Bioinformatics, 2026, **42**(3), btag095
https://doi.org/10.1093/bioinformatics/btag095
Published: 26 February 2026
Original Paper | Gene expression

OXFORD

scDBic: a novel deep learning-based biclustering algorithm for analyzing scRNA-seq data

Xiaoqi Tang, Caihua Liu, Chaowang Lan*

Guangxi Key Laboratory of Robot Intelligent Perception and Control, School of Artificial Intelligence, Guilin University of Electronic Technology, No.1 Jinji Road, 541004, Guilin, China
*Corresponding author. Guangxi Key Laboratory of Robot Intelligent Perception and Control, School of Artificial Intelligence, Guilin University of Electronic Technology, No.1 Jinji Road, 541004, Guilin, China. E-mail: chaowanglan@guet.edu.cn.
Associate Editor: Jonathan Wren

Abstract

Motivation: **Clustering single-cell RNA sequencing (scRNA-seq) data plays a vital role in the study of cellular heterogeneity. Many algorithms have been developed to cluster scRNA-seq data. However, traditional clustering algorithms often fail to capture local consistency, whereas biclustering algorithms suffer from issues such as cell loss, poor adaptability to high-dimensional data, and iterative selection challenges.**

Results: **In this paper, we introduce scDBic, a novel deep learning-based biclustering algorithm specialized for scRNA-seq data. It comprises three main steps: cell clustering with a deep autoencoder, gene clustering, and identification of key gene clusters using the reverse strategy. The key idea is that the deep autoencoder captures the main information of gene expression and the reverse strategy identifies the key genes of cell groups. Therefore, cell clustering performance can be improved. The results demonstrate that our algorithm not only discovers cell groups in scRNA-seq data but also identifies the key genes of the cell groups. Furthermore, the clustering performance of our algorithm is better than that of traditional clustering and biclustering algorithms. This novel technique can be directly applied to discover cell groups and identify key genes in cell groups.**

Availability and implementation: **The source code and test data are freely available at GitHub (https://github.com/Xiaoqi-Tang/scDBic) and archived on Zenodo (DOI: 10.5281/zenodo.18676401).**

1 Introduction

A large number of biological processes are accomplished through the collective efforts of cells within organisms (Celiker and Gore 2013). Understanding how cells partition tasks, coordinate activities, and manage biological processes remains at a rudimentary stage. Single-cell RNA sequencing (scRNA-seq) technology enables researchers to precisely dissect biological processes at the single-cell level (Rosenberg *et al.* 2018).

However, the analysis of scRNA-seq data faces several challenges. Owing to cell heterogeneity, high cell dropout rates, low expression genes, technical biases, and batch effects, scRNA-seq data is high-dimensional, sparse, and noisy, affecting subsequent analysis.

Clustering is a key computational method for analyzing scRNA-seq data and is an effective method for studying cellular heterogeneity, which provides insights into disease mechanisms. For example, previous studies have confirmed that CD8+ T cells are critical for autoimmune lymphoproliferative syndromes (Eberhardt *et al.* 2021). However, intervention therapy targeting all CD8+ T cells indiscriminately could pose significant harm to the human body, potentially outweighing the risks of the disease. Therefore, identifying pathogenic CD8+ T cells is of utmost importance in the treatment of autoimmune lymphoproliferative syndrome. In addition, clustering can reveal disease-causing mechanisms (Li *et al.* 2021b), identify different cell types and functional states (Zheng *et al.* 2017), discover rare cell populations, reconstruct cell developmental trajectories (Trapnell *et al.* 2014), and build spatial models of complex tissues (Satija *et al.* 2015), etc.

Traditional clustering methods apply global gene information to determine different cell types, making it difficult to capture local consistency (Hussain and Ramazan 2016). This phenomenon refers to local gene-cell co-regulatory patterns where specific subsets of genes are co-regulated only within specific subsets of cells. Existing studies have demonstrated that only a small subset of genes play a role in specific cell types or states within scRNA-seq data (Sun *et al.* 2023). Therefore, the traditional clustering method may not be suitable for clustering the scRNA-seq data. Biclustering methods cluster both cells and genes

Received: 19 July 2025. **Revised:** 30 December 2025. **Accepted:** 16 February 2026
© The Author(s) 2026. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

simultaneously and cannot only identify different cell types but also discover potential key genes that are active in different cell types.

