DCCNV: Enhanced CNV Detection in Single-Cell Sequencing Using Diffusion Process and Contrastive Learning

Mostafa Karami
mostafa.karami@uconn.edu
University of Connecticut
Department of Computer Science & Engineering
Storrs, Connecticut, USA

Bingjun Li
bingjun.li@uconn.edu
University of Connecticut
Department of Computer Science & Engineering
Storrs, Connecticut, USA

Samson Weiner
samson.weiner@uconn.edu
University of Connecticut
Department of Computer Science & Engineering
Storrs, Connecticut, USA

Sahand Hamzehei
sahand.hamzehei@uconn.edu
University of Connecticut
Department of Computer Science & Engineering
Storrs, Connecticut, USA

Sheida Nabavi
sheida.nabavi@uconn.edu
University of Connecticut
Department of Computer Science & Engineering
Storrs, Connecticut, USA

ABSTRACT

Detecting copy number variations (CNVs) in single-cell DNA sequencing (scDNA-seq) data is challenging due to substantial noise and variability. To address this, we present DCCNV, a novel method that integrates diffusion processes, contrastive learning, and circular binary segmentation (CBS) for reliable CNV detection. Our method employs adaptive k-nearest neighbors (KNN) and multi-scale diffusion to reduce noise while preserving key biological signals, followed by contrastive learning to distinguish true genomic alterations from technical noise. The CBS algorithm is then used to partition the enhanced signals into discrete copy number segments. We compared the performance of DCCNV with those of several current single-cell CNV detection methods, including DeepCopy, rcCAE, SCOPE, SCONE, HMMcopy, SeCNV, as well as filtering-based CNV detection approaches that employ commonly used filters such as Wavelet, Median, and Gaussian filters. This comparison was conducted using both simulated and real data. The results show that DCCNV outperforms these approaches in terms of accuracy and computational efficiency. The code used in this research is publicly available at https://github.com/NabaviLab/DCCNV.

CCS CONCEPTS

• **Computing methodologies** → *Machine Learning; Neural Network; Diffusion Process; Contrastive Learning, Unsupervised learning, Bioinformatics, Computational genomics, Graph algorithms analysis.*

ACM Reference Format:
Mostafa Karami, Bingjun Li, Samson Weiner, Sahand Hamzehei, and Sheida Nabavi. 2024. DCCNV: Enhanced CNV Detection in Single-Cell Sequencing Using Diffusion Process and Contrastive Learning. In *15th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '24), November 22–25, 2024, Shenzhen, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3698587.3701395

1 INTRODUCTION

Single-cell sequencing enables precise genomic, transcriptomic, and epigenomic analysis, significantly advancing cancer research by addressing cancer cell heterogeneity [4, 5, 8, 9, 18, 21]. Its key application is detecting copy number variants (CNVs), which influence genetic diversity and are linked to cancer [23, 25]. CNV identification at the single-cell level reveals malignant clones, rare cell groups, and genetic drivers of disease progression [17]. To detect CNVs in scDNA-seq, read count signals quantify how often specific genomic regions are sequenced, providing a key indicator of CNVs when adjusted for sequencing depth and cellular DNA content [35]. In CNV detection, denoising eliminates unwanted technical interferences while retaining genuine biological signals. It also reduces data dimensionality, simplifying the visualization and interpretation of patterns [38]. Conventional filters like Gaussian [28], median [31], and wavelet [6] reduce scDNA-seq noise but struggle with high dimensionality and complex noise, often failing to fully separate biological and technical signals. Recent computational tools address complex noise patterns and high dimensionality in scDNA-seq but still require improvement when managed by traditional filters. Each CNV detection tool has unique methods and limitations. For instance, SCOPE [33] uses cross-sample segmentation to normalize and estimate copy numbers but is time-consuming. SeCNVs [26] employs structural entropy and a local Gaussian kernel for robust performance, though it struggles with highly noisy data. HMMCopy [27] uses hidden Markov models (HMM) [2] for CNV detection but faces scalability challenges. SCONCE [14] offers precise CNV detection in cancer progression, yet struggles with high noise levels. These methods highlight the need for improved noise management while maintaining accuracy.

