Briefings in Bioinformatics, 2026, 27, bbag163
https://doi.org/10.1093/bib/bbag163
Published: 14 April 2026
Problem Solving Protocol

OXFORD

GDSim: accurate simulation for single-cell transcriptomes based on the guided diffusion model

Tao Wang[1,2,*], Heyan Dong[1,2,‡], Hui Zhao[3,‡], Peimeng Zhen[1,2], Yongtian Wang[1,2], Xuequn Shang[1,2], Jiajie Peng[1,2], Bing Xiao[3,*], Jing Chen[4,5,*]

[1]School of Computer Science, Northwestern Polytechnical University, 1 Dongxiang Road, 710072 Xi'an, Shaanxi, China
[2]Key Laboratory of Big Data Storage and Management, Ministry of Industry and Information Technology, Northwestern Polytechnical University, 1 Dongxiang Road, 710072 Xi'an, Shaanxi, China
[3]School of Automation, Northwestern Polytechnical University, 1 Dongxiang Road, 710072 Xi'an, Shaanxi, China
[4]School of Automation (School of Artificial Intelligence), Beijing Information Science and Technology University, No. 55 Taihang Road, Changping District, 102206, Beijing, China
[5]School of Computer Science and Engineering, Xi'an University of Technology, No. 5 South Jinhua Road, 710048 Xi'an, Shaanxi, China

*Corresponding authors. Tao Wang, E-mail: twang@nwpu.edu.cn; Bing Xiao, E-mail: xiaobing@nwpu.edu.cn; Jing Chen, E-mail: chen-jing@xaut.edu.cn.
‡Heyan Dong and Hui Zhao contributed equally to this work.

Abstract

The advent of single-cell RNA sequencing (scRNA-seq) has transformed our ability to explore cellular heterogeneity and developmental processes at the single-cell level. Despite its transformative potential, challenges such as technical limitations, high costs, and sample scarcity can lead to insufficient scRNA-seq data, limiting its effectiveness in downstream analysis. In particular, there is often a lack of baseline data or an inadequate number of training samples for building robust computational models. To address these issues, we present GDSim, a novel deep generative network for the simulation of scRNA-seq data. GDSim leverages a label-guided diffusion-based model to capture the complex gene expression dependencies within scRNA-seq data, generating simulated datasets that closely reflect the true distribution of the original data. Experimental evaluations demonstrate that GDSim achieves superior performance in recovering data distribution characteristics compared with state-of-the-art methods. Moreover, GDSim maintains high consistency with real data in cell subtype clustering and differential gene expression analysis, offering a powerful tool for scRNA-seq simulation and downstream biological applications.

Keywords single-cell RNA-seq, simulation, diffusion model, deep learning

Introduction

Recent advances in single-cell RNA sequencing (scRNA-seq) technologies have revolutionized the study of cellular heterogeneity by enabling the profiling of gene expression at the resolution of individual cells [1, 2]. These high-resolution data have uncovered new cell types, revealed dynamic processes such as differentiation and development, and provided insights into disease states, including cancer progression and immune responses [3–5]. Unlike bulk RNA sequencing, which averages gene expression across large cell populations, scRNA-seq captures the complexity and diversity of cellular states within heterogeneous tissues, making it a powerful tool for understanding biological processes at an unprecedented scale [6, 7].

Despite the promise of scRNA-seq, analyzing such data presents significant computational challenges [8–10]. The high-dimensional nature of scRNA-seq data, combined with its inherent sparsity and noise, complicates tasks, such as clustering, trajectory inference, and differential expression analysis [10, 11]. Developing reliable computational methods to address these challenges requires comprehensive benchmarking, which is difficult to achieve using experimental data alone [12]. Real biological datasets often lack a ground truth, vary widely in quality, and are influenced by technical artifacts such as batch effects or sequencing depth [13–15]. These factors limit the ability to systematically evaluate and compare computational tools.

To overcome these challenges, there is a growing need for scRNA-seq simulation frameworks that can generate biologically realistic datasets under controlled conditions. Simulated datasets provide several advantages. First, they enable the benchmarking of computational methods by offering datasets with known ground truth, allowing for objective performance evaluation [16–18]. Second, simulations

Received: October 16, 2025. **Revised:** February 15, 2026. **Accepted:** March 17, 2026

allow researchers to explore rare or transient cellular states that are difficult to capture experimentally [19–21]. Finally, simulated data play a crucial role in experimental design by helping researchers optimize parameters such as sample size and sequencing depth, ultimately reducing the cost and complexity of experiments [22–24].

Numerous methods for simulating scRNA-seq data have been developed, which can be broadly categorized based on their underlying modeling assumptions: parametric and non-parametric approaches [25, 26]. Parametric methods typically rely on predefined statistical distributions, such as negative binomial [27, 28] or zero-inflated negative binomial distributions [29, 30], to capture technical noise and sparsity. While more flexible hybrid models like gamma-normal [31] and beta-Poisson mixtures models [32] have been proposed, these parametric frameworks still face challenges in accurately modeling the intricate, high-dimensional distribution characteristics of scRNA-seq data due to their rigid distributional assumptions. In contrast, non-parametric modeling avoids such constraints by learning the data distribution directly from observed samples without assuming a fixed functional form. As a sophisticated implementation of this non-parametric philosophy, deep generative models leverage the universal approximation power of neural networks to implicitly represent complex gene expression manifolds [33]. For instance, cscGAN [34] employs adversarial training to capture gene regulation patterns across cell types. However, as an implementation choice, these deep generative architectures often require large-scale datasets for stable training and may struggle with imbalanced data, potentially leading to inaccurate simulations for rare cell populations.

Despite these advancements, there remains a need for a simulation framework that can accurately capture the complex variability of scRNA-seq data, while offering flexibility in terms of user-defined cell type configurations. To this end, we propose GDSim, a deep generative network based on diffusion theory, for simulating scRNA-seq data guided by cell types. GDSim operates by gradually introducing noise into scRNA-seq data until it becomes pure noisy data, after which a deep generative network predicts and removes the added noise to recover biologically realistic gene expression profiles. This noise-to-data formulation provides an intuitive and principled mechanism for learning the complex distribution of single-cell transcriptomes. By transforming the generative task into a sequence of conditional denoising problems, GDSim avoids directly modeling the highly irregular and high-dimensional data distribution of scRNA-seq data. Instead, the model learns how biologically meaningful structure is progressively reconstructed from increasingly corrupted representations. At early stages of denoising, GDSim captures coarse-grained global characteristics such as major cell-type separations, while later stages refine fine-scale gene–gene dependencies and subtle transcriptional variations. This multi-scale learning process enables GDSim to faithfully approximate the underlying biological manifold and generate realistic single-cell expression profiles with high stability. Similar diffusion-based generative paradigms have demonstrated remarkable effectiveness in other domains characterized by complex, high-dimensional structured data, including image synthesis [35, 36] and protein structure modeling [37, 38], further supporting the suitability of diffusion theory for modeling the intricate variability of scRNA-seq data. Moreover, GDSim incorporates user-defined cell type annotations, enabling researchers to precisely control the composition and abundance of simulated cell populations. This flexibility facilitates the generation of tailored datasets for benchmarking rare cell types, evaluating clustering algorithms, or optimizing experimental designs.

We conducted a systematic evaluation of GDSim against seven state-of-the-art methods using multiple real scRNA-seq datasets and comprehensive evaluation metrics. Our results demonstrate that GDSim not only faithfully recapitulates the complex distribution patterns of real scRNA-seq data but also preserves critical biological features with remarkable accuracy (ACC). Compared with existing approaches, GDSim shows superior performance in maintaining the intrinsic clustering relationships between cell types and reproducing authentic differential expression signatures. This high-fidelity simulation capability positions GDSim as an invaluable tool for both single-cell transcriptomics research and rigorous benchmarking of computational methods.

Materials and methods

scRNA-seq preprocessing

GDSim employs a rigorous preprocessing pipeline to ensure high-quality input data for the diffusion model while maximizing the preservation of biological heterogeneity. A critical challenge in single-cell analysis is that unsupervised filtering—relying solely on global expression frequency—often inadvertently removes specific marker genes essential for identifying low-abundance cell populations. To address this, we developed a cell-type-specific feature selection (CSFS) strategy to safeguard these vital biological signals. Specifically, when cell-type information is available, we utilize the Wilcoxon rank-sum test to perform differential expression analysis across all annotated cell types in a one-versus-rest manner. This process identifies genes that are significantly up-regulated within specific clusters, including ultra-rare populations comprising <1% of the total dataset. These identified marker genes are explicitly added to a "whitelist" and are forcibly retained regardless of their global detection frequency. Standard filtering criteria, which remove genes detected in fewer than 1% of cells, are only applied to the remaining feature set after these cell-type-specific signatures have been secured. The retained gene expression values are then converted to counts per million (CPM) and log-transformed to normalize technical variations across samples. Finally, we apply standardization to achieve zero mean and unit variance for each gene, facilitating stable model training. For efficient processing by the neural network, we reshape the preprocessed expression matrices into a structured 2D format $x \in \mathbb{R}^{1\times m\times n}$, where $m \times n$ matches the total number of genes after zero-padding for dimensional consistency. This transformation preserves the high-dimensional relationships between genes while optimizing computational efficiency. The framework supports both supervised and unsupervised learning paradigms. When cell types are available, we encode them numerically as $l_i \in \{0, 1, 2, \ldots, n\}$, enabling cell-type-conditioned generation during the diffusion process for precise simulation of specific populations. In the absence of such information, GDSim automatically learns the underlying data distribution to generate realistic profiles. The model dynamically adapts to either scenario, maintaining robust performance regardless of annotation availability.

Forward diffusion process

The full architecture of GDSim is provided in Supplementary Fig. S1. In the forward diffusion process, GDSim systematically adds noise to the data at each time step, transforming the original data x_0 into a series of progressively noisier data points $\{x_1, x_2, \ldots, x_T\}$. This process follows a

Markov chain, with the transition probability defined as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I), \tag{1}$$

where $\mathcal{N}$ represents the normal distribution, and β_t is the variance coefficient that controls the amount of noise added at each step. To ensure efficient diffusion and smooth noise scheduling, GDSim employs a cosine annealing schedule to define the variance coefficients β_t.The definition of β_t can be found in the Supplementary file. As t increases, β_t grows larger, causing x_t to approach pure noise. This schedule ensures a non-linear but consistent increase in noise over time, enhancing the model's ability to retain structural information while simulating realistic scRNA-seq data.

