as copy numbers. In addition, runtime performance analysis (Supplementary Table 2) shows our method is more efficient than other methods and requires less than 130 s to process a dataset with 300 cells. To further examine the effect of data heterogeneity, we compare detection results of DeepCNA on simulated datasets with different numbers of clones. Specifically, the number of clones changes from 4 to 8, and the number of cells is fixed to 100. The results (Supplementary Fig. S8) show increasing the number of clones does not appreciably affect the performance of DeepCNA, suggesting our method can accurately disentangle copy number segments from noisy read counts regardless of the data heterogeneity. These results demonstrate the applicability of DeepCNA to complex datasets.

We proceed to check if our method can still accurately estimate copy number segments without normal cells mixed in the input data. To achieve this, we exclude normal cells from the simulated datasets (Simulations A-C), and run DeepCNA under same parameter configurations as used on full datasets. The results in Supplementary Fig. S9 demonstrate our method can still yield accurate results, which suggests DeepCNA is applicable to datasets containing only tumor cells.

Evaluation of DeepCNA on a breast cancer dataset

To further verify the effectiveness of DeepCNA, we apply it to a triple negative ductal carcinoma dataset (denoted as T10) consisting of 100 cells [28]. Sequencing data of the cells are obtained from NCBI SRA (accession number SRA018951), and further processed using BMA to

generate read alignments. A previous study [28] has analyzed copy number profiles of the cells and identified 4 subpopulations including D+P, H, AA and AB. The copy number profiles indicate D+P comprises mainly diploid cells, H is a tumor clone that undergoes hemizygous deletions on multiple chromosomes, AA and AB are aneuploid clones showing copy number amplifications across the chromosomes.

When running DeepCNA on this dataset, we use bin size of 500 kb to calculate the read counts, and finally obtain an LRC matrix with size of 100×5120 after applying the preprocessing steps. For training the autoencoder network, the latent dimension, number of epochs, batch size and learning rate are set to 2, 500, 256 and 0.0001, respectively. We randomly select a cell from each subpopulation to compare the copy number profiling results of DeepCNA as shown in Fig. 5. As expected, cell "SRR053611" from the D+P subpopulation is a diploid cell that exhibits few CNAs across the genome, and cell "SRR053678" from the H clone has extensive hemizygous deletions on the chromosomes such as 1p, 4q, 5q, 13q and 14. Instead, cells "SRR054594" and "SRR054634" are mainly characterized with copy number gains on most of the chromosomes. The ploidy of the cells is similar to the previously reported results [19, 28].

We also run SCOPE, SCYN, SCICoNE and rcCAE on this dataset by setting bin size to 500 kb. SeCNV and SCONCE are excluded from evaluation as SeCNV throws runtime errors when calling CNAs, and SCONCE needs to be manually provided with normal cells to create negative control but we have no prior knowledge about which cells are normal cells. As the ground truth breakpoints and copy number profiles are not available for the T10 dataset, we just perform a qualitative analysis of the results inferred by different methods. In general, DeepCNA yields results that better explains the observed data (Supplementary Figs. S10-13), and SCICoNE tends to overestimate the cell ploidy as observed on the simulated datasets. For instance, a copy number amplification on chromosome 7 of cell "SRR053675" is detected by DeepCNA (Supplementary Fig. S10), while SCOPE, SCYN and rcCAE predict it as normal copy number, and their predictions obviously deviate from the observed distribution of depth or LRC data. In addition, our method detects a small copy number segment on chromosome 7 of cell "SRR054618" that is missed by SCOPE, SCICoNE and rcCAE (Supplementary Fig. S11). Similarly, existing methods except SCYN underestimate the number of breakpoints on chromosome 2q of cell "SRR089402" (Supplementary Fig. S12), thus yield biased predictions of the copy numbers.