To handle the high dimensionality and complex noise in scDNA-seq, deep learning and machine learning methods have emerged. The rcCAE (reconstruction convolutional autoencoder) [36] uses

BCB '24, November 22–25, 2024, Shenzhen, China

ACM ISBN 979-8-4007-1302-6/24/11…$15.00
https://doi.org/10.1145/3698587.3701395

a convolutional autoencoder [24] to improve read count accuracy, grouping cells into subpopulations via a Gaussian mixture model (GMM) [7] and identifying CNAs with an HMM [12]. DeepCNA [19] transforms high-dimensional read counts into a lower-dimensional latent space, applies CBS to detect breakpoints, and uses a mixture model to estimate copy numbers. This approach reduces reconstruction error and enhances accuracy through the expectation-maximization (EM) algorithm [20]. While these techniques improve CNV detection, they require significant processing resources and careful hyperparameter tuning but represent progress in scDNA-seq.

We present a novel method combining self-supervised learning and graph theory to denoise scDNA-seq data and extract true biological signals. Tested on simulated and real datasets, it improves CNV detection accuracy by adapting to noise variations and distinguishing noise from signals with greater precision. The key contributions are:

- **Robustness to Noise**: The self-supervised learning architecture adapts to diverse noise distributions, ensuring flexibility across data formats and noise levels in scDNA-seq.
- **Fusion of advance learning method and graph theory**: Integrating deep learning with graph theory improves signal/noise distinction, enhancing clarity and CNV reliability.
- **Improved CNV Detection**: Enhanced denoising improves CNV identification, capturing subtle variations and providing crucial insights into genetic diversity and its implications.

2 METHOD AND DATASETS

Our methodology follows four steps (Figure 1) to improve accuracy and reliability. We start with data preprocessing to manage technical biases, followed by graph analysis to integrate information across cells. Next, we apply denoising to reduce noise while preserving biological signals, and finally, use circular binary segmentation (CBS) to detect CNVs. The read count data is generated using a sliding window approach, segmenting the genome into smaller units for analysis.

2.1 Data Preprocessing

Accurate CNV detection in scDNA-seq data requires minimizing biases and noise. Preprocessing focuses on GC content and read mappability. High or low GC content can cause uneven coverage, affecting CNV detection accuracy. We filter out bins with GC content below 0.3 or above 0.7. For mappability bias, we remove bins with a mappability score below 0.9, ensuring only regions with reliable mapping are included.

2.2 Graph Analysis

We considered a set of cell read counts, each represented by a vector $\mathbf{x}_i \in \mathbb{R}^N$, where N is the number of genomic bins. The set of all cell read count vectors is denoted as $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$.

The cosine similarity between two cell vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is computed as:

$$Sim(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}, \tag{1}$$

where $\mathbf{x}_i \cdot \mathbf{x}_j$ is their dot product, and $\|\mathbf{x}_i\|$ and $\|\mathbf{x}_j\|$ are their magnitudes.

We construct the k-nearest neighbors [11] (k-NN) graph for each vector based on cosine similarity. The value of K is adaptively chosen based on the average distance to the nearest neighbors. Specifically, K is set as the smallest value such that the distance to the K-th nearest neighbor exceeds the average neighbor distance. We cap K at a maximum value of 10 to maintain a balance between capturing local details and excluding distant, noisy neighbors. This dynamic adjustment of K ensures the k-NN graph accurately represents the underlying data structure.

The affinity between vectors is computed using a Gaussian kernel:

$$A_{ij} = \exp\left(-\frac{Sim(\mathbf{x}_i, \mathbf{x}_j)^2}{2\sigma_i^2}\right), \tag{2}$$

where σ controls how quickly the affinity diminishes with distance. We determine σ adaptively for each data point by calculating the cosine distance to its adaptively determined K-th nearest neighbor. This ensures that the affinity matrix accurately reflects the local structure of the data and enhances the effectiveness of the diffusion process by tuning the Gaussian kernel to each data point's characteristics.

The degree matrix D and the normalized graph Laplacian L_{norm} are defined as:

$$D_{ii} = \sum_{j=1}^{n} A_{ij}, \tag{3}$$

$$L_{\text{norm}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}. \tag{4}$$

2.3 Denoising Technique

The diffusion process [10], inspired by the heat equation [3], simulates the propagation of biological signals across cell networks. DNA short read counts typically show gradual transitions across datasets, and abrupt shifts often indicate noise or errors.