The relationship between x_t and the initial data x_0 is described by:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I), \tag{2}$$

where $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$. Here, $\bar{\alpha}_t$ represents the cumulative product of the noise reduction factors over time, ensuring that the noise addition process is gradual and controlled. Based on the above equation, we only need the x_0 to derive x_t at any time.

Reverse sampling process

The reverse sampling process in GDSim reconstructs the original data x_0 from the noisy input x_T by leveraging an UNet-based neural network to predict the mean and variance of the reverse sampling process distribution. Specifically, the reverse distribution is defined as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{3}$$

where $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ represent the predicted mean and variance of the reverse process at time step t, as determined by the network, and θ represents the network parameters. This prediction is critical for accurately reversing the noise added during the forward diffusion process and recovering meaningful gene expression data.

Once the network is trained, it is employed in the reverse sampling process to reconstruct the gene expression matrix from the noisy data x_T. This process iteratively refines the noisy input back into a biologically meaningful state. If cell labels were incorporated during training, the reverse process generates gene expression data corresponding to the specified cell types. Otherwise, the output reflects an unsupervised simulation.

Deep neural network architecture and optimization

Network architecture

At the core of GDSim lies a modified UNet architecture designed to predict both the mean and variance parameters for the reverse diffusion process. The network employs a symmetric encoder–decoder structure with four hierarchical levels, connected through skip connections to preserve spatial information across scales. The encoder pathway consists of four convolutional blocks, where each block (except the first) performs 3× downsampling followed by four residual modules with Swish activation. Similarly, the decoder contains four upsampling blocks, with each block (excluding the last) employing transposed convolution for 3× upsampling before residual processing. Between corresponding encoder and decoder levels, skip connections facilitate gradient flow and feature reuse.

To capture long-range dependencies in the gene expression data, we implement a multi-head attention mechanism (four heads) at three intermediate resolution levels (layers 2–4). This global attention module operates on flattened feature maps, enabling the network to model relationships between distant genomic features. The attention layers are particularly crucial for maintaining coherent cell-type-specific patterns during the reverse diffusion process.

Embedding layer

The model processes time steps and cell type labels through separate embedding pathways that are subsequently merged.

Time step encoding

We employ sinusoidal position embedding to convert continuous time steps into vector representations:

$$\mathrm{E}_k = t \cdot \exp\left(-\frac{k\log(10^4)}{d/2-1}\right), \quad \text{for } k = 0, \ldots, \lfloor d/2 \rfloor - 1 \tag{4}$$

$$\mathrm{h}(t) = \left[\sin(E_0), \cos(E_0), \ldots, \sin(E_{\lfloor d/2 \rfloor - 1}), \cos(E_{\lfloor d/2 \rfloor - 1})\right] \in \mathbb{R}^d \tag{5}$$

where t is the time step, d denotes the embedding dimension, and the square brackets represent concatenation along the feature dimension. This encoding scheme produces d-dimensional vectors that preserve temporal relationships through sinusoidal patterns with exponentially decreasing frequencies. Then apply two fully connected layers to further extract features of the time step size:

$$\mathrm{TimeEmbedding}(t) = W_{FC_2}\left(\sigma\left(W_{FC_1} \cdot h_{(t)}\right)\right) \tag{6}$$

Here, $h_(t)$ represents the latent representation of time step obtained from the sinusoidal position embedding, σ denotes the *SiLU* activation function, W_{FC_1} and W_{FC_2} are trainable parameters.

Label embedding

Cell type labels are embedded using a trainable lookup table:

$$\mathrm{LabelEmbedding}(y) = W_y \in \mathbb{R}^d \tag{7}$$

where $W_y \in \mathbb{R}^{C\times d}$ is an embedding matrix for C distinct cell types.

Feature fusion

The time and label embeddings are combined through element-wise summation:

$$\mathrm{Emb} = \mathrm{TimeEmbedding}(t) + \mathrm{LabelEmbedding}(y) \tag{8}$$

This fused representation serves as input to subsequent ResNet blocks in the architecture.

Complete architectural specifications, including channel dimensions at each hierarchical level and hyperparameter values, are provided in Supplementary Figs S2–S9.

Learning rate optimization

During the training process, we use a cosine annealing decay strategy with preheating to optimize the learning rate. This strategy enables the model to start with a small learning rate, increase it to a predefined value, and then decay it gradually according to a cosine function. Specifically, at the beginning of training, a smaller learning rate is chosen. After training for a certain number of rounds, it is modified to a pre-set learning rate for training. In all experiments, this value is

set to 1e-4. After the preheating stage, the learning rate will gradually decrease according to the cosine function, as shown in Equation (9):

$$l_r = \begin{cases} l_{\min} + (l_{\max} - l_{\min}) * 0.1/T_w, & T_{\text{cur}} = 0 \\ l_{\min} + (l_{\max} - l_{\min}) * T_{\text{cur}}/T_w, & 0 < T_{\text{cur}} \leq T_w \\ l_{\min} + 0.5 * (l_{\max} - l_{\min}) * (1 + \cos(\frac{T_{\text{cur}}}{T}\pi)), & T > T_w \end{cases} \quad (9)$$

Here, $l_{\min}$ and $l_{\max}$ represent the minimum and maximum learning rates during the preheating phase, set to 1×10^{-9} and 1×10^{-4}, respectively. T denotes the total number of training rounds, T_w indicates the preheating period, and T_{cur} represents the current training round. In all experiments, T_w is set to 5, allowing the learning rate to decay smoothly according to the cosine annealing strategy after the preheating phase.

Loss function and hyperparameters

The training objective of the neural network is to minimize the difference between the predicted noise and the actual noise applied in the forward diffusion process. This objective is formalized through a hybrid loss function:

$$\mathcal{L}_{\text{hybrid}} = \mathcal{L}_{\text{simple}} + \lambda \mathcal{L}_{\text{KL}} \quad (10)$$

where $\mathcal{L}_{\text{simple}}$ is the mean squared error between the predicted noise and the true noise:

$$\mathcal{L}_{\text{simple}} = E_{x_0, \tilde{z}_t}\left[\left\|\tilde{z}_t - z_\theta(x_t, t)\right\|^2\right] \quad (11)$$

In this expression, z_θ denotes the noise predicted by the network and z represents the actual noise added in the forward process. Consequently, L_{simple} functions as a reconstruction loss that directs the model to accurately recover the underlying gene expression patterns from the perturbed data. The second term, $\mathcal{L}_{\text{KL}}$, is the Kullback–Leibler (KL) divergence between the forward diffusion posterior $q(x_{t-1}|x_t, x_0)$ and the reverse process distribution $p_\theta(x_{t-1}|x_t)$:

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}\left(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_{x_t})\right) \quad (12)$$

The closed-form calculation of this divergence is defined as:

$$\mathcal{L}_{\text{KL}} = 0.5 * \left(-1 + \log(\sigma_2^2) - \log(\sigma_1^2) + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{\sigma_2^2}\right) \quad (13)$$

Here, μ_1, σ_1 and μ_2, σ_2 correspond to the mean and variance of the q and p_θ distributions, respectively. Physically, $\mathcal{L}_{\text{KL}}$ serves as a distributional regularizer. By minimizing the divergence between the forward and reverse Markov chains, GDSim ensures that the generated simulated data faithfully align with the complex, non-parametric distributions characteristic of real single-cell transcriptomes. The weighting factor λ, set to 0.1, balances these two components to achieve an optimal trade-off between noise reconstruction precision and the preservation of biological signals. It is essential to distinguish between model hyperparameters and internal parameters. These hyperparameters—including layer counts, attention heads, learning rate schedules, and loss weighting—are prespecified architectural and optimization configurations rather than variables updated via gradient descent during the training phase. While the optimization process focuses exclusively on learning the model's internal weights, these higher-level settings remain adjustable, providing the flexibility for users to adapt the GDSim architecture to specific biological contexts or data scales.

Dataset availability

We evaluated GDSim using three scRNA-seq datasets representing diverse sequencing platforms and biological systems, ensuring robust validation of our method's performance across different experimental conditions.

The human pluripotent stem cell (hPSC) dataset (GSE75748), generated using the Smart-seq2 platform, served as our primary benchmark for method evaluation [39]. This comprehensive dataset contains 1018 high-quality scRNA-seq cells representing seven distinct cellular states that span a spectrum of pluripotency and differentiation. The dataset includes differentiated lineages such as neuronal progenitor cells (NPC, $n = 173$), definitive endoderm cells (DEC, $n = 138$), endothelial cells (EC, $n = 105$), and trophoblast-like cells (TB, $n = 69$), along with undifferentiated controls consisting of H1 embryonic stem cells ($n = 212$), H9 embryonic stem cells ($n = 162$), and human foreskin fibroblasts (HFF, $n = 159$). After filtering genes detected in fewer than 1% of cells, we retained 16 531 genes for analysis.

For additional validation, we analyzed the human T lymphocyte line dataset (Jurkat dataset) from 10× Genomics [40], comprising 3258 cells. Unsupervised clustering revealed eight distinct subpopulations in this dataset. Following identical quality control procedures, we retained 11 550 genes for downstream analysis.

To assess platform-independent performance, we included mouse embryonic stem cells (mESCs dataset, GSE90047) [41] sequenced using the SMARTer platform. This developmental time-course contains 447 cells across seven stages, including E10.5 ($n = 54$), E11.5 ($n = 70$), E12.5 ($n = 41$), E13.5 ($n = 65$), E14.5 ($n = 70$), E15.5 ($n = 77$), and E17.5 ($n = 70$). The filtered expression matrix contained 19 862 genes after applying the same 1% detection threshold.