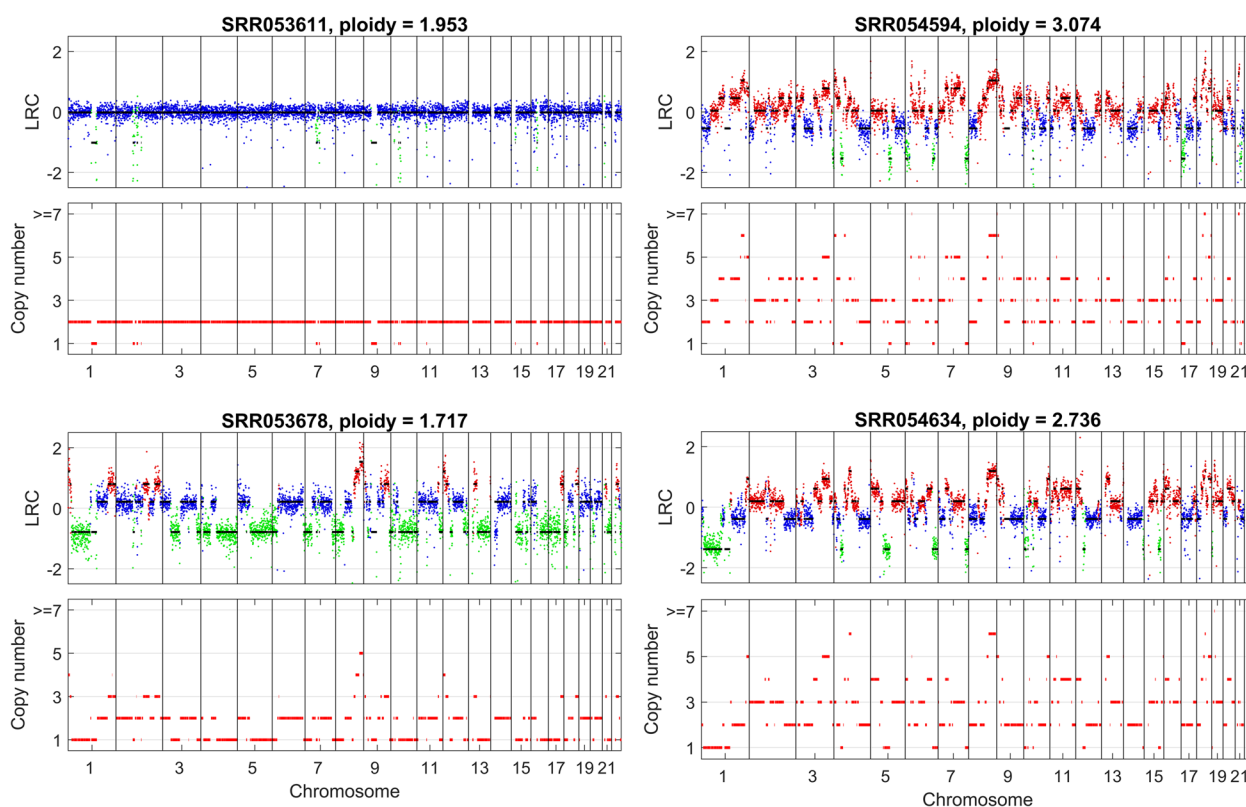


Fig. 5 Copy number estimation results of DeepCNA on the breast cancer dataset. A previous study [28] has clustered the cells into 4 subpopulations, and we select four cells for comparison including “SRR053611” from D+P subpopulation, “SRR053678” from H clone, “SRR054594” from AA clone, and “SRR054594” from AB clone. Cell “SRR053611” is a diploid cell, and cell “SRR053678” shows hemizygous deletions on multiple chromosomes. Instead, cells “SRR054594” and “SRR054634” have copy number amplifications on most of the chromosomes

Evaluation of DeepCNA on a 10X Genomics dataset

We also evaluate DeepCNA on a large dataset including 1446 cells from a breast tumor tissue [16], and sequencing data of this 10X Genomics dataset (denoted as E) can be freely downloaded from https://support.10xgenomics.com/singlecell-dna/datasets/1.0.0/breast_tissue_E_2k. Sequencing coverage of each cell varies between 0.02X and 0.05X. The original BAM file contains read alignment results of 2053 cells, of which 607 cells are identified to be low sequencing coverage, doublets or with S-phase of the cell cycle [16], and therefore eliminated from CNA analysis. For the remaining 1446 cells, CHISEL algorithm [16] has found one diploid subpopulation comprised of 388 cells and also 5 tumor clones (labeled as I-V), of which each contains 168, 58, 20, 782 and 30 cells, respectively. We use DeepCNA to call breakpoints as well as estimate single-cell copy numbers from this dataset, and check if it could give reliable detection results. As the ground truth breakpoints and copy numbers are not available, we therefore use the results of CHISEL as the baseline to evaluate different methods. This is appropriate since the copy number profiles inferred by CHISEL are further

verified by allele frequency data. CHISEL gives a CNA matrix with size of 1446×570 obtained from 5 Mb consecutive bins, and we find 104 breakpoints by scanning the columns of the matrix. For performance comparison, the existing methods except rcCAE are unable to process the merged BAM file, therefore we only obtain the results of rcCAE on this dataset.