In our method, we apply multiple diffusion scales (e.g., 1, 5, 10) rather than a single time step τ, modulating diffusion intensity to balance signal retention and noise reduction. For each batch X_b, the diffusion process is:

$$\mathbf{f}_b^{(m+1)} = e^{-\tau L_{\text{norm},b}} \mathbf{f}_b^{(m)}, \tag{5}$$

where $\mathbf{f}_b^{(m)}$ represents the denoised state at iteration m. We use multiple scales to iteratively enhance denoising, preserving biological signals while minimizing noise.

The diffusion operator $e^{-\tau L_{\text{norm}}}$, based on the normalized graph Laplacian L_{norm}, captures local connections between data points (cells), preserving the dataset's geometry during the diffusion process.

The exponential of the graph Laplacian $e^{-tL_{\text{norm}}}$ is computed using the matrix exponential:

$$e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k, \tag{6}$$

where A represents the product of $-t$ and L_{norm}. This series expansion converts continuous-time diffusion into a computable discrete form, modeling the cumulative pathways in the graph.

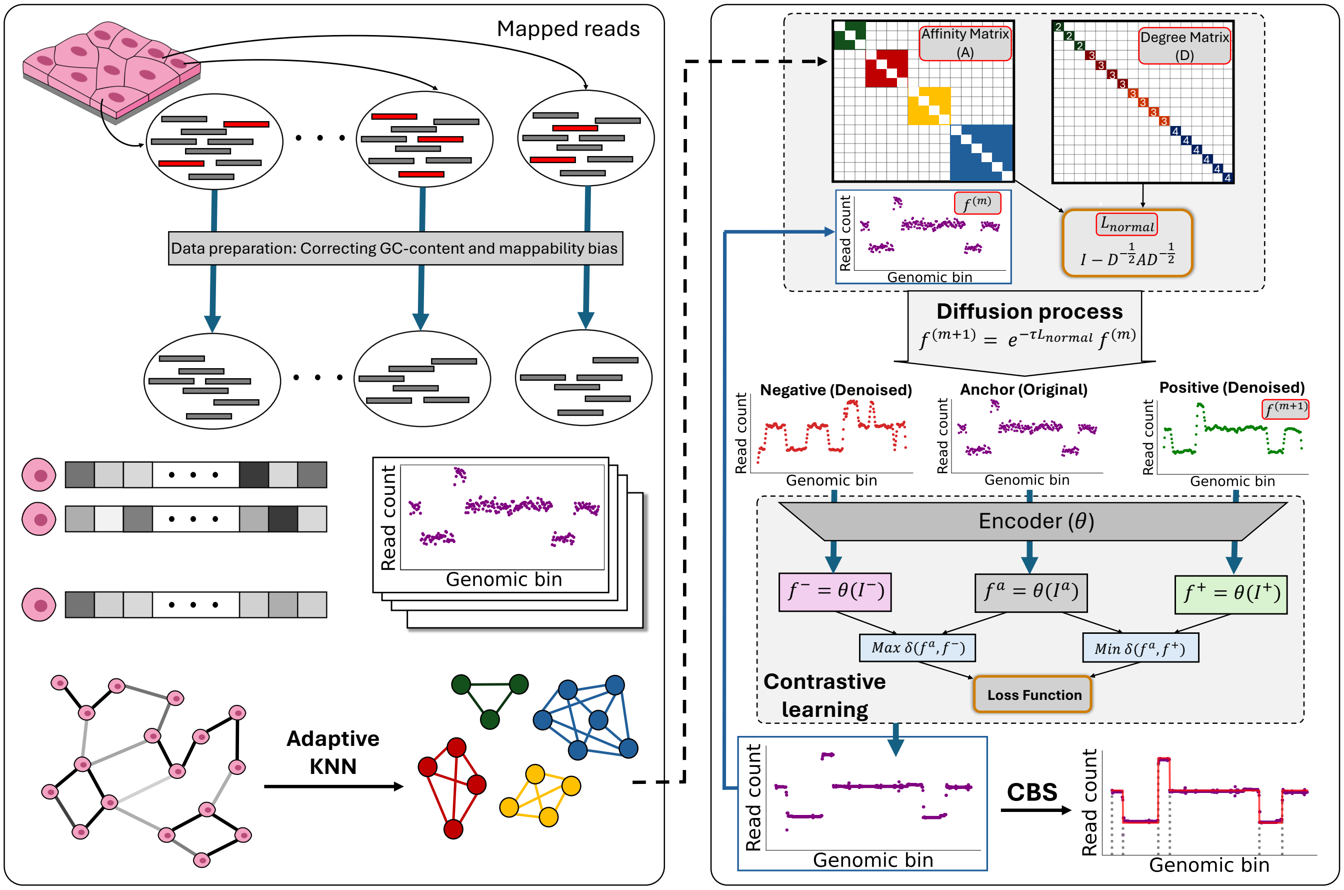

Figure 1: The DCCNV workflow includes: I) Data preparation (read count extraction, GC content/mappability correction, normalization). II) Initial denoising with cosine similarity and adaptive k-NN graph construction. III) Signal enhancement through the diffusion process, followed by contrastive learning. IV) Final signal segmentation using CBS to detect CNV locations.

The diffusion process enhances scDNA-seq data by merging each cell's read counts with its neighbors, reducing random fluctuations while preserving biological patterns. This approach efficiently addresses the variability and noise in scDNA-seq data.

Batch processing is primarily used to improve computational efficiency when handling large datasets. Batches are randomly sampled for contrastive learning, and while the k-NN graph is crucial for the diffusion process, the formation of batches is independent of the graph. This ensures efficient processing without targeting specific local structures.

Dividing the dataset X into subsets $X_1, X_2, \ldots, X_B$ allows independent batch processing, given these advantages:

- **Computational Feasibility:** Denoising becomes manageable for large datasets, reducing memory and resource demands.
- **Focused Diffusion:** Batch processing allows diffusion to be adjusted to the specific attributes of each subset, particularly useful for scDNA-seq data, where cell subpopulations may exhibit distinct copy number patterns and noise levels.