Biclustering algorithms can be broadly categorized into greedy methods (Xue *et al.* 2018, Torres-Jimenez and Perez-Torres 2019), divide-and-conquer methods (Bandyopadhyay and Mallik 2018), exhaustive enumeration methods (Rodriguez-Baena *et al.* 2011, Serin and Vingron 2011), factorization-based methods (Moran *et al.* 2021), dynamic programming-based methods (Orzechowski *et al.* 2018), and graph-based methods (Shi and Huang 2017, Dong and Yuan 2020, Xie *et al.* 2020). Existing biclustering methods may have the following two issues. Firstly, there is a scarcity of biclustering methods specifically designed for scRNA-seq data, and they cannot effectively handle the inherent challenges posed by the high dimensionality, sparsity, and noise characteristics of scRNA-seq data. Secondly, current biclustering methods often fail to effectively extract key gene clusters and lack proper explanations.

To address the aforementioned issues, we initially explored the integration of deep learning with biclustering in our previous work, scDCABC (Tang and Lan 2024). While scDCABC successfully introduced an iterative framework, it relied on a passive trial-and-error strategy that often-incurred computational redundancy. Building upon this foundation, we present scDBic, an optimized and novel deep learning-based biclustering algorithm. scDBic effectively uncovers hidden patterns in gene expression matrices by iteratively combining deep cell clustering and gene clustering through three synergistic phases: the cell clustering phase, the gene clustering judgment phase, and the gene clustering phase.

In the cell clustering phase, deep neural layers are employed to approximate non-linear transformations, automatically mapping the gene expression matrix to a lower-dimensional latent space. Subsequently, a Shared Nearest Neighbor (SNN) graph is constructed to refine clustering, addressing the challenge of processing high-dimensional data. The gene clustering judgment phase applies a recursive mechanism akin to hierarchical clustering that adapts automatically to the data, effectively solving the issue of determining optimal iteration counts. In the gene clustering phase, key gene clusters are identified by comparing intra-cluster similarity; crucially, all cell information within these clusters is retained, preventing the cell loss typical of traditional biclustering. By capturing local gene-cell co-regulatory patterns instead of relying on global distances, our method exhibits robustness against the "curse of dimensionality" and noise (Xie *et al.* 2016).

2 Materials and methods

The scDBic algorithm for biclustering scRNA-seq data comprises three phases: cell clustering phase, gene clustering judgment phase, and gene clustering phase. The flowchart of this algorithm is presented in Fig. 1. Cell clustering and gene clustering are the core phases that form the body biclustering. The subsequent sections will provide a more detailed description of each phase.

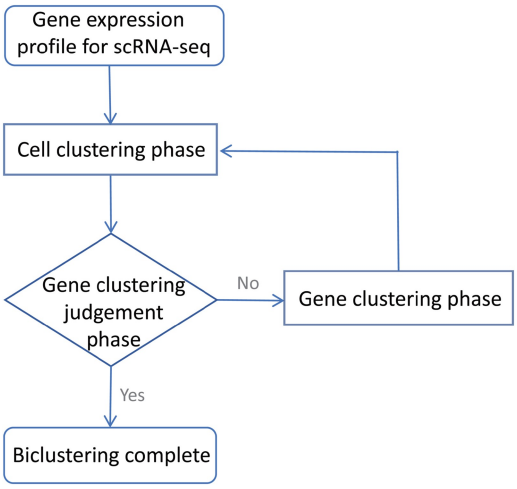

Figure 1 The flowchart of scDBic.

2.1 Cell clustering phase

Cell clustering phase contains four modules: preprocessing module, deep autoencoder module, constructing the SNN graph module, and Walktrap clustering for cells module. The framework for cell clustering is illustrated in Fig. 2.