Evaluation metrics

We systematically evaluate the quality of simulated scRNA-seq data through a multi-dimensional assessment framework encompassing dataset attribute estimation, clustering performance, and biological signal preservation. The evaluation of dataset attributes focuses on 12 key statistical properties, including mean–variance relationships and zero-inflation patterns, which are quantitatively compared between real and simulated data. For clustering performance, we examine cell-type separation fidelity using established metrics such as silhouette width and adjusted Rand index (ARI). Biological signal preservation is assessed through differential expression detection ACC and gene–gene correlation maintenance. This approach ensures rigorous validation of both technical and biological data characteristics across all simulation methods.

Attribute estimation

In terms of dataset attribute estimation, a total of 12 dataset attributes are involved from the perspectives of univariate and binary attributes to comprehensively measure simulated data.

Univariate attributes measure cell-to-cell biological variability (biological coefficient of variation, BCV), library characteristics (Library Size, TMM Normalization Factors, and Effective Library Size), expression distribution (gene expression abundance, Fraction of zero per sample, and Fraction of zero per feature), and correlation structures

(Sample–sample correlations and Feature–feature correlations). Each of these attributes is defined as follows.

BCV quantifies the cell-to-cell variability in gene expression attributable to intrinsic biological heterogeneity, distinct from technical noise. Formally, for a given gene, BCV is defined as the standard deviation of its log-normalized expression values across cells divided by the mean expression (coefficient of variation), after accounting for technical artifacts. We used the prior degrees of freedom (prior.df) to quantify the strength of empirical Bayes shrinkage applied to gene-wise dispersion estimates in edgeR's negative binomial model [42]. A smaller prior.df indicates stronger shrinkage toward the trended dispersion–mean relationship, suggesting higher biological consistency across genes (typical of real scRNA-seq data), while larger values reflect weaker shrinkage and more gene-specific dispersion estimates (often seen in oversimplified simulations).

Library Size: This refers to the total read count for each cell, representing the overall gene expression in that cell.

TMM Normalization Factors: The trimmed mean of M-values (TMM) is a data normalization method commonly used in scRNA-seq analysis. M-values are the $\log_2$ ratios of gene expression between a given sample and a reference sample. To calculate the TMM normalization factors, M and A values (average expression levels) are computed between the reference and all other samples. Genes with M-values in the top 30% and bottom 30%, as well as genes with A-values in the top 5% and bottom 5%, are excluded. The remaining genes are used to calculate the weighted average of M-values, which serves as the normalization factor for each sample.

Effective library size: This is determined by dividing the calculated library size for each sample by its corresponding TMM normalization factor.

Gene expression distribution: This attribute describes the distribution of average gene expression abundance. Specifically, it is quantified by the log CPM (counts per million) value of each gene, with the average CPM used to represent its overall abundance.

Fraction of zero counts per sample: This represents the proportion of genes with zero expression out of the total number of genes in each sample.

Fraction of zero counts per feature: Similar to the previous attribute, this refers to the proportion of genes with zero expression across all samples.

Sample–sample correlations: This is the Spearman correlation between any two samples based on the gene expression matrix.

Feature–feature correlations: This refers to the Spearman correlation between pairs of genes in the gene expression matrix.

Binary attributes represent relationships between pairs of attributes, particularly focusing on the average–variance ratio, library size-to-zero ratio, and average expression-to-zero ratio. For the definitions of these three binary attributes, refer to the earlier definitions of the corresponding univariate attributes.

To quantitatively assess each attribute, six measurement parameters are utilized: average silhouette width (ASW), average local silhouette width (ALSW), Nearest Neighbor (NN) Rejection Fraction, Kolmogorov–Smirnov (K–S) Statistic, Scaled Area Between Empirical Cumulative Distribution Functions (eCDFs), and Runs Statistics. Each metric is described in detail as follows:

ASW: Measures clustering consistency by calculating the Euclidean distance between each sample and all other samples. The silhouette width for a sample is calculated using $s(i)$ for the simulated dataset and $r(i)$ for the real dataset. The contour width for a single sample is defined by Equation (7), and the ASW value is derived by averaging the contour widths across all samples. The ASW ranges from $[-1, 1]$, with values close to 1 indicating well clustering, and values near 0 indicating that the sample is near the decision boundary between clusters.

$$w(i) = \frac{s(i) - r(i)}{\max(r(i), s(i))} \tag{14}$$

ALSW: Similar to ASW, but computes the average distance to the k-nearest neighbors instead of all samples. It provides a more localized assessment of clustering quality.

NN Rejection Fraction: This metric assesses local dataset similarity by analyzing k-nearest neighbor (kNN) compositions. It is based on the null hypothesis that if the simulation is accurate, the proportion of real and simulated cells in any cell's k-neighborhood should reflect the global sample ratio. A chi-square test is applied to each cell; a P-value $< .05$ rejects the null hypothesis, indicating poor local mixing. The rejection fraction is the percentage of rejected cells among the total population. A lower value indicates a more faithful replication of the original local manifold structure.

K–S Statistic: Measures the distance between the cumulative distribution functions of the simulated data and real data. This statistic quantifies the maximum difference between the CDFs, and values closer to 0 indicate higher similarity between the distributions.

Scaled Area Between eCDFs: Quantifies the difference between two distributions by calculating the area between their cumulative distribution functions (eCDFs). Given samples X and Y, with sizes N and M, respectively, the area between the eCDFs $F_{X,N}$ and $F_{Y,M}$ is computed as shown in Equation (8). The product of M and N normalizes the area, adjusting for sample size and eliminating its influence on the distribution difference.

$$A = \frac{1}{mn} \int \left| F_{X,N}(x) - F_{Y,M}(x) \right| dx \tag{15}$$

Runs Statistic: Examines the randomness of observations to evaluate the similarity between two distributions. A "run" refers to a continuous set of the same observation. A negative value indicates fewer runs than expected, while a positive value indicates more. The statistic tests whether the two distributions are significantly different by comparing the observed runs to the expected number, with a P-value used to determine statistical significance.

Clustering performance

This paper primarily focuses on the differences between predicted and true labels for clustering performance metrics. The core metrics include ACC, ARI, Area Under the Curve (AUC), and F-Score. Each of these metrics is defined as follows:

ARI: ARI quantifies the similarity between clustering results and true categories, adjusting for random chance. A higher ARI indicates better clustering. The formula is given in Equation (15):

$$\mathrm{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] \Big/ \binom{n}{2}}{\frac{1}{2} \left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right] \Big/ \binom{n}{2}} \tag{16}$$

where $\binom{n_{ij}}{2}$ represents combinations, n_{ij} is the number of samples in category i in the true labels and category j in the predicted labels, a_i is the number of samples in category i in the true labels, b_j is the number

of samples in category j in the predicted labels, and n is the total sample sizes.

ACC: ACC is the ratio of correctly classified samples to the total number of samples, as shown in Equation (9):

$$\mathrm{ACC} = \frac{1}{N}\sum_{i=1}^{N} y_i = \mathbf{map}\left(p_i\right) \tag{17}$$

where p_i represents the real label of the sample i, y_i is the predicted label, and map(·) is a function that redistributes cluster labels, typically using the Hungarian method.

AUC: AUC is calculated by the area under the ROC curve, representing the probability that a randomly selected positive sample has a higher score than a randomly selected negative sample, as shown in Equation (10):

$$\mathrm{AUC} = \frac{\sum I(P_{\mathrm{pos}}, P_{\mathrm{neg}})}{M \times N} \tag{18}$$

Where $I(P_{\mathrm{pos}}, P_{\mathrm{neg}}) =$. In this context, P_{pos} and P_{neg} represent the predicted scores or probabilities assigned by the clustering model for a truly positive sample and a truly negative sample, respectively. M is the number of positive samples, and N is the number of negative samples.

F-Score: F-Score is a comprehensive evaluation metric considering both precision and recall. It is used to assess classification performance, as shown in Equation (11):

$$\mathbf{F.score} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{19}$$

Precision measures the proportion of true positive samples among those classified as positive, while *Recall* measures the proportion of actual positives identified correctly.

Biological signal detection

The preservation of biological signals is evaluated through differential analysis of five key transcriptional variation patterns. Differentially expressed (DE) genes are identified using the `limma` package with threshold of adjusted $P < .01$, representing genes with significant expression differences between cell populations. Differentially variable (DV) genes, showing distinct variability patterns, are detected through Bartlett's test of variance homogeneity ($P < .05$).

The analysis further examines differentially distributed (DD) genes using K–S tests between cell populations ($P < .05$), capturing genes with distinct expression distributions. Differential proportion (DP) genes are identified when genes exhibit significantly different expression proportions (chi-square test, $P < .05$), where expression is defined as $\mathbf{log}_2$ CPM > 1. Additionally, bimodal distribution (BD) genes are characterized through mixture modeling of expression distributions, revealing genes with distinct subpopulations.

DE genes are subclassified into up-regulated ($\mathbf{log}_2$ FC ≥ 0.5) and down-regulated ($\mathbf{log}_2$ FC ≤ -0.5) categories. The performance of each simulation method is quantified through two key metrics: the Recovery Rate, calculated as the proportion of real biological signals correctly identified in simulated data (|Simulated ∩ Real|/|Real|), and Precision, measuring the fraction of detected signals that match real observations (|Simulated ∩ Real|/|Simulated|).

Results and discussion

The framework of GDSim

We present GDSim, a novel deep generative framework for simulating scRNA-seq data with cell-type specificity. Built upon diffusion modeling principles, GDSim employs a neural network to predict the parameters of the reverse sampling distribution, enabling high-fidelity reconstruction of scRNA-seq data from noise while preserving cell-type characteristics. The complete workflow is illustrated in Fig. 1, with additional architectural details provided in Supplementary Figs S2–S9.

The GDSim framework operates through three fundamental phases. First, the data preprocessing stage prepares both the gene expression matrix and any available cell label information for model input. This involves normalizing expression values and converting categorical cell labels into numerical representations compatible with neural network processing (Fig. 1D).

Following preprocessing, the forward diffusion process systematically introduces controlled noise to the expression data through incremental perturbations. This gradual transformation preserves essential biological structures while guiding the data toward a noise-dominated state. When cell type information is provided, the diffusion process becomes label-conditioned, maintaining cell-type-specific patterns throughout the noise injection (Fig. 1C).