- **Noise Variability:** The diffusion process can be calibrated separately for each batch, adjusting parameters like σ or τ to account for varying technical noise.

The iteration process within each batch continues until convergence, indicated by signal stability or minimal change. This prevents excessive alteration of the biological signal, crucial for accurate analysis.

Our contrastive learning framework uses a contrastive loss function to enhance the quality of the denoised signal. Given an original vector $\mathbf{x}_i$, its denoised counterpart $\mathbf{y}_i$, and a negative sample $\mathbf{y}_j$, the loss is:

$$L(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y}_j) = \max\big(d(f(\mathbf{x}_i, \theta), f(\mathbf{y}_i, \theta)) \\ - d(f(\mathbf{x}_i, \theta), f(\mathbf{y}_j, \theta)) + \gamma, 0\big), \tag{7}$$

here, d denotes distance in the embedding space, θ the parameters of the embedding function f, and γ the margin for separating positive (original and denoised) from negative pairs. γ is empirically chosen, and negative samples are selected using hard negative sampling based on cosine similarity to challenge the model.

The objective is to minimize this contrastive loss:

$$\theta^* = \arg\min_{\theta} \sum_{b=1}^{B} \sum_{i,j} L(\mathbf{x}_i, \mathbf{y}_i^b, \mathbf{y}_j^b). \tag{8}$$

This aligns original and denoised pairs while distancing them from unrelated negative samples, improving signal reliability for downstream analysis. Regularization terms prevent overfitting and are fine-tuned on validation sets to balance model complexity with generalization. The denoising algorithm is detailed in Algorithm 1, provided in the Supplementary Material.

2.4 Segmentation

After denoising, CBS [22] is used for CNV detection and segmentation. Denoising enhances CBS input by reducing noise and improving signal integrity, essential for accurate genomic segmentation.

CBS divides genomic bins into segments with consistent copy numbers, solving the change-point problem statistically. For each change-point i and j, the statistic Z_{ij} is calculated:

$$Z_{ij} = \sqrt{\frac{j-i}{n-j+i}} \cdot \left(\frac{S_j - S_i}{j-i} - \frac{S_n - S_j + S_i}{n-j+i} \right), \tag{9}$$

where n is the number of bins, S_k the cumulative sum of log ratios up to marker k, and Z_{ij} measures the mean difference between segments at i and j. The maximum $|Z_{ij}|$ identifies significant change-points.

Once a change-point is found, the chromosome is segmented, and the process repeats recursively. This ensures thorough exploration of genomic structure.