For clarity, in each iteration, we consistently represent the preprocessed gene expression profile as $X(R \in m * n)$, with the number of cells and genes denoted as n and m, respectively.

2.1.1 Preprocessing module

The data preprocessing is mainly performed by log-normalization and standardization, corresponding to step a in Fig. 2. The log1p function is applied to decrease the skew of gene expression data (Khan *et al.* 2022). Then, Z-Score normalization is employed to normalize gene expression values.

2.1.2 Deep autoencoder module

The deep autoencoder, a specialized neural network for dimensionality reduction, is applied to address the challenges posed by the "curse of dimensionality" in scRNA-seq data (Hinton and Salakhutdinov 2006). It comprises input layer, encoding layer, bottleneck layer, decoding layer, and output layer (Vincent *et al.* 2008), corresponding to step b in Fig. 2.

- **Input layer:** The input layer accepts the preprocessed gene expression profile X with m (the numbers of genes) neurons.
- **Encoding layers:** The encoding layers compress the high-dimensional X into a low-dimensional bottleneck. It consists of three fully connected layers. The numbers of the neurons in each layer are 1024, 512, and 256, respectively, and the activation function in each layer is Scaled Exponential Linear Unit (SELU).
- **Bottleneck layer:** The bottleneck layer is the actual output of the deep autoencoder module, which consists of 128 neurons and uses SELU as the activation function, captures the essence of the gene expression profile. For clarity, in each iteration, the downscaled gene expression profile of the bottleneck layer is denoted as $X_d(R \in 128 * n)$.

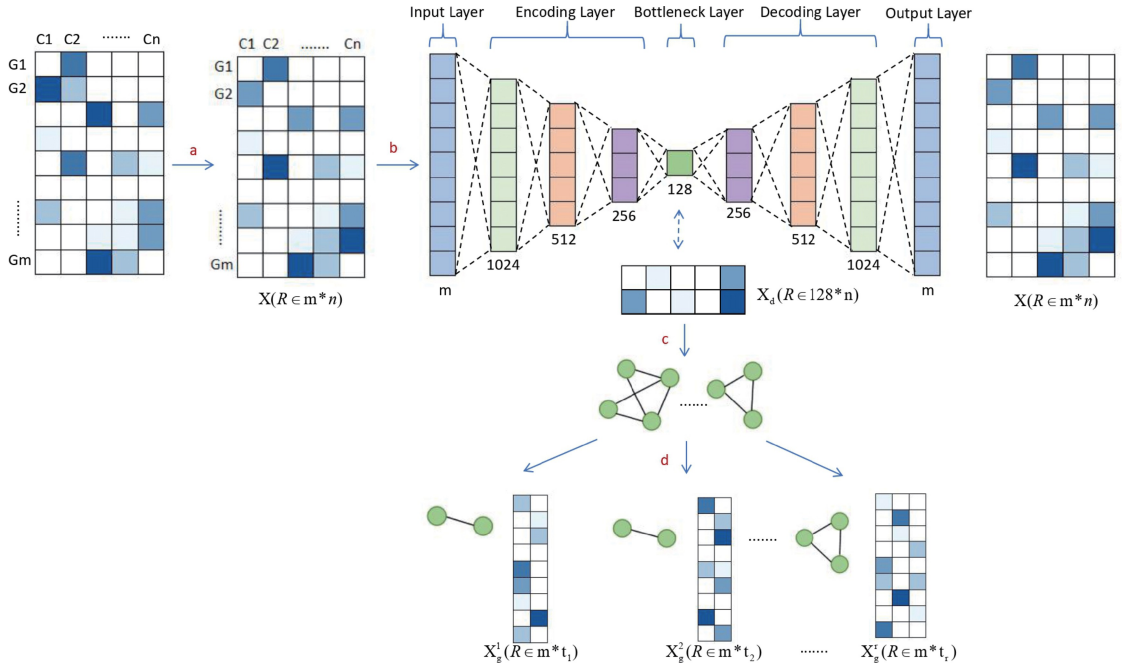

Figure 2 The workflow of cell clustering phase. Step a, b, c, and d respectively represent preprocessing module, deep autoencoder module, constructing the SNN graph module, and Walktrap clustering for cells module.