Read counts in 500 kb bins are fetched from the BAM file and further preprocessed to improve the data quality, which results in an LRC matrix with size of 1446×5139 . By setting the latent dimension d to 2, we train the autoencoder network 500 epochs with learning rate of 0.0001 and batch size of 256. We randomly select a cell from each subpopulation inferred by CHISEL and compare the breakpoints and copy number estimation results of DeepCNA as shown in Fig. 6. There are significant differences between copy number profiles of distinct cell subpopulations. Cell with barcode “TAAGAG ATCAAGAAAC” is a diploid cell that presents neutral copy number across the genome. The distinctive feature of cell “ATTGGACTCCCAAAGT” from tumor clone I is that it has copy number of 3 on chromosomes 2–3

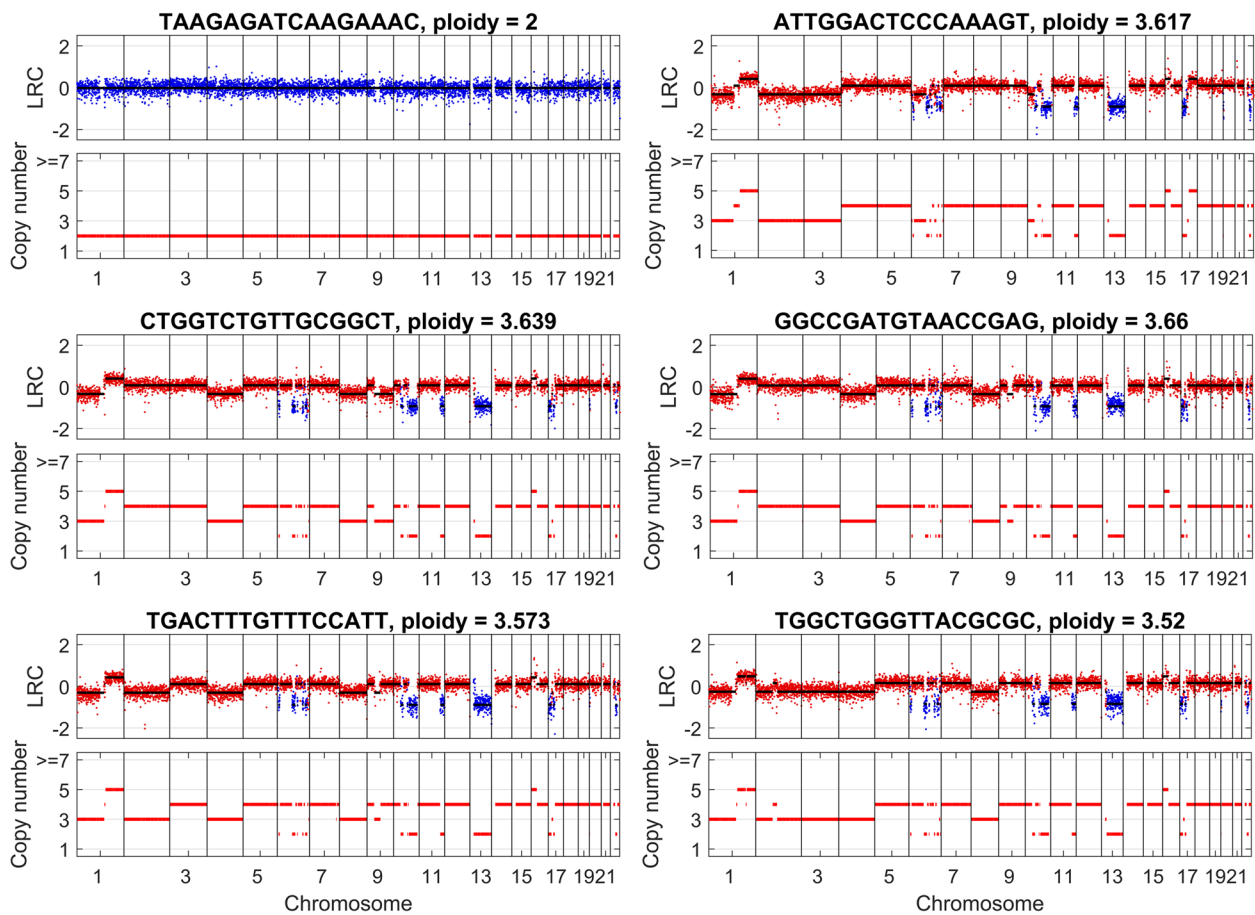


Fig. 6 Copy number estimation results of DeepCNA on the 10X Genomics dataset. CHISEL algorithm [16] has clustered the cells into 6 subgroups, and we select one cell from each subpopulation for illustration. Copy number profiles of these cells exhibit significant differences. For instance, cell with barcode “TAAGAGATCAAGAAAC” is a diploid cell that shows few CNAs across the genome; Cells “CTGGTCTGTTGCGGCT” and “GGCCGATGTAACCGAG” show different copy number on chromosome 9q; compared to cell “GGCCGATGTAACCGAG”, cell “TGACTTTGTTCCATT” has copy number of 3 on chromosome 2, and cell “TGGCTGGGTTACGCGC” shows copy number of 3 on chromosomes 2–3

while copy number of 4 on chromosome 4. By comparison, the corresponding copy numbers of cell “CTGGTCTGTTGCGGCT” from clone II are 4 and 3, respectively. Cell “GGCCGATGTAACCGAG” from clone III has very similar copy number profiles to the cell “CTGGTCTGTTGCGGCT” on most of the genome regions, but shows different copy number on chromosome 9q. Compared to the cell “GGCCGATGTAACCGAG”, the main difference of cell “TGACTTTGTTCCATT” from clone IV is that its copy number on chromosome 2 is 3, and cell “TGGCTGGGTTACGCGC” from clone V shows copy number of 3 on chromosomes 2–3. Taken together, these results imply our method successfully deciphers the copy number profiles of underlying tumor clones.

To check if our method gains advantage over rcCAE in inferring breakpoints and estimating copy numbers, we calculate three performance metrics including breakpoint distance, w and MAD by treating CHISEL results as