The final reverse sampling phase reconstructs the gene expression profiles through iterative denoising. A trained neural network predicts and removes the added noise in a stepwise manner, effectively recovering biologically plausible expression patterns. For label-conditioned simulations, this process generates data corresponding to specified cell types, while unsupervised operation produces data reflecting the general characteristics of the input distribution (Fig. 1A and B).

To rigorously evaluate GDSim's performance, we conducted comprehensive experiments assessing its ability to preserve key data attributes, maintain cluster relationships, and capture biological signals. Our benchmarking analysis compared GDSim against seven leading simulation methods using multiple evaluation metrics across diverse datasets.

GDSim accurately captures scRNA-seq data characteristics

To evaluate the biological fidelity of simulated data, we analyzed 12 key properties (9 univariates and 3 bivariates, detailed in Materials and methods) comparing GDSim against 7 established methods: ZingeR [43], POWSC [44], SPSimSeq [45], SPARSim [46], scDesign [31], scDesign2 [47], and SCRIP [48]. Using the countsimQC package [49], we systematically compared property distributions between real and simulated data across three benchmark datasets, including the human pluripotent stem cells dataset (hPSCs) [39], the human T lymphocyte line dataset (Jurkat) [40], and the mouse embryonic stem cells dataset (mESCs) [41] (See Dataset availability for details). Furthermore, to provide a more comprehensive evaluation, we extended our benchmarking to include five additional state-of-the-art generative methods (scDiffusion [50], scVI [51], ACTIVA [52], cscGAN [33], and scDesign3 [53]). Specifically, we conducted an in-depth comparison on the hPSC dataset, focusing on the distribution patterns and expression scaling characteristics of the ground truth (detailed results are provided in Supplementary Fig. S25).

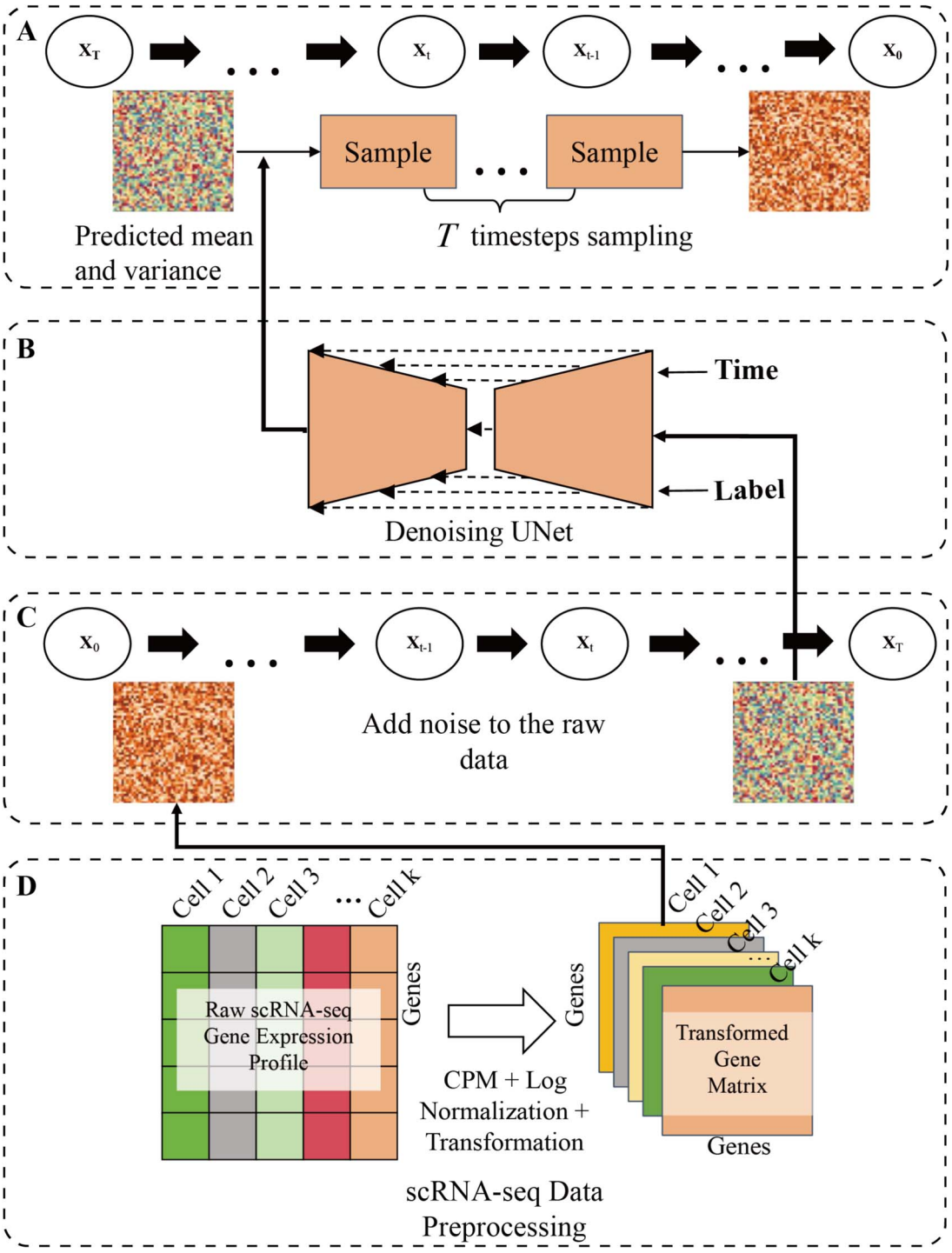

Figure 1 Overview of the GDSim framework for single-cell gene expression data simulation. The process consists of four key stages: (A) The reverse sampling process. (B) The training phase where the model learns to predict the added noise. (C) The forward diffusion process where noise is gradually introduced. (D) The preprocessing of scRNA-seq data. Once trained, the diffusion model accurately predicts the noise introduced during the forward process, allowing it to estimate the mean and variance of the reverse Markov process. This enables the generation of simulated scRNA-seq data by progressively removing the noise through reverse sampling.

Analysis of the hPSC dataset (GSE75748) [39] revealed that GDSim consistently generated the most biologically realistic simulations, as evidenced by its accurate preservation of both univariate and bivariate data properties (Fig. 2). Corresponding results for the remaining datasets are provided in Supplementary Figs S10–S21. Among the eight evaluated methods, GDSim achieved the closest match to real data distributions, along with POWSC, scDesign2, and SPARSim. For example, GDSim generated simulation data derived the smallest prior.df estimation, indicating superior maintenance of gene-wise dispersion relationships characteristic of authentic scRNA-seq

data. Our method also accurately reproduced other critical univariate properties.

Next, we conducted the quantitative assessment to evaluate how accurately the simulated data reproduced key properties of real scRNA-seq data. Our analysis employed six carefully selected distributional similarity metrics: ASW to assess cluster separation preservation, ALSW for evaluating fine-grained neighborhood structure, nearest neighbor (NN) rejection fraction quantifying local cell–cell relationships, K–S statistic comparing global expression distributions, scaled area between eCDFs measuring cumulative distribution alignment, and runs statistics testing spatial randomness in expression patterns (see Evaluation metrics for details).

We systematically evaluated the simulation quality by computing six distributional similarity metrics for twelve key data properties across all three benchmark datasets (hPSCs, Jurkat, and mESCs; Supplementary Tables S1–S12). For each property, metric scores were averaged across datasets, enabling a comprehensive ranking of all eight simulation methods based on their averaged performance (Supplementary Tables S13–S24). These comparative results are visualized in Fig. 3, which presents the relative rankings of each method across all evaluated properties through an intuitive heatmap representation. The gradient coloring scheme immediately reveals GDSim's consistent top-tier performance, with the highest contrast levels indicating superior rankings across nearly all property-metric combinations.

GDSim preserves cellular heterogeneity and enables faithful data augmentation

We evaluated GDSim's ability to maintain biological fidelity through two complementary analyses: (i) preservation of cellular subpopulation structure and (ii) low-error data amplification. First, we assessed whether simulated data retained the original dataset's spatial organization by comparing UMAP projections [54, 55] of real versus simulated cells in the hPSCs dataset. This dataset contains 1018 single cells representing seven distinct cell states (see Dataset availability). Notably, two closely related but distinct undifferentiated cell types (H1 and H9) can serve as a sensitive test for cluster preservation. As shown in Fig. 4A, our comparative analysis revealed that GDSim-generated data most accurately reproduced the global cellular architecture observed in the original dataset. When projecting all seven cell populations into UMAP space, GDSim simulations demonstrated near-identical spatial organization to real cells, maintaining both the relative positioning and distinct separation of each subpopulation. This fidelity was particularly evident in preserving the characteristic clustering patterns between the closely related H1 and H9 pluripotent stem cells while correctly maintaining their separation from differentiated lineages. Among the compared methods, only GDSim, POWSC, and SPARSim achieved this balance, while others either collapsed the distinction or introduced artificial separation. Quantitative validation using SC3 clustering [56] with four metrics (ACC, AUC, ARI, and F-score) confirmed these observations (Fig. 4B). GDSim-generated data showed high agreement with reference cell type labels. Notably, while methods like scDesign and scDesign2 maintained cluster structure, they lost subtle biological distinctions between related cell types.

We next evaluated each method's capacity for faithful data augmentation by assessing whether simulated cells could seamlessly integrate with real cells while preserving biological signatures. After merging real and simulated datasets (2:1 ratio), we performed UMAP visualization and quantitative mixing analysis. In the resulting projections (Fig. 4C), GDSim-generated cells showed near-perfect overlap with real cells across all seven subpopulations, with no systematic spatial segregation between real and simulated cells within each cell type cluster. SPARSim showed moderate but detectable separation artifacts, while other methods exhibited clear batch effects. GDSim-generated cells maintained robust cell-type-specific characteristics, as evidenced by their clear separation into seven distinct clusters corresponding to the original cell types (Fig. 4D). This clustering performance demonstrates that the simulated cells preserve the intrinsic transcriptional profiles of their respective cell types with high fidelity. Among all evaluated methods, only SPARSim achieved comparable cluster separation.