CBS applied to our denoised data improves CNV detection by enhancing signal clarity. To assign integer copy numbers to segments, read counts are normalized based on coverage (C), read length (L_{read}), and region length (L_{region}):

$$\text{Expected Read Count} = \left(\frac{C \times L_{\text{region}}}{L_{\text{read}}} \right) \times 2. \tag{10}$$

Normalized read counts near 1 correspond to diploid (copy number 2), with higher values indicating duplications.

2.5 Dataset

We evaluated our method on simulated and real scDNA-seq data. For the simulated data, we used CNAsim [34] that generates accurate copy number profiles for simulated tumor cells, covering a range of CNA pathways and addressing scDNA-seq biases. It includes whole-genome duplications, whole-chromosomal CNAs, chromosome-arm CNAs, and subclonal population structures with normal diploid and pseudo-diploid cells.

The real data used SNS [21] to study tumor populations. Two breast cancer scDNA-seq datasets were analyzed: T10, featuring a tumor with multiple genetic abnormalities, and T16, involving a primary tumor and its liver metastasis. In T10, 100 cells were sequenced, revealing three clonal subpopulations. T16 involved 100 cells sequenced from the primary tumor and its metastasis.

3 EXPERIMENTS AND RESULTS

3.1 Evaluation Metrics

To evaluate DCCNV, we use breakpoints to compare the segmented denoised read count signals with the ground truth in simulated data. The primary metric is the mean absolute differences (MAD) [14] between the actual and estimated copy numbers for each cell across shared bins.

Let $\mathbf{C}_{\text{true}}$ and $\mathbf{C}_{\text{est}}$ represent the matrices of true and estimated copy numbers (size $N \times M$, where N is the number of cells and M is the number of genomic bins). The MAD for cell i is:

$$\text{MAD}_i = \frac{1}{M} \sum_{j=1}^{M} \left| C_{\text{true},ij} - C_{\text{est},ij} \right|. \tag{11}$$

The overall MAD is the mean across all cells:

$$\text{MAD}_{\text{mean}} = \frac{1}{N} \sum_{i=1}^{N} \text{MAD}_i. \tag{12}$$

Smaller MAD values indicate better alignment with the ground truth, reflecting higher accuracy. We also use sensitivity, precision, and F1-score to further assess the breakpoint detection performance.

3.2 Baseline Models

We compared DCCNV with several baseline methods, including DeepCNA [19], rcCAE [36], SCOPE [33], SCONE [14], HMMCopy [27], and SeCNV [26], as well as traditional filter-based methods like Wavelet [1], Median [31], and Gaussian [28]. To ensure fairness, the same segmentation and pipeline procedures were applied across all methods.

3.3 Analyzing Simulated Data

We used simulated data to evaluate the model's performance in identifying breakpoints and CNVs. After preprocessing, 2955 out of 3450 bins remained. The model was trained with a latent dimension of $d = 128$, over 500 epochs, using a learning rate of 0.001 and a batch size of 32 (Supplementary Table 4 shows the architecture). The four simulated datasets (A, B, C, and D) are described in Table 1, each varying in the number of cells, bin length, and other characteristics. MAD results for CNV detection across the four simulated datasets are shown in Figure 2. For visualization purposes, see Supplementary Figure 1.

In dataset A, DCCNV achieved the lowest MAD (0.10), indicating the highest accuracy, followed by DeepCNA (0.11) and rcCAE (0.12). Filter-based methods like Gaussian (0.37), Wavelet (0.33), and Median (0.35) had higher MAD values.

For dataset B, DCCNV maintained high performance with a MAD of 0.12, while DeepCNA and rcCAE had MADs of 0.13 and 0.15. Filter-based methods continued to perform worse, with the Gaussian method reaching up to 0.42.

In dataset C, DCCNV achieved a MAD of 0.15, followed by DeepCNA (0.16) and rcCAE (0.18). SCOPE and SCONE performed better than filter-based methods but were outperformed by DCCNV.