the ground truth. Common bins covered by both methods are used for evaluation. rcCAE infers 450 breakpoints of which most are false positive calls and yields breakpoint distance of 462 and w of 4.3. By comparison, DeepCNA infers 103 breakpoints, it shows larger breakpoint distance of 1270 while a lower w of 0.99. As CHISEL infers copy numbers in low-resolution 5 Mb bins, the real breakpoints may locate far from the bin boundaries, and this may explain why DeepCNA gives similar number of breakpoints but large breakpoint distance. The copy number estimation results may also give evidence that DeepCNA achieves better performance than rcCAE in suppressing false positive calls of breakpoints. The MADs of rcCAE and DeepCNA are 0.08 and 0.07, respectively, which suggests breakpoints inferred by DeepCNA are more likely to be real breakpoints and thus aid in accurate estimation of copy numbers. We further investigate if copy numbers shared by few cells are accurately

detected by each method, and results of cell “AAGGCA GGTTCGCGTG” are given in Supplementary Fig. S14. rcCAE produces highly similar predictions to DeepCNA on all chromosomes except chromosome 2, while it probably yields biased predictions on chromosome 2 because the expected LRC mean deviates from the observed distribution of LRC. As the LRC data used in rcCAE are obtained from the outputs of an autoencoder network, and this distillation process may suppress LRC signal of low-prevalence copy numbers. By comparison, the copy numbers estimated by DeepCNA well match the distribution of LRC data. Furthermore, we calculate Pearson correlation coefficient to compare DeepCNA with rcCAE by treating CHISEL results as the baseline. The results in Supplementary Fig. S15 show our method achieves a median Pearson correlation coefficient of 0.92, while rcCAE gets a median Pearson correlation coefficient of 0.89. This gives enhanced evidence that our method performs better than rcCAE on the 10X Genomics dataset.

Evaluation of DeepCNA on non-tumor datasets

We further assess if DeepCNA can still accurately estimate copy numbers from the scDNA-seq datasets that only contain normal/non-tumor cells. Such datasets can be used as negative controls to find somatic single-cell CNAs.

First, we apply DeepCNA to a composite control dataset that is a subset of the 10X Genomics dataset. Specifically, as the diploid subpopulation of the 10X Genomics dataset is mainly composed of normal cells, we use the 388 cells contained in this subpopulation to construct a negative control dataset, and run DeepCNA under the same parameters as used for the full dataset. The results show our method achieves high accuracy in measuring copy numbers of the normal cells with mean MAD of 0.01. Despite that DeepCNA yields more breakpoints than the expected (78 vs. 45), copy numbers of the over-segmented regions are still precisely estimated.