To quantitatively assess whether simulated cells are indistinguishable from real cells within each cell type, we applied the SC3 clustering algorithm to predict cell origins (real vs. simulated) and evaluated the performance using three metrics: Rand Index (RI), ACC, and AUC. In an ideal scenario where simulated cells perfectly match real cells, all three metrics should approach 0.5, indicating that the clustering algorithm cannot reliably distinguish between real and simulated cells. Our results demonstrate that GDSim consistently achieved this benchmark across all seven cell types, with all three metrics tightly clustered around 0.5 (Fig. 4E). SPARSim showed the next best performance, while the other five methods exhibited significantly poorer results, with metrics deviating substantially from the ideal 0.5 threshold.

These findings have important practical implications: when only limited real data are available due to cost or ethical constraints, GDSim can reliably generate expanded datasets while preserving the biological authenticity of the original data. This capability enables more robust downstream analyses by providing sufficient sample sizes without compromising data quality. Notably, GDSim's superior performance across all cell types suggests it is particularly well suited for applications requiring high-fidelity simulation of diverse cellular populations.

GDSim preserves biological signals in differential expression analysis

To evaluate whether the simulated dataset retains the biological signals of the real dataset, we selected the two types of cells with the largest population in the GSE75748 dataset, namely H1 cells ($n = 212$) and NPC cells ($n = 173$), and performed gene differential expression analysis on these two types of cells. We also simulated the same number of cells in each group and performed similar differential expression analysis. We detected a total of five types of DE genes, including DE genes, DV genes, DD genes, DP genes, and BD genes (see Evaluation metrics). The proportions of the five DE genes detected in the simulated data and the real data were compared, as shown in Fig. 5A. In the results of DE, DV, DD, and BD, the simulated data produced by GDSim retained the most similar proportion of DE genes to the real data. Three methods, including zingeR, SPsimSeq, and SCRIP, detected fewer DE genes, which does not conform to the situation of real data.

Focusing on DE genes (adjusted P-value $\leq .01$), Fig. 5B demonstrates GDSim's superior performance in recovering true biological signals. The simulated data generated by GDSim captured 88% of the 8033 DE genes identified in real data while maintaining an 88% detection ACC—the highest among all methods except SPARSim (91% ACC). This strong concordance with biological ground truth significantly

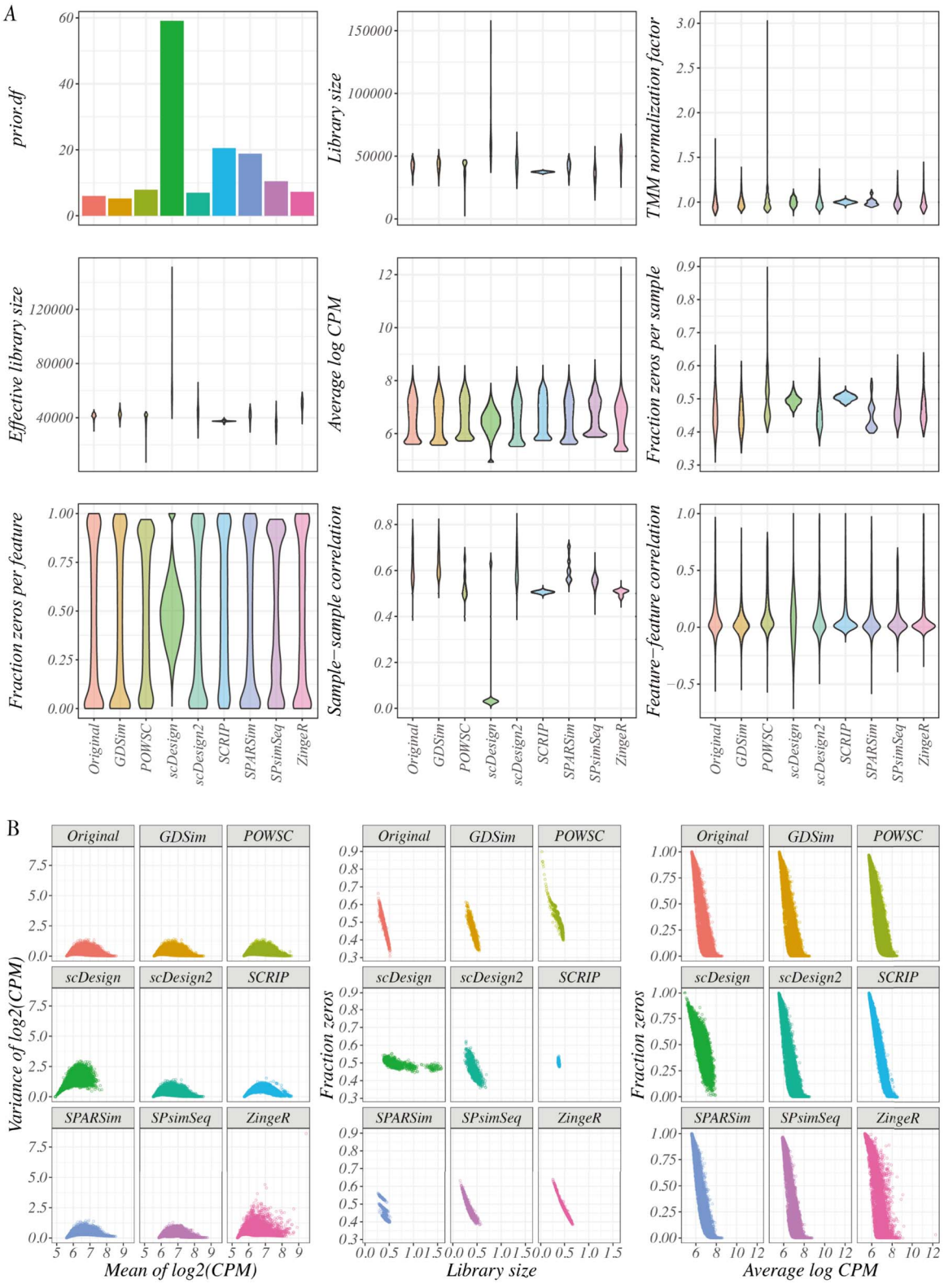

Figure 2 The estimation results of dataset attributes. (A) The estimation results of the nine univariate attributes of the dataset itself, where each subplot represents a method on the x-axis and the results for each attribute are represented on the y-axis. (B) The estimation results of the three bivariate attributes of the dataset itself, where each subplot has one univariate attribute on the x-axis and another on the y-axis, with different colors indicating different methods.

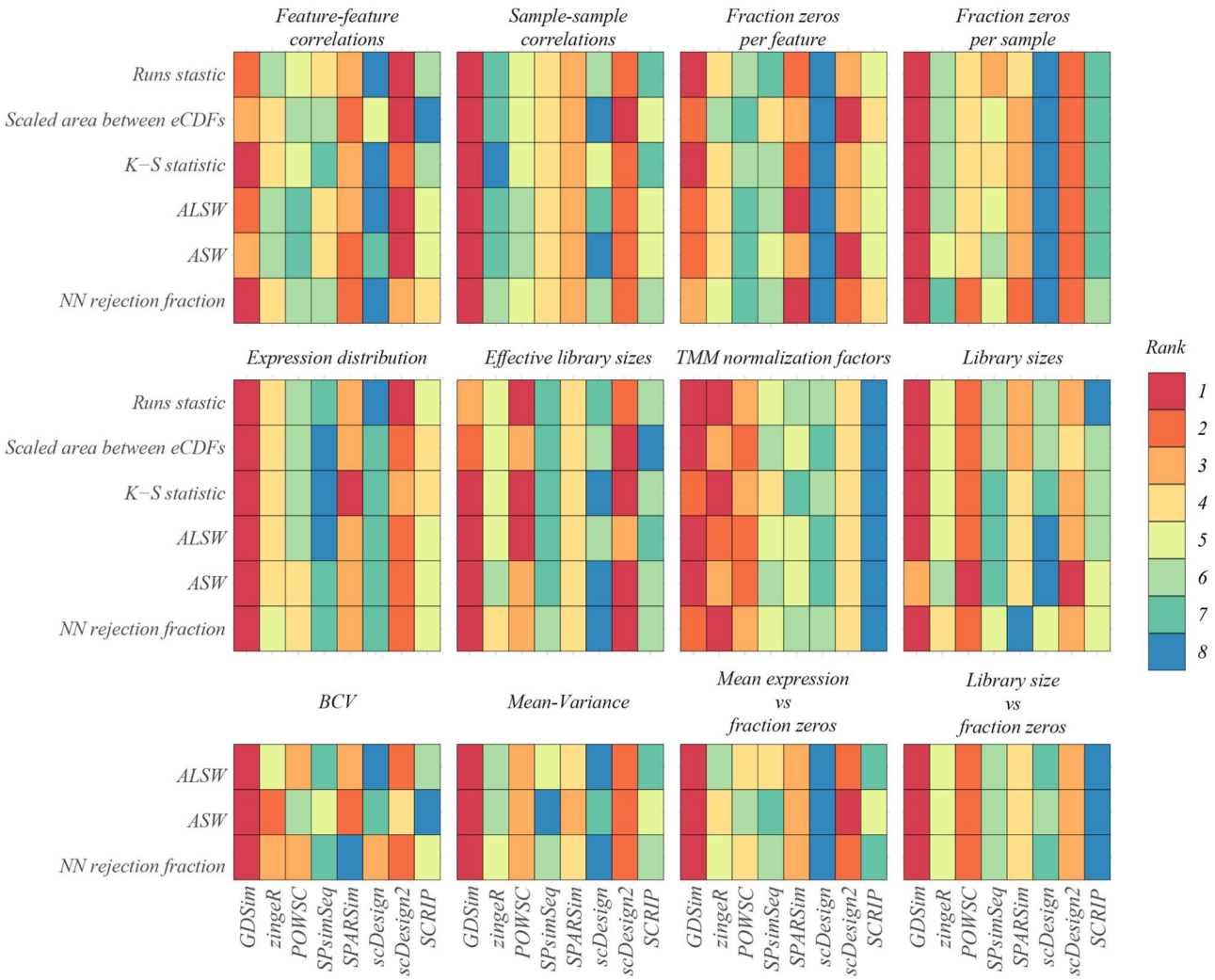

Figure 3 The comprehensive performance of scRNA-seq data simulation methods across three datasets. The heatmap illustrates the ability of eight methods to recover data attributes on three different datasets. Each sub-heatmap represents the quantitative discrepancy between simulated data and real data estimation results for a specific data attribute. The x-axis represents the methods, while the y-axis represents the quantitative results. Binary attributes are evaluated using only the last three quantitative metrics (ASW, ALSW, and NN rejection fraction), as they capture pairwise structure and clustering similarity. In contrast, the remaining metrics are designed for univariate distribution testing and are not suitable for assessing inter-variable relationships.

outperformed competing approaches: POWSC, SPsimSeq, scDesign, and scDesign2 showed substantially lower detection rates, while zingeR and SCRIP failed to identify any DE genes (Supplementary Table S25). The superior performance of GDSim remained consistent when applying less stringent statistical thresholds (adjusted *P*-value $\leq .1$, Supplementary Figs S23 and S24).