In dataset D, DCCNV had a MAD of 0.17, with DeepCNA and rcCAE close behind at 0.18 and 0.20. HMMCopy showed comparable performance but was outperformed by DCCNV. Filter-based

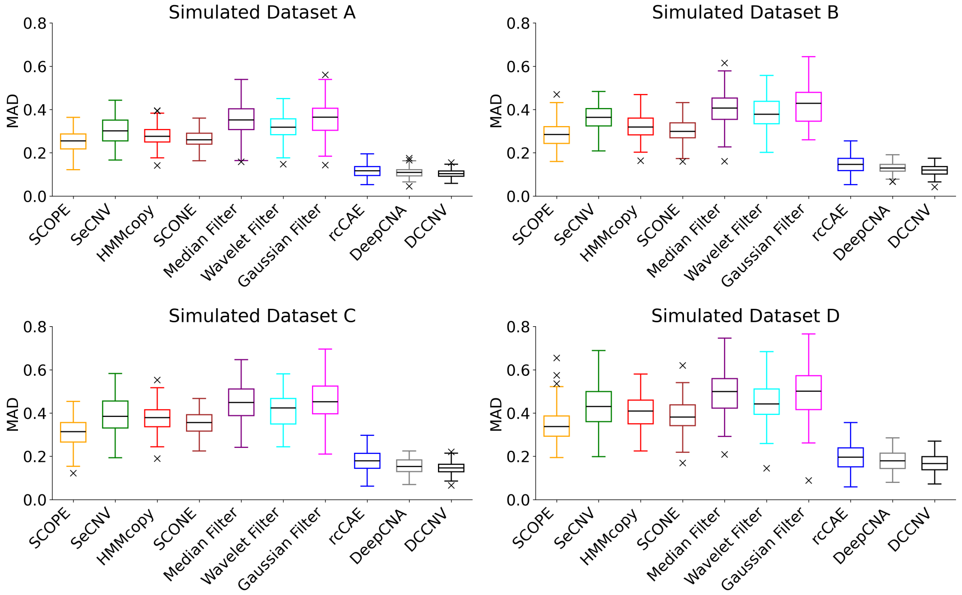

Figure 2: Copy number estimation accuracy of the proposed methods on simulated datasets. We compute the MAD between the estimated and real copy numbers for every cell.

Table 1: Summary of Simulated Datasets

Dataset	# Cells	Bin Length	# Clones	Mean CN Length	Min CN Length	Chromosomal Events	Coverage	Error Rate
A	100	1,000,000	3	5,000,000	10,000	Chromosomal level	0.1	0.02
B	500	5,000,000	7	5,000,000	10,000	WGD present	0.05	0.02
C	1,000	5,000,000	12	5,000,000	10,000	High rate of arm-level	0.05	0.02
D	10,000	5,000,000	12	5,000,000	1,000	Arm-level changes	0.02	0.02

methods had the highest MAD values, reinforcing the superiority of CNV detection tools.

DCCNV consistently achieved the lowest MAD values, demonstrating superior performance. The combination of diffusion processes and contrastive learning significantly enhances its ability to detect CNVs accurately.

Table 3 shows the scalability and efficiency of DCCNV, made possible by batching techniques. While filter-based methods are faster, they lack the precision and reliability of more advanced approaches like DCCNV for accurate CNV identification.

3.4 Analyzing Real Data

We evaluated the efficacy of DCCNV and baseline models using the T10 and T16 datasets, with CHISEL [37] serving as the reference point, as ground truth breakpoints and copy numbers are unavailable.

Table 2 compares the accuracy in terms of sensitivity, specificity, and F1 score. For T10, DCCNV achieved a sensitivity of 0.92, a specificity of 0.94, and an F1 score of 0.93, outperforming all baselines. DeepCNA and rcCAE followed with F1 scores of 0.91 and 0.89, respectively, while traditional filter-based methods showed lower performance.

In T16, DCCNV maintained strong performance (sensitivity: 0.90, specificity: 0.91, F1: 0.89). DeepCNA showed similar results, and rcCAE had slightly lower performance (specificity: 0.90, F1: 0.88).

A detailed ablation study assessing the contributions of key components, including the diffusion process and contrastive learning, is provided in the Supplementary Material. It shows that both the diffusion process and contrastive learning are essential for achieving high accuracy in CNV detection. As shown in Supplementary Material Tables 1, 2, and 3, removing either component results in a notable decrease in the F1 score, demonstrating the importance of each in preserving signal clarity and minimizing noise.

4 CONCLUSION

In this study, we introduced DCCNV, a novel pipeline for detecting CNVs in scDNA-seq data, integrating diffusion processes and contrastive learning. Extensive experiments on both simulated and real datasets showed DCCNV's superior performance over existing methods, including DeepCNA, rcCAE, SCOPE, SCONE, HMMcopy, SeCNV, and traditional filter-based pipelines.