- **Decoding layers:** Mirroring the encoding layers, the decoding layers reconstruct X_d back as X. This reconstruction is facilitated by three fully connected layers with neuron configurations of 256, 512, and 1024, employing SELU to maintain consistent activation functionality across the network.
- **Output layer:** The output layer obtains the reconstructed X with m neurons, using Sigmoid as the activation function. It is important to note that this is not the final output of the scDBic's deep autoencoder module, but rather the reconstructed approximation of the input.

We employ the scaled cosine error (SCE) loss to enforce directional similarity between the input x_i and reconstructed $\widehat{x}_i$ profiles:

$$L_{SCE} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{x_i \cdot \widehat{x}_i}{\|x_i\|_2 \|\widehat{x}_i\|_2}\right) \tag{1}$$

where N is the batch size, and $\|\cdot\|_2$ denotes the L_2 norm. This formulation prioritizes relative expression patterns over absolute magnitudes. The autoencoder involves two key hyperparameters: the choice of optimizer and number of training epochs. Following extensive hyperparameter tuning experiments (see Section 2 and Chapter 2, available as supplementary data at *Bioinformatics* online), we set RMSprop as the default optimizer and recommend an epoch range between 70 and 150, with 100 epochs used by default.

2.1.3 Constructing the SNN graph module

The SNN graph is a structure constructed based on the counts of nearest neighbors shared among the data points (Xu and Su 2015), corresponding to step c in Fig. 2. In this study, the down-scaled gene expression profile X_d is applied to construct a cell–cell SNN graph $G = (V, E)$, where node V is the set of cells and E is the set of all edges. Given any two cells c_i and c_j, if the Euclidean distance from c_i to c_j ranks among the top l shortest distances from c_i to other cells (with l set to 20 by default, as determined through hyperparameter tuning experiments; see Section 2 and Chapter 2, available as supplementary data at *Bioinformatics* online for details), c_j is deemed the nearest neighbor of c_i. When c_i is the nearest neighbor of c_j and c_j is also the nearest neighbor of c_i, we use an edge to connect cell c_i and c_j. The weight of the edge is calculated using the following formula:

$$W_{i,j} = |T(c_i) \cap T(c_j)| \tag{2}$$

where $T(c_i)$ and $T(c_j)$ represent the nearest neighbors of cells c_i and c_j, respectively.

2.1.4 Walktrap clustering for cells module

The Walktrap clustering method aims to cluster the SNN graph, corresponding to step d in Fig. 2.

Initially, each cell (vertex) in the SNN graph is treated as an independent community, resulting in a starting partition with N communities (where N is the number of cells). The distance between each pair of communities is then calculated to identify the pair with the smallest distance. The distance is defined as:

$$d(c_i, c_j) = \sqrt{\sum_p (P_{i,p}^t - P_{j,p}^t)^2}, \quad P_{i,p}^t = \left(\frac{W_{i,p}}{W_i}\right)^t \tag{3}$$

where $P_{i,p}^t$ and $P_{j,p}^t$ denote the probabilities of reaching community c_p after t steps of a random walk starting from communities c_i and c_j, respectively. Here, $W_{i,p}$ represents the edge weight between community c_i and c_p in the SNN graph, calculated using

the same method as $W_{i,j}$ during SNN construction, and W_i is the total sum of edge weights from community c_i to all other communities.

After identifying the two closest communities, the modularity of the current community structure is computed. The two communities are then merged, and the modularity after merging is recalculated; if the modularity improves, the merge is retained; otherwise, it is discarded. Modularity quantifies the reduction in total random walk distances across the graph, serving as an indicator of the quality of community divisions. The modularity increases between any two communities c_i and c_j is given by:

$$\Delta Q = \frac{1}{2w} \sum_{i,j \in c_i \cup c_j} \left(A_{ij} - \frac{W_i W_j}{2w} \right) \tag{4}$$

where A_{ij} denotes the elements of the adjacency matrix of graph G, W_i and W_j are the total edge weights associated with nodes i and j, respectively, and w is the total edge weight sum of all nodes. A larger ΔQ value indicates that the merged community structure is of higher quality than before.