We conducted a detailed examination of differential expression patterns by analyzing up-regulated and down-regulated genes (adjusted *P*-value $\leq .01$ and $|\log_2 \text{FC}| > 0.5$) through volcano plot visualization (Fig. 5C). The plots display $\log_2$ fold changes between H1 and NPC cells on the horizontal axis and statistical significance ($-\log_{10}$(adjusted *P*-value)) on the vertical axis, with up-regulated genes shown in red (left), down-regulated genes in blue (right), and non-DE genes in gray (middle). GDSim demonstrated remarkable fidelity in reproducing the differential expression patterns observed in real data, achieving a Pearson correlation coefficient of 0.92 for $\log_2$ fold change values. This performance significantly outperformed competing methods such as scDesign, scDesign2, and SPsimSeq, which showed substantial deviations from the ground truth. Both GDSim and SPARSim maintained exceptional ACC, exceeding 90% in DE gene detection, as detailed in Supplementary Tables S26 and S27. Figure 5D illustrates this capability through the perfect reproduction of the most significant DE gene (ESRP1) with additional top 3 significant examples (IL34, TDGF1, and IGFBP5) provided in Supplementary Fig. S22. These results collectively demonstrate GDSim's superior ability to maintain cell-type-specific expression patterns during simulation.

Computational performance and robustness of GDSim

To provide a practical reference for users, we conducted a comprehensive quantitative analysis of the computational overhead for GDSim and 12 representative baseline methods using the hPSC dataset. The evaluation encompassed total runtime, peak memory consumption, and sampling speed. To ensure a fair comparison, all methodologies were implemented on a high-performance server equipped with an

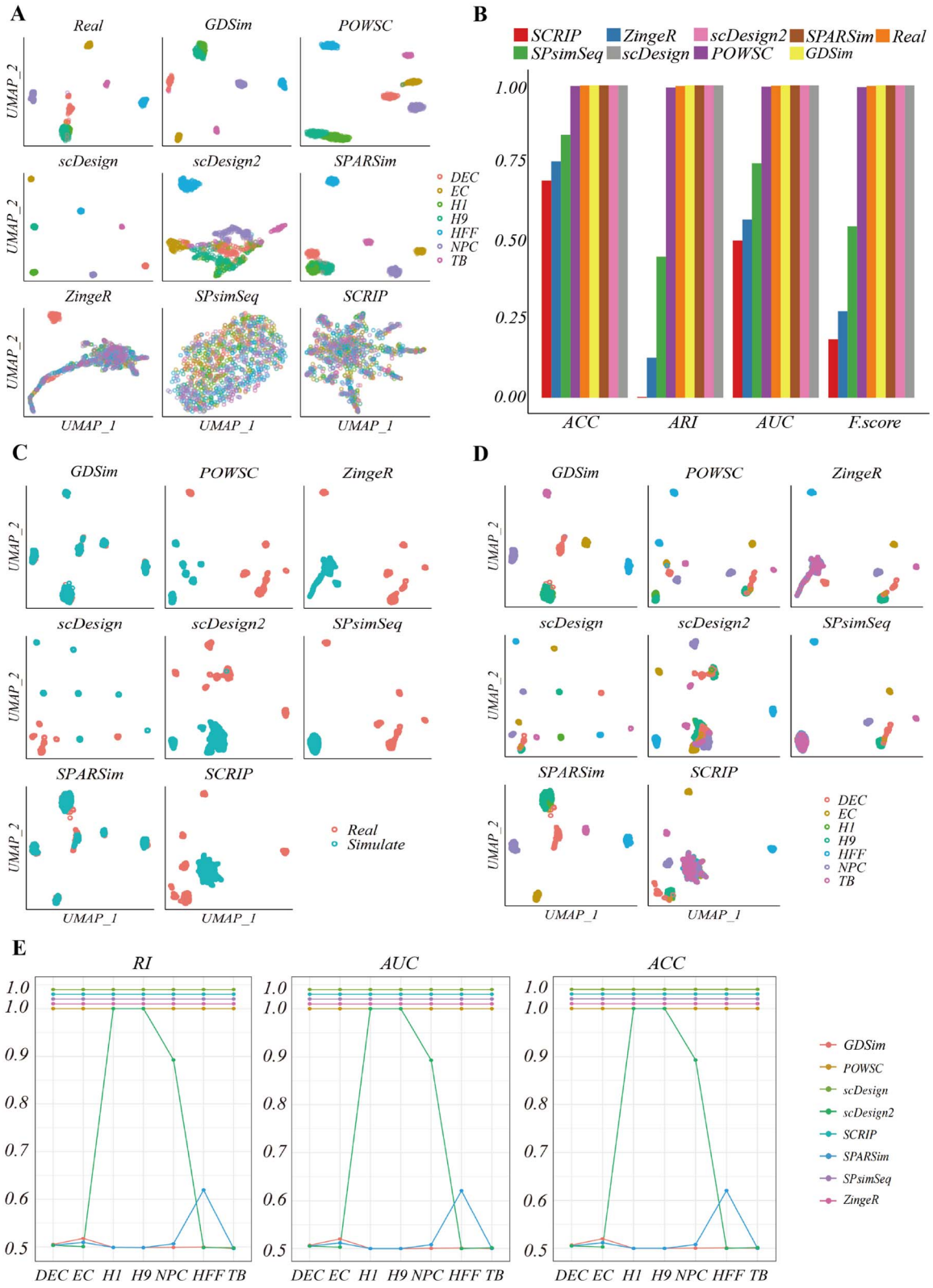

Figure 4 Comprehensive evaluation of simulated data quality through clustering analysis. (A) UMAP visualization comparing the global transcriptional structure between real data and simulated data, demonstrating preservation of cell-type clusters and population heterogeneity. (B) Quantitative assessment using SC3 clustering, with performance metrics (ACC, AUC, ARI, and F-score) comparing predicted versus true cell-type annotations. (C) Joint UMAP visualization of merged real and simulated data, showing near-perfect overlap when colored by data source, confirming the indistinguishability of GDSim-generated cells. (D) Cell-type distribution in the merged dataset, demonstrating that GDSim simulated cells maintain appropriate type-specific positioning relative to their real counterparts. (E) Prediction results for cell origin (real versus simulated) in the merged dataset, with optimal performance approaching random chance (0.5).

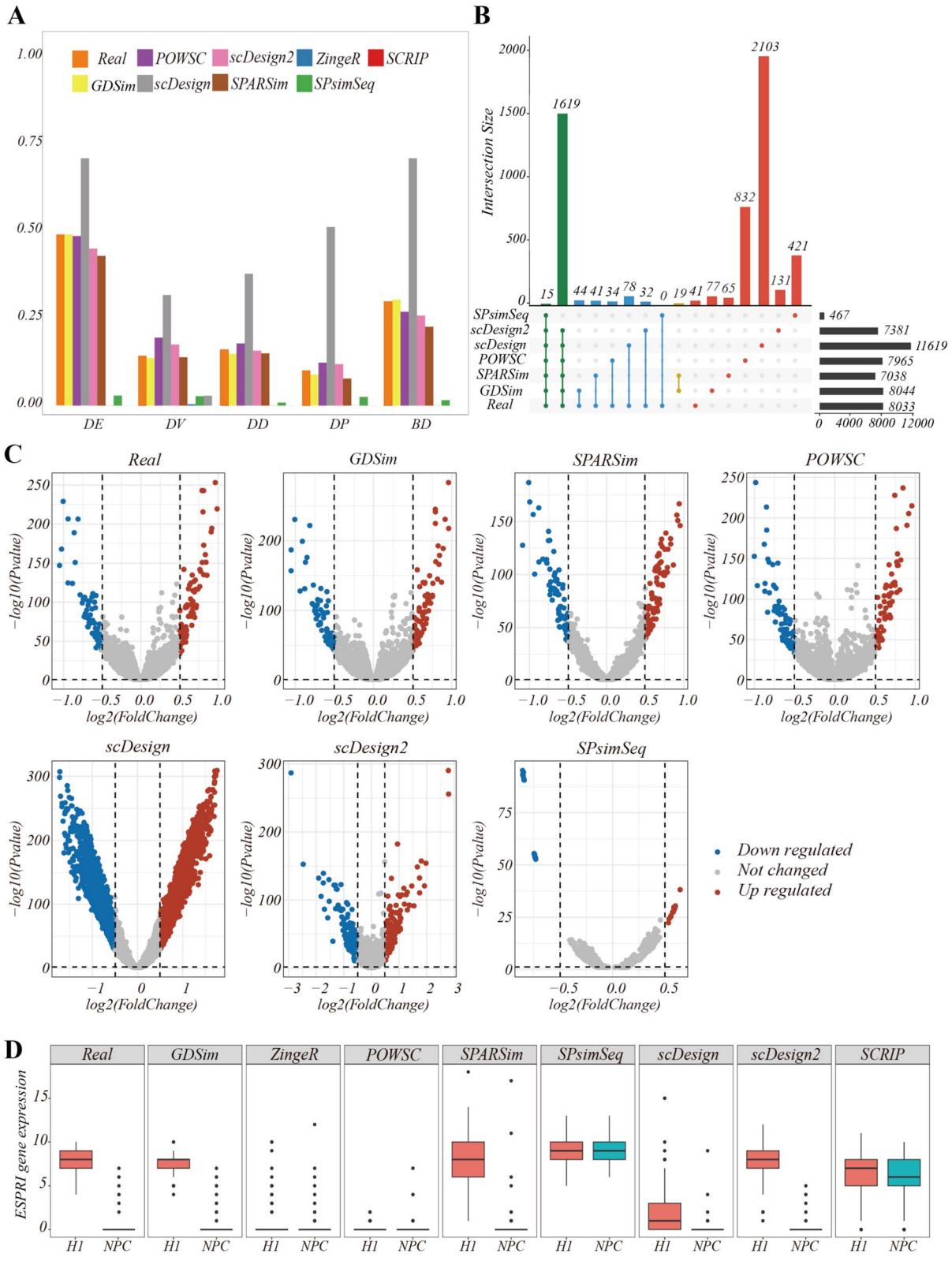

Figure 5 Evaluation of differential gene expression preservation in simulated datasets. (A) Comparison of five differential gene classes detected across methods. Bars show counts of DE, DV, DD, DP, and BD genes, with GDSim most closely matching real data. (B) Upset diagrams quantifying overlap between DE genes identified in real data versus simulated datasets. (C) Volcano plots comparing differential expression between H1 and NPC cells. (D) Expression distributions of top DE gene ESRP1 across datasets.