DCCNV consistently achieves the lowest MAD values in simulated data, indicating high accuracy and robustness. In real datasets, using CHISEL's results as the ground truth, DCCNV also showed higher sensitivity, specificity, and F1 scores, further validating its effectiveness.

Table 2: Comparison of Breakpoint Detection Accuracy for T10 and T16 Datasets (Sensitivity, Specificity, F1 score)

Method	T10 Dataset			T16 Dataset		
	Sens.	Speci.	F1	Sens.	Speci.	F1
DCCNV	**0.92**	**0.94**	**0.93**	**0.90**	0.91	0.89
DeepCNA	0.90	0.92	0.91	0.90	**0.93**	**0.90**
rcCAE	0.88	0.91	0.89	0.87	0.90	0.88
SCOPE	0.85	0.88	0.86	0.84	0.87	0.85
SCONE	0.83	0.86	0.84	0.82	0.85	0.83
HMMcopy	0.80	0.84	0.82	0.79	0.83	0.81
SeCNV	0.78	0.82	0.80	0.77	0.81	0.79
Wavelet-based	0.75	0.80	0.77	0.74	0.79	0.76
Median-based	0.73	0.78	0.75	0.72	0.77	0.74
Gaussian-based	0.70	0.76	0.73	0.69	0.75	0.72

Table 3: Runtime performance of all methods on simulated datasets with different cell counts, run on a server with 48 CPU cores, 128 GB RAM, and 1 NVIDIA TESLA M10 GPU.

Method	100 cells	500 cells	1000 cells	10000 cells
DCCNV	53	73	94	288
SCOPE	3813	8245	14429	28750
HMMcopy	367	449	562	1251
SCONE	149	255	343	832
SeCNV	1408	1947	2474	5194
rcCAE	69	116	177	491
DeepCNA	65	95	124	353
Wavelet-based	49	53	61	154
Median-based	42	50	53	147
Gaussian-based	43	45	59	138

While promising, DCCNV could be further enhanced by improving noise reduction and normalization algorithms. Currently, GC content and mappability are addressed, but incorporating advanced methods like batch correction [32] and improved normalization for DNA sequencing data could reduce biases and improve CNV precision. Refining the graph creation process by using alternative metrics like Pearson correlation [29] or mutual information [30], along with adaptive clustering methods [16], could better capture cell relationships and enhance diffusion. Also, incorporating models like GANs [13] for synthetic data generation or reinforcement learning [15] for hyperparameter optimization could further improve DCCNV's performance and CNV detection precision.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation (NSF) under grant No. 1942303, PI: Nabavi.

REFERENCES

[1] Kaushalya C Amarasinghe et al. 2014. Inferring copy number and genotype in tumour exome data. *BMC Genomics* 15 (2014), 1–12.

[2] Leonard E Baum et al. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.* 41 (1970), 164–171.

[3] John R Cannon. 1984. *The one-dimensional heat equation.* Cambridge Univ. Press.

[4] Scott L Carter et al. 2012. Absolute quantification of somatic DNA alterations in human cancer. *Nat. Biotechnol.* 30 (2012), 413–421.

[5] Simone Ciccolella et al. 2019. Effective clustering for single-cell sequencing cancer data. In *Proc. 10th ACM Int. Conf. Bioinformatics, Comput. Biol. Health Informatics.* 437–446.

[6] Ingrid Daubechies. 1988. Orthonormal bases of compactly supported wavelets. *Commun. Pure Appl. Math.* 41 (1988), 909–996.

[7] Arthur P Dempster et al. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* 39 (1977), 1–22.

[8] James Eberwine et al. 2014. The promise of single-cell sequencing. *Nat. Methods* 11 (2014), 25–27.

[9] James Eberwine and Junhyong Kim. 2015. Cellular deconstruction: finding meaning in individual cell variation. *Trends Cell Biol.* 25 (2015), 569–578.

[10] William Feller. 1951. Diffusion processes in genetics. In *Proc. 2nd Berkeley Symp. Math. Stat. Probab.*, Vol. 2. 227–247.

[11] Evelyn Fix. 1985. *Discriminatory analysis: nonparametric discrimination, consistency properties.* USAF Sch. Aviat. Med.