Next, we evaluate DeepCNA on a non-tumor dataset consisting of 1094 cells from BJ Fibroblast Euploid Cell Line, and the merged BAM file containing read alignments of all cells can be freely downloaded from <https://www.10xgenomics.com/resources/datasets/1-k-cells-from-bj-fibroblast-euploid-cell-line-1-standard-1-1-0>. As the ground truth copy number profiles are not available for this dataset, we only perform qualitative analysis of the results of DeepCNA. Read counts are measured for 500 kb bins and preprocessed to filter outlier cells and bins, which results in an LRC matrix with size of 1094×5120 . We then train the autoencoder model 500 epochs with learning rate of 0.0001, batch size of 256 and latent dimension of 1. DeepCNA predicts all cells as diploidy with the minimum ACN of 1.931 and maximum

ACN of 2.072, which suggests our method correctly identifies the ploidy of all cells. An example of the copy number estimation results as depicted in Supplementary Fig. S16 indicates the cells present broad copy neutral regions across the whole genome. These results demonstrate that our method can still yield reliable estimations of single-cell copy number profiles from scDNA-seq datasets composed of only non-tumor cells, and thus help to find somatic single-cell CNAs.

Discussion

Breakpoint detection and copy number estimation for each segment are two basic tasks of single-cell copy number analysis. As tumor evolves by accumulating CNAs, breakpoints are shared among different cell subpopulations, and cross-cell segmentation of high-dimensional read count sequence is essential for reducing false positive calls of the breakpoints caused by local fluctuation of read counts. Existing methods for breakpoint detection are usually based on manually designed statistics of read counts, and less sensitive to real breakpoints or less robust to local noise of read counts. In this paper, we develop a novel method for cross-cell segmentation and single-cell copy number estimation, the proposed DeepCNA treats read counts of all cells of each bin as a feature vector and automatically learns its latent representations by leveraging an autoencoder network, which makes it convenient to call breakpoints along each latent dimension and yield the final set of breakpoints by merging the results from all dimensions. The approach exploits the distillation process of the autoencoder to effectively represent each bin over a low-dimensional space and gets higher robustness to the outliers present in read counts. To the best of our knowledge, DeepCNA is the first deep learning method that performs cross-cell segmentation of read count sequence. For estimating single-cell copy numbers, a mixture model is employed to cluster segments into different copy number states, and cell ploidy is determined by exploring the baseline shift of LRC data. Although DeepCNA follows the same idea as proposed in rcCAE to estimate cell ploidy, its main difference to rcCAE lies in the fact that DeepCNA jointly segments the read counts of all cells while rcCAE segments the read counts under a cell-by-cell manner, and this makes DeepCNA more powerful in suppressing false positive calls of breakpoints.

To comprehensively evaluate the performance of DeepCNA, we generate various datasets by mimicking different cell ploidy. Evaluation results suggest our method is highly effective in finding breakpoints and recognizing cell ploidy, thus yields accurate estimations of copy numbers. With DeepCNA, we elucidate single-cell copy number profiles from a breast cancer dataset, and obtain

similar prediction results as previously reported. Compared to the existing methods, DeepCNA generally better explains the observed depth data. In addition, we test DeepCNA on a large 10X Genomics dataset containing 1446 cells, and the estimation results provide clear indication of copy number profile difference between cell subpopulations. We also show DeepCNA is able to call low-prevalence CNAs, and these are missed by rcCAE due to deficient recovery of the LRC signal for low-prevalence copy numbers.

Despite that DeepCNA shows advantages in inferring breakpoints and single-cell copy number profiles, there are still two potential directions to further improve its accuracy. First, DeepCNA does not consider the relative positions of bins when learning latent representation, which probably makes DeepCNA yield over-segmented results when processing extensively fluctuated read counts. Positional encoding is an effective way to represent the relative position of each bin and can be used as additional features when learning the latent representation. Therefore, how to encode the positions and utilize positional encoding features in the representation learning model is one of our future works to improve DeepCNA. Second, DeepCNA implements breakpoint detection and copy number estimation as two separate procedures, and errors made in breakpoint detection may be propagated to the CNA analysis. Joint inference of breakpoints and copy numbers from all cells is a feasible solution to improve the accuracy, while this may be a very challenging task as the candidate breakpoints and copy number states form a huge search space. Some of the existing methods have employed HMMs to simultaneously detect breakpoints and CNAs under a cell-by-cell manner, and mixture hidden Markov model (MHMM) [31] could be used to implement cross-cell segmentation and copy number estimation in a single model provided that the hidden states and mixture components are properly defined. The mixture components could be defined as cell subpopulations showing distinct copy number profiles between each other, which makes it convenient to identify tumor clones.