Intel(R) Xeon(R) Gold 6248R CPU (48 total cores, 512 GB total memory) and an NVIDIA RTX A6000 GPU (48 GB VRAM).

As illustrated in Supplementary Fig. S26, our results reveal a clear trade-off between computational investment and simulation fidelity across different modeling paradigms. Traditional parametric statistical methods, such as POWSC and ZingeR, exhibit the highest efficiency, requiring only seconds to minutes with minimal memory footprint. In contrast, complex generative architectures, including GDSim and the peer diffusion-based method scDiffusion, necessitate a greater computational investment to achieve higher-fidelity data reconstruction. The primary contributor to GDSim's resource demands is its guided diffusion architecture, which utilizes a 1000-step iterative denoising process to ensure biological ACC. Specifically, during the sampling phase, GDSim (6.8 s/cell) is more demanding than non-iterative frameworks like scVI (VAE-based) or cscGAN (GAN-based). However, it remains significantly more efficient than the most time-consuming statistical method, SPsimSeq (11.9 s/cell), demonstrating that iterative diffusion is not necessarily the slowest approach in the current simulation landscape. Notably, GDSim exhibits superior resource management in terms of memory usage (18.09 GB). In comparison, several highly parameterized statistical simulators, such as scDesign2, scDesign3, and SCRIP, all exceeded 27 GB in peak memory usage. This suggests that while GDSim is a deep learning model, it avoids the risk of "memory explosion" often associated with complex statistical parameter estimation when scaling to large datasets.

To further assess the scalability and generalization capability of GDSim on large-scale datasets, we conducted a systematic evaluation using the Tabula Sapiens human cell atlas [57], containing 49 357 cells spanning five different organs. To rigorously challenge the model, only 10% of the original data were used for training, with the goal of inferring and reconstructing the full-scale data manifold (i.e. 49 357 cells). As shown in Supplementary Fig. S27, GDSim effectively reconstructs the global topological structure of the atlas. In the integrated UMAP space, GDSim-generated cells exhibit seamless alignment and extensive intermixing with the original cells across both major cell-type clusters and relatively sparse populations. This deep integration pattern indicates that GDSim learns the underlying continuous probability density of the data rather than simply memorizing training samples.

A critical benchmark for a high-fidelity simulator is its ability to capture continuous biological processes, such as cellular development and differentiation. To address this, we evaluated GDSim using a mouse embryonic stem cell dataset (GSE90047) encompassing seven continuous developmental stages from E10.5 to E17.5. We employed the PAGA algorithm to reconstruct the developmental backbone and compare the topological structure between real and simulated data. As illustrated in Supplementary Fig. S28, GDSim successfully recovers the intrinsic continuous structure of the developmental manifold. At the global topological level, the model accurately reconstructs the developmental backbone, capturing the sequential evolution from early-stage progenitors to late-stage differentiated cells. Specifically, it accurately reproduces high-confidence connections in the PAGA graph between critical stages, indicating that GDSim has learned the underlying probability density gradients along the developmental trajectory.

Finally, we evaluated the representational requirements and robustness of GDSim under class-imbalanced conditions using a systematic subsampling experiment on the hPSC dataset. Specifically, we selected transcriptionally similar H1 and H9 cells, together with the more distinct NPC population. As shown in Supplementary Fig. S29, GDSim demonstrates strong robustness against extreme class imbalance. Even when a specific cell type (e.g. H9) is reduced to only 5% of the total population (47.5%: 47.5%: 5%), GDSim generates well-defined clusters without loss or erroneous merging of the low-abundance class. Despite the high transcriptional similarity between H1 and H9 cells, GDSim successfully captures the subtle expression differences required to distinguish them.

In summary, these results collectively indicate that GDSim achieves a favorable balance among computational efficiency, memory robustness, and generalization performance on large-scale heterogeneous datasets. Even under a very low training sampling rate or extreme class imbalance, GDSim is able to effectively reconstruct high-fidelity data manifolds, underscoring its practical utility and reliability for atlas-scale single-cell data simulation tasks.

Conclusions

The generation of realistic scRNA-seq data through computational simulation has become indispensable for advancing biological research, enabling critical applications in method benchmarking, experimental design, and rare cell population analysis. While existing approaches have provided foundational solutions, they remain fundamentally limited by either simplistic statistical assumptions that fail to capture biological complexity or inadequate preservation of crucial molecular signatures.

To address these limitations, we developed GDSim, a novel guided diffusion model that redefines scRNA-seq simulation through two key innovations. First, our framework replaces restrictive parametric assumptions with a dynamic denoising process that learns authentic gene expression distributions directly from data. Second, the incorporation of cell-type guidance enables precise generation of specific cellular populations while maintaining their characteristic biological signatures. Rigorous evaluation demonstrates that GDSim outperforms current methods in preserving both global data structure and fine-grained transcriptional patterns, achieving superior performance in cluster fidelity, differential expression recovery, and data distribution metrics. By bridging the gap between computational simulation and biological reality, we believe GDSim will establish itself as a valuable software for single-cell genomics research.

Key points

- This work presents GDSim, a novel deep generative framework that leverages diffusion models to enable efficient and accurate simulation of single-cell RNA sequencing (scRNA-seq) data.
- Unlike conventional approaches that rely on restrictive distributional assumptions, GDSim employs a cell-type guided diffusion process combined with non-parametric modeling to generate realistic gene expression profiles for specified cell populations.
- GDSim significantly improves the ACC and reliability of essential scRNA-seq analyses, including quantitative estimation of dataset attributes, identification of cell subpopulations through clustering, and detection of DE genes.
- Benchmarking against seven state-of-the-art methods demonstrates GDSim's superior performance in preserving biologically meaningful patterns.

Author contributions

T.W., J.C., and X.S. conceived the study and experiments. H.Z., H.D., and J.C. conducted the experiments, T.W., H.Z., H.D, Y.W., X.S., J.P., J.C., and B.X. analyzed the results. T.W., J.C., H.Z., and H.D. wrote and reviewed the manuscript.

Supplementary material

Supplementary material is available at *Briefings in Bioinformatics* online.

Conflicts of interest

None declared.

Funding

This work has been supported by the National Key R&D Program of China (2025YFC3410200) and National Natural Science Foundation of China (grant numbers: 62402382, 62433016, and 62102319).

Data availability

All data used in this study is publicly available and can be accessed through: (i) GSE75748: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE75748, (ii) Jurkat: https://www.10xgenomics.com/datasets/jurkat-cells-1-standard-1-1-0, (iii) GSE90047: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE90047, (iv) Tabula Sapiens human cell atlas: https://figshare.com/s/49b29cb24b27ec8b6d72. The codebase for GDSim is publicly available at https://github.com/Galaxy8172/GDSim.