[12] Jane Fridlyand et al. 2004. Hidden Markov models approach to the analysis of array CGH data. *J. Multivar. Anal.* 90 (2004), 132–153.

[13] Ian Goodfellow et al. 2020. Generative adversarial networks. *Commun. ACM* 63 (2020), 139–144.

[14] Sandra Hui and Rasmus Nielsen. 2022. SCONCE: a method for profiling copy number alterations in cancer evolution using single-cell whole genome sequencing. *Bioinformatics* 38 (2022), 1801–1808.

[15] Leslie P Kaelbling et al. 1996. Reinforcement learning: A survey. *J. Artif. Intell. Res.* 4 (1996), 237–285.

[16] Hans-Peter Kriegel et al. 2011. Density-based clustering. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 1 (2011), 231–240.

[17] Yalan Lei, Rong Tang, et al. 2021. Applications of single-cell sequencing in cancer research: progress and perspectives. *J. Hematol. Oncol.* 14 (2021), 91.

[18] Jilong Li, Jie Hou, et al. 2015. From gigabyte to kilobyte: A protocol for mining large RNA-seq data. *PLoS One* 10 (2015), e0125000.

[19] Furui Liu et al. 2024. Inferring single-cell copy number profiles through cross-cell segmentation of read counts. *BMC Genomics* 25 (2024), 25.

[20] Xiao-Li Meng and David Van Dyk. 1997. The EM algorithm—an old folk-song sung to a fast new tune. *J. R. Stat. Soc. Ser. B* 59 (1997), 511–567.

[21] Nicholas Navin et al. 2011. Tumour evolution inferred by single-cell sequencing. *Nature* 472 (2011), 90–94.

[22] Adam B Olshen et al. 2004. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics* 5 (2004), 557–572.

[23] Ondrej Pös et al. 2021. DNA copy number variation: Main characteristics, evolutionary significance, and pathological aspects. *Biomed. J.* 44 (2021), 548–559.

[24] Marc'Aurelio Ranzato et al. 2007. Sparse feature learning for deep belief networks. *Adv. Neural Inf. Process. Syst.* 20 (2007).

[25] Richard Redon et al. 2006. Global variation in copy number in the human genome. *Nature* 444 (2006), 444–454.

[26] Wang Ruohan et al. 2022. Resolving single-cell copy number profiling for large datasets. *Brief. Bioinform.* 23 (2022), bbac264.

[27] Sohrab P Shah et al. 2006. Integrating copy number polymorphisms into array CGH analysis using a robust HMM. *Bioinformatics* 22 (2006), e431–e439.

[28] Claude E Shannon. 1948. A mathematical theory of communication. *Bell Syst. Tech. J.* 27 (1948), 379–423.

[29] Stephen M Stigler. 1989. Francis Galton's account of the invention of correlation. *Stat. Sci.* (1989), 73–79.

[30] M Thomas and A Joy. 2006. *Elements of information theory.* Wiley.

[31] John W Tukey. 1977. *Exploratory data analysis.* Springer.

[32] Po-Yuan Tung et al. 2017. Batch effects and the effective design of single-cell gene expression studies. *Sci. Rep.* 7 (2017), 39921.

[33] Rujin Wang et al. 2020. SCOPE: a normalization and copy-number estimation method for single-cell DNA sequencing. *Cell Syst.* 10 (2020), 445–452.

[34] Samson Weiner and Mukul S Bansal. 2023. CNAsim: improved simulation of single-cell copy number profiles and DNA-seq data from tumors. *Bioinformatics* 39 (2023), btad434.

[35] Chao Xie and Martti T Tammi. 2009. CNV-seq, a new method to detect copy number variation using high-throughput sequencing. *BMC Bioinform.* 10 (2009), 1–9.

[36] Zhenhua Yu et al. 2023. rcCAE: a convolutional autoencoder method for detecting intra-tumor heterogeneity and single-cell copy number alterations. *Brief. Bioinform.* 24 (2023), bbad108.

[37] Simone Zaccaria and Benjamin J Raphael. 2021. Characterizing allele- and haplotype-specific copy numbers in single cells with CHISEL. *Nat. Biotechnol.* 39 (2021), 207–214.

[38] Fatima Zare et al. 2017. Bias and Noise Cancellation for Robust Copy Number Variation Detection. In *Proc. 8th ACM Int. Conf. Bioinformatics, Comput. Biol. Health Informatics.* 591–591.