Conclusions

In summary, we introduce a new method DeepCNA for profiling single-cell CNAs from scDNA-seq data. DeepCNA leverages deep representation learning to cope with the difficulty of segmenting high-dimensional read count sequence and employs an effective mixture model to decipher copy numbers for each cell. Evaluation results demonstrate DeepCNA has advantages over other methods in inferring breakpoints and single-cell CNAs. We believe our work will promote development of computational methods for downstream ITH and phylogenetic analysis.

Abbreviations

scDNA-seq	Single-cell DNA sequencing
CNA	Copy number alteration
ITH	Intra-tumor heterogeneity
AE	Autoencoder
BIC	Bayesian information criterion
HMM	Hidden Markov model
CBS	Circular binary segmentation
ACN	Average copy number

Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s12864-023-09901-5>.

Additional file 1: Fig. S1. An example of segmentation results on simulated datasets. The top four subfigures depict LRC data of four tumor clones, and the bottom subfigure shows the learned latent sequence and segmentation results. **Fig. S2.** Analysis of the relationship between the ground truth average copy number and estimated baseline shift of LRC signal. Data marked by blue, green and red are from tetraploid, triploid and diploid cells, respectively. **Fig. S3.** Comparison of cell ploidy estimation results between different methods. Average copy number (ACN) is calculated for each cell based on the inferred copy number profiles, and difference between the ground truth and inferred ACNs (denoted as Δ ACN) is analyzed for each method. Values of Δ ACN that are closer to 0 indicate better estimation results. **Fig. S4.** Analysis of the effect of bin size on copy number segments detection accuracy of DeepCNA. The values in {100kb, 200kb, 500kb, 1000kb} are tested for the bin size. Four performance metrics including mean of absolute distances (MAD), difference between the ground truth and inferred average copy numbers (denoted as Δ ACN), breakpoint distance and ratio between the number of inferred breakpoints and the number of real breakpoints (denoted as w) are calculated for comparison. **Fig. S5.** Analysis of the effect of latent dimensionality on breakpoint detection accuracy. The values in {1, 2, 3} are tested for the latent dimensionality d . Breakpoint distance and the metric were calculated for comparison. Data points marked with an asterisk represent mean values. **Fig. S6.** Segmentation results of DeepCNA on three simulated datasets when the latent dimension is set to 2. The first dimension well captures most of the breakpoints, while sometimes under-represents the breakpoints that may be shared by only a few cells, and these breakpoints are detectable from the second dimension. It is also observed that some breakpoints can be simultaneously detected from both dimensions. **Fig. S7.** Analysis of DeepCNA' performance on different-sized datasets. The numbers of cells in {100, 200, 300} are tested. Four performance metrics including mean of absolute distances (MAD), difference between the ground truth and inferred average copy numbers (denoted as Δ ACN), breakpoint distance and ratio between the number of inferred breakpoints and the number of real breakpoints (denoted as w) are calculated for comparison. **Fig. S8.** Analysis of DeepCNA' performance on datasets with varying data heterogeneity. The numbers of clones change from 4 to 8. Four performance metrics including mean of absolute distances (MAD), difference between the ground truth and inferred average copy numbers (denoted as Δ ACN), breakpoint distance and ratio between the number of inferred breakpoints and the number of real breakpoints (denoted as w) are calculated for comparison. C denotes the number of clones. **Fig. S9.** Analysis of DeepCNA' performance on datasets with no normal cells mixed in the data. Four performance metrics including mean of absolute distances (MAD), difference between the ground truth and inferred average copy numbers (denoted as Δ ACN), breakpoint distance and ratio between the number of inferred breakpoints and the number of real breakpoints (denoted as w) are calculated for comparison. **Fig. S10.** Copy number estimation results on cell "SRR053675" from the breast cancer dataset. DeepCNA detects a copy number amplification on chromosome 7, while SCOPE, SCYN and rcCAE predict it as normal copy number, and their predictions deviate from the observed data. SCICoNE overestimates the cell ploidy. Copy number deletion, normal copy number and copy number amplification are marked by green, blue and red, respectively. **Fig. S11.** Copy number estimation results on cell "SRR054618" from the breast cancer dataset. DeepCNA detects a small copy number