References

1. Mereu E, Lafzi A, Moutinho C *et al.* Benchmarking single-cell RNA-sequencing protocols for cell atlas projects. *Nat Biotechnol* 2020;**38**:747–55. https://doi.org/10.1038/s41587-020-0469-4
2. Xiangyu W, Yang X, Dai Y *et al.* Single-cell sequencing to multi-omics: technologies and applications. *Biomarker Res* 2024;**12**:110. https://doi.org/10.1186/s40364-024-00643-4
3. Cha J, Lee I. Single-cell network biology for resolving cellular heterogeneity in human diseases. *Exp Mol Med* 2020;**52**:1798–808. https://doi.org/10.1038/s12276-020-00528-0
4. Hsieh C-Y, Wen J-H, Lin S-M *et al.* scDrug: from single-cell RNA-seq to drug response prediction. *Comput Struct Biotechnol J* 2023;**21**:150–7. https://doi.org/10.1016/j.csbj.2022.11.055
5. Jovic D, Liang X, Zeng H *et al.* Single-cell RNA sequencing technologies and applications: a brief overview. *Clin Transl Med* 2022;**12**:e694. https://doi.org/10.1002/ctm2.694
6. Bagnoli JW, Ziegenhain C, Janjic A *et al.* Sensitive and powerful single-cell RNA sequencing using mcSCRB-seq. *Nat Commun* 2018;**9**:2937. https://doi.org/10.1038/s41467-018-05347-6
7. Li X, Wang C-Y. From bulk, single-cell to spatial RNA sequencing. *Int J Oral Sci* 2021;**13**:36. https://doi.org/10.1038/s41368-021-00146-0
8. Stegle O, Teichmann SA, Marioni JC. Computational and analytical challenges in single-cell transcriptomics. *Nat Rev Genet* 2015;**16**:133–45. https://doi.org/10.1038/nrg3833
9. Andrews TS, Kiselev VY, McCarthy D *et al.* Tutorial: guidelines for the computational analysis of single-cell RNA sequencing data. *Nat Protoc* 2021;**16**:1–9. https://doi.org/10.1038/s41596-020-00409-w
10. Chen G, Ning B, Shi T. Single-cell RNA-seq technologies and related computational data analysis. *Front Genet* 2019;**10**:317. https://doi.org/10.3389/fgene.2019.00317
11. Mustapha SMFDS. High-dimensional data analysis using parameter free algorithm data point positioning analysis. *Appl Sci* 2024;**14**:4231. https://doi.org/10.3390/app14104231
12. Dai C, Jiang Y, Yin C *et al.* scIMC: a platform for benchmarking comparison and visualization analysis of scRNA-seq data imputation methods. *Nucleic Acids Res* 2022;**50**:4877–99. https://doi.org/10.1093/nar/gkac317
13. Chen W, Zhao Y, Chen X *et al.* A multicenter study benchmarking single-cell RNA sequencing technologies using reference samples. *Nat Biotechnol* 2021;**39**:1103–14. https://doi.org/10.1038/s41587-020-00748-9
14. Nguyen HCT, Baik B, Yoon S *et al.* Benchmarking integration of single-cell differential expression. *Nat Commun* 2023;**14**:1570. https://doi.org/10.1038/s41467-023-37126-3
15. Zhen C, Wang Y, Jiaquan Geng L *et al.* A review and performance evaluation of clustering frameworks for single-cell Hi-C data. *Brief Bioinform* 2022;**23**:bbac385. https://doi.org/10.1093/bib/bbac385
16. Hu Y, Wan S, Luo Y *et al.* Benchmarking algorithms for single-cell multi-omics prediction and integration. *Nat Methods* 2024;**21**:2182–94. https://doi.org/10.1038/s41592-024-02429-w
17. Tian L, Dong X, Freytag S *et al.* Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nat Methods* 2019;**16**:479–87. https://doi.org/10.1038/s41592-019-0425-8
18. Li Z, Patel ZM, Song D *et al.* Benchmarking computational methods to identify spatially variable genes and peaks. *Genome Biol* 2025;**26**:285. https://doi.org/10.1101/2023.12.02.569717
19. Benham-Pyle BW, Brewster CE, Kent AM *et al.* Identification of rare, transient post-mitotic cell states that are induced by injury and required for whole-body regeneration in Schmidtea mediterranea. *Nat Cell Biol* 2021;**23**:939–52. https://doi.org/10.1038/s41556-021-00734-6
20. Wang T, Zhao H, Yungang X *et al.* scMultiGAN: cell-specific imputation for single-cell transcriptomes with multiple deep generative adversarial networks. *Brief Bioinform* 2023;**24**:bbad384.
21. Shu H, Chen J, Jialu H *et al.* stSCI: a multi-task learning framework for integrative analysis of single-cell and spatial transcriptomics data. *Innovation* 2025;**7**:101220. https://doi.org/10.1016/j.xinn.2025.101220
22. Gevertz JL, Kareva I. Minimally sufficient experimental design using identifiability analysis. *NPJ Syst Biol Appl* 2024;**10**:2. https://doi.org/10.1038/s41540-023-00325-1
23. Yang W, Wang P, Shouping X *et al.* Deciphering cell–cell communication at single-cell resolution for spatial transcriptomics with subgraph-based graph attention network. *Nat Commun* 2024;**15**:7101. https://doi.org/10.1038/s41467-024-51329-2
24. Wang T, Rentería ME, Tian Z *et al.* Data mining and statistical methods for knowledge discovery in diseases based on multimodal omics, volume II[J]. *Frontiers in Genetics* 2023;**14**:1270862. https://doi.org/10.3389/fgene.2023.1270862
25. Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol* 2017;**18**:174. https://doi.org/10.1186/s13059-017-1305-0
26. Schiebinger G, Shu J, Tabaka M *et al.* Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell* 2019;**176**:928–943.e22. https://doi.org/10.1016/j.cell.2019.01.006

27. Vieth B, Ziegenhain C, Parekh S *et al.* powsimR: power analysis for bulk and single cell RNA-seq experiments. *Bioinformatics* 2017;**33**:3486–8. https://doi.org/10.1093/bioinformatics/btx435
28. Korthauer KD, Chu L-F, Newton MA *et al.* A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biol* 2016;**17**:1–15. https://doi.org/10.1186/s13059-016-1077-y
29. Risso D, Perraudeau F, Gribkova S *et al.* A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat Commun* 2018;**9**:284. https://doi.org/10.1038/s41467-017-02554-5
30. Van den Berge K, Perraudeau F, Soneson C *et al.* Observation weights unlock bulk RNA-seq tools for zero inflation and single-cell applications. *Genome Biol* 2018;**19**:1–17. https://doi.org/10.1186/s13059-018-1406-4
31. Li WV, Li JJ. A statistical simulator scDesign for rational scRNA-seq experimental design. *Bioinformatics* 2019;**35**:i41–50. https://doi.org/10.1093/bioinformatics/btz321
32. Zhang X, Chenling X, Yosef N. Simulating multiple faceted variability in single cell RNA sequencing. *Nat Commun* 2019;**10**:2611. https://doi.org/10.1038/s41467-019-10500-w
33. Marouf M, Machart P, Bansal V *et al.* Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks. *Nat Commun* 2020;**11**:166. https://doi.org/10.1038/s41467-019-14018-z
34. Liu Y, Wang W, Fang F *et al.* CscGAN: conditional scale-consistent generation network for multi-level remote sensing image to map translation. *Remote Sens* 2021;**13**:1936. https://doi.org/10.3390/rs13101936
35. Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. *Adv Neural Inf Proces Syst* 2020;**33**:6840–51.
36. Rombach R, Blattmann A, Lorenz D *et al.* High-resolution image synthesis with latent diffusion models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, 2022, 10684–95. https://doi.org/10.48550/arXiv.2112.10752
37. Watson JL, Juergens D, Bennett NR *et al.* De novo design of protein structure and function with RFdiffusion. *Nature* 2023;**620**:1089–100. https://doi.org/10.1038/s41586-023-06415-8
38. Singh A. Chroma is a generative model for protein design. *Nat Methods* 2024;**21**:10–0. https://doi.org/10.1038/s41592-023-02155-9
39. Chu L-F, Leng N, Zhang J *et al.* Single-cell RNA-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome Biol* 2016;**17**:1–20. https://doi.org/10.1186/s13059-016-1033-x
40. Abraham RT, Weiss A. Jurkat T cells and development of the T-cell receptor signalling paradigm. *Nat Rev Immunol* 2004;**4**:301–8. https://doi.org/10.1038/nri1330
41. Yang L, Wang W-H, Qiu W-L *et al.* A single-cell transcriptomic analysis reveals precise pathways and regulatory mechanisms underlying hepatoblast differentiation. *Hepatology* 2017;**66**:1387–401. https://doi.org/10.1002/hep.29353
42. Chen Y, McCarthy D, Ritchie M *et al.* edgeR: differential analysis of sequence read count data user's guide. R Packag. 2020;1–121.
43. Van den Berge K, Soneson C, Love MI *et al.* Zinger: unlocking RNA-seq tools for zero-inflation and single cell applications. biorxiv. *Preprint*. 2017;**10**:157982. https://doi.org/10.1101/157982
44. Kenong S, Zhijin W, Hao W. Simulation, power evaluation and sample size recommendation for single-cell RNA-seq. *Bioinformatics* 2020;**36**:4860–8. https://doi.org/10.1093/bioinformatics/btaa607
45. Assefa AT, Vandesompele J, Thas O. SPsimSeq: semi-parametric simulation of bulk and single-cell RNA-sequencing data. *Bioinformatics* 2020;**36**:3276–8. https://doi.org/10.1093/bioinformatics/btaa105
46. Baruzzo G, Patuzzi I, Di Camillo B. SPARSim single cell: a count data simulator for scRNA-seq data. *Bioinformatics* 2020;**36**:1468–75. https://doi.org/10.1093/bioinformatics/btz752
47. Sun T, Song D, Li WV *et al.* scDesign2: a transparent simulator that generates high-fidelity single-cell gene expression count data with gene correlations captured. *Genome Biol* 2021;**22**:163. https://doi.org/10.1186/s13059-021-02367-2
48. Qin F, Luo X, Xiao F *et al.* SCRIP: an accurate simulator for single-cell RNA sequencing data. *Bioinformatics* 2022;**38**:1304–11. https://doi.org/10.1093/bioinformatics/btab824
49. Soneson C, Robinson MD. Towards unified quality verification of synthetic count data with countsimQC. *Bioinformatics* 2018;**34**:691–2. https://doi.org/10.1093/bioinformatics/btx631
50. Luo E, Hao M, Wei L *et al.* scDiffusion: conditional generation of high-quality single-cell data using diffusion model. *Bioinformatics* 2024;**40**:btae518.
51. Gayoso A, Lopez R, Xing G *et al.* A python library for probabilistic analysis of single-cell omics data. *Nat Biotechnol* 2022;**40**:163–6. https://doi.org/10.1038/s41587-021-01206-w
52. Ali Heydari A, Davalos OA, Zhao L *et al.* ACTIVA: realistic single-cell RNA-seq generation with automatic cell-type identification using introspective variational autoencoders. *Bioinformatics* 2022;**38**:2194–201. https://doi.org/10.1093/bioinformatics/btac095
53. Song D, Wang Q, Yan G *et al.* scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics. *Nat Biotechnol* 2024;**42**:247–52. https://doi.org/10.1038/s41587-023-01772-1
54. Pierson E, Yau C. ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol* 2015;**16**:1–10.
55. Sun S, Zhu J, Ma Y *et al.* Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis. *Genome Biol* 2019;**20**:1–21. https://doi.org/10.1186/s13059-019-1898-6
56. Kiselev VY, Kirschner K, Schaub MT *et al.* Sc3: consensus clustering of single-cell RNA-seq data. *Nat Methods* 2017;**14**:483–6. https://doi.org/10.1038/nmeth.4236
57. The Tabula Sapiens Consortium, Jones RC, Karkanias J *et al.* The tabula sapiens: a multiple-organ, single-cell transcriptomic atlas of humans. *Science* 2022;**376**:eabl4896.

Briefings in Bioinformatics, 2026, **27**, bbag163
https://doi.org/10.1093/bib/bbag163
Problem Solving Protocol