segment on chromosome 7 that is missed by SCOPE, SCICoNE and rcCAE. Copy number deletion, normal copy number and copy number amplification are marked by green, blue and red, respectively. **Fig. S12.** Copy number estimation results on cell "SRR089402" from the breast cancer dataset. Existing methods except SCYN underestimate the number of breakpoints on chromosome 2q, thus yield biased predictions of the copy numbers. SCICoNE overestimates the cell ploidy. Copy number deletion, normal copy number and copy number amplification are marked by green, blue and red, respectively. **Fig. S13.** Copy number estimation results on cell "SRR054632" from the breast cancer dataset. SCOPE and SCICoNE predict a small segment on chromosome 6 to copy number amplification. rcCAE fails to find a small segment on chromosome 7. Copy number deletion, normal copy number and copy number amplification are marked by green, blue and red, respectively. **Fig. S14.** Copy number estimation results on the cell with barcode "AAGGCAGTTTCGCGTG" from the 10X Genomics dataset. rcCAE' predictions on chromosome 2 probably deviate from the ground truth. Copy number deletion, normal copy number and copy number amplification are marked by green, blue and red, respectively. **Fig. S15.** Comparison of Pearson correlation coefficients on the 10X Genomics dataset. By using CHISEL results as the baseline, Pearson correlation coefficients are calculated based on copy numbers inferred by each method. **Fig. S16.** An example of copy number estimation results of DeepCNA on the BJ Fibroblast Euploid Cell Line dataset. DeepCNA predicts all cells as diploidy. **Table S1.** The parameters used to run each method on simulated datasets. **Table S2.** The runtime performance comparison results of all investigated methods. The results are obtained from simulated datasets with varying number of cells. All the experiments are conducted on a computational server with 64 CPU cores, 128 GB RAM and 1 GeForce RTX 2080 Ti GPU.

Acknowledgements

Not applicable.

Authors' contributions

Z.Y conceived the study and designed the algorithm. F.L implemented the algorithm. F.L and Z.Y drafted the manuscript. F.L and F.S analyzed the data. Z.Y supervised the software development and manuscript preparation. All authors read and approved the final manuscript for publication.

Funding

This work was supported by the Key Research and Development Program of Ningxia (Grant No. 2023BEG02009) and Natural Science Foundation of Ningxia Province (Grant No. 2023AAC05006). The funders had no role in the design of the study, collection, analysis and interpretation of data and in writing the manuscript.

Availability of data and materials

The T10 dataset is available from NCBI SRA (accession number SRA018951), the E dataset is available from https://support.10xgenomics.com/single-cell-dna/datasets/1.0.0/breast_tissue_E_2k, and the BJ Fibroblast Euploid Cell Line dataset is available from <https://www.10xgenomics.com/resources/datasets/1-k-cells-from-bj-fibroblast-euploid-cell-line-1-standard-1-1-0>.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 17 May 2023 Accepted: 12 December 2023

Published online: 02 January 2024

References

- Greaves M, Maley CC. Clonal evolution in cancer. *Nature*. 2012;481(7381):306–13.
- Burrell RA, Mcgranahan N, Bartek J, Swanton C. The Causes and Consequences of Genetic Heterogeneity in Cancer Evolution. *Nature*. 2013;501(7467):338–45.
- Turajlic S, Sottoriva A, Graham T, Swanton C. Resolving genetic heterogeneity in cancer. *Nat Rev Genet*. 2019;20(7):404–16.
- Alkodsai A, Louhimo R, Hautaniemi S. Comparative analysis of methods for identifying somatic copy number alterations from deep sequencing data. *Brief Bioinform*. 2015;16(2):242–54.
- Yu Z, Li A, Wang M. CLIMAT-HET: detecting subclonal copy number alterations and loss of heterozygosity in heterogeneous tumor samples from whole-genome sequencing data. *BMC Med Genomics*. 2017;10(1):1–11.
- Yuan X, Li J, Bai J, Xi J. A Local Outlier Factor-Based Detection of Copy Number Variations From NGS Data. *IEEE/ACM Trans Comput Biol Bioinform*. 2021;18(5):1811–20.
- Kuipers J, Jahn K, Beerenwinkel N. Advances in understanding tumour evolution through single-cell sequencing*. *Biochim Biophys Acta*. 2017;1867:127–38.
- Gawad C, Koh WLC, Quake SR. Single-cell genome sequencing: current state of the science. *Nat Rev Genet*. 2016;17(3):175–88.
- Navin NE. Cancer genomics: one cell at a time. *Genome Biol*. 2014;15(8):1–13.
- Yu Z, Liu H, Du F, Tang X. GRMT: Generative Reconstruction of Mutation Tree From Scratch Using Single-Cell Sequencing Data. *Front Genet*. 2021;12: 692964.
- Garvin T, Aboukhalil R, Kendall J, Baslan T, Atwal GS, Hicks J, Wigler M, Schatz MC. Interactive analysis and assessment of single-cell copy-number variations. *Nat Methods*. 2015;12(11):1058–60.
- Wang X, Chen H, Zhang NR. DNA copy number profiling using single-cell sequencing. *Brief Bioinform*. 2018;19(5):731–6.
- Wang R, Lin DY, Jiang Y. SCOPE: A Normalization and Copy-Number Estimation Method for Single-Cell DNA Sequencing. *Cell Syst*. 2020;10(5):445–52.
- Kuipers J, Tuncel MA, Ferreira P, Jahn K, Beerenwinkel N. Single-cell copy number calling and event history reconstruction. *bioRxiv*. 2020.04.28.065755. <https://doi.org/10.1101/2020.04.28.065755>.
- Feng X, Chen L, Qing Y, Li R, Li C, Li SC. SCYN: single cell CNV profiling method using dynamic programming. *BMC Genomics*. 2021;22(5):1–13.
- Zaccaria S, Raphael BJ. Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL. *Nat Biotechnol*. 2021;39(2):207–14.
- Hui S, Nielsen R. SCONCE: a method for profiling copy number alterations in cancer evolution using single-cell whole genome sequencing. *Bioinformatics (Oxford, England)*. 2022;38(7):1801–8.
- Ruohan Y, Yuwei Z, Mengbo W, Xikang F, Jianping W, Shuai Cheng L. Resolving single-cell copy number profiling for large datasets. *Brief Bioinform*. 2022;23(4):264.
- Yu Z, Liu F, Shi F, Du F. rcCAE: a convolutional autoencoder method for detecting intra-tumor heterogeneity and single-cell copy number alterations. *Brief Bioinform*. 2023;24(3):bbad108.
- Mallory XF, Edrisi M, Navin N, Nakhleh L. Methods for copy number aberration detection from single-cell DNA-sequencing data. *Genome Biol*. 2020;21(1):1–22.
- Rashid S, Shah S, Bar-Joseph Z, Pandya R. Dhaka: variational autoencoder for unmasking tumor heterogeneity from single cell genomic data. *Bioinformatics*. 2021;37(11):1535–43.
- Yan J, Ma M, Yu Z. bmVAE: a variational autoencoder method for clustering single-cell mutation data. *Bioinformatics (Oxford, England)*. 2023;39(1):790.
- Olshen AB, Venkatraman ES, Lucito R, Wigler M. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics (Oxford, England)*. 2004;5(4):557–72.
- Ross EM, Markowitz F. OncoNEM: inferring tumor evolution from single-cell sequencing data. *Genome Biol*. 2016;17(1):1–14.
- Yu Z, Du F, Sun X, Li A. SCSsim: an integrated tool for simulating single-cell genome sequencing data. *Bioinformatics (Oxford, England)*. 2020;36(4):1281–2.
- Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*. 2009;25(14):1754–60.

27. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*. 2009;25(16):2078–9.
28. Navin N, Kendall J, Troge J, Andrews P, Rodgers L, McIndoo J, Cook K, Stepansky A, Levy D, Esposito D, et al. Tumour evolution inferred by single-cell sequencing. *Nature*. 2011;472(7341):90–4.
29. Barnett DW, Garrison EK, Quinlan AR, Strömberg MP, Marth GT. BamTools: a C++ API and toolkit for analyzing and managing BAM files. *Bioinformatics* (Oxford, England). 2011;27(12):1691–2.
30. Ha G, Roth A, Lai D, Bashashati A, Ding J, Goya R, Giuliany R, Rosner J, Oloumi A, Shumansky K, et al. Integrative analysis of genome-wide loss of heterozygosity and monoallelic expression at nucleotide resolution reveals disrupted pathways in triple-negative breast cancer. *Genome Res*. 2012;22(10):1995–2007.
31. Helske S, Helske J. Mixture hidden markov models for sequence data: The seqhmm package in r. *J Stat Softw*. 2019;83(3):1–32.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