This merging process continues iteratively until no further merging results in a significant improvement in modularity score. The number of communities after convergence is taken as the final number of clusters K determined by the Walktrap method.

Following the results of Walktrap clustering, we partition the original gene expression matrix in accordance with the Walktrap clustering labels to ensure that all gene information is preserved within each cell cluster. This preparation facilitates the next phase of gene clustering, maintaining the integrity of genetic profiles within each cluster. The partitioned cell clusters set is also the output from this phase, denoted by $X_g = \{X_g^1, X_g^2, \ldots, X_g^r\}$, where r indicates the number of cell clusters formed. It is important to note that each element of X_g will subsequently undergo further gene clustering judgment phase and gene clustering phase. For clarity, we use $X_g^\tau (R \in m * t_\tau)$ as an example in subsequent descriptions, where t_τ represents the number of cells in X_g^τ cluster.

2.2 Gene clustering judgment phase

We design a recursive iterative mechanism similar to hierarchical clustering incorporating an "Explicit Logic Gate." Unlike the passive "compute-first, check-later" strategy employed in previous deep biclustering methods (e.g. scDCABC), where the reverse strategy is indiscriminately applied to all clusters causing resource wastage, scDBic's judgment phase acts as an active gatekeeper. It evaluates the topological necessity of further decomposition before triggering the computationally expensive gene clustering and reverse strategy. For any given $X_g^\tau (R \in m * t_\tau)$, scDBic sequentially checks whether it satisfies the gene clustering judgment condition. If no new cluster is formed ($r = 1$) or the number of newly generated cells is <6 ($t_r < 6$), the condition is considered satisfied. If the condition is not met, the algorithm continues iterating until all elements in X_g satisfy the condition, at which point the scDBic algorithm terminates. The detailed mathematical description is provided in Chapter 3, available as supplementary data at *Bioinformatics* online.

The final output of scDBic includes the cell labels from the biclusters and the key bicluster corresponding to each label. The key bicluster is the result of the last iteration of gene clustering before the judgment condition is satisfied.

2.3 Gene clustering phase

The gene clustering phase corresponds to the "K-means clustering for genes" module and the "Identifying key candidate biclusters via reverse strategy" module, aiming to cluster key genes into the same group and to identify, through a reverse strategy, which group contains the critical gene information. Compared with traditional clustering, the added gene clustering phase can, on the one hand, identify local patterns of gene-cell co-regulation and uncover differential expression of the same gene in different subpopulations; on the other hand, it can eliminate the influence of irrelevant genes on subsequent cell clustering.

2.3.1 K-means clustering for genes

Given a cell cluster, if the judgment condition is not satisfied, this cell cluster can proceed to the gene clustering phase. This phase aims to filter out key genes through clustering, thus enabling more refined sub-cell clusters. Different types of cell clusters can be distinguished using a small subset of genes. These small subsets of genes are called key genes, and identifying these key genes helps identify cell clusters. Therefore, clustering is applied to identify key genes. The process of gene clustering phase is shown in Fig. 3.

For a given cell cluster X_g^τ, we proceed to perform gene clustering using K-means method to derive k gene clusters (with k set to 5 by default, as determined through hyperparameter selection experiments; see Section 2 and Chapter 2, available as supplementary data at *Bioinformatics* online for details). These gene clusters can also be considered as biclusters, denoted as $b = \{b_1, b_2, \ldots, b_5\}$, corresponding to step e in Fig. 3. Not all biclusters contain key genes. Consequently, it is essential to identify the biclusters that contain key genes.

2.3.2 Identifying key candidate biclusters in reverse strategy

We propose a reverse strategy for identifying the key candidate biclusters. Given a bicluster $b_\Gamma \in b$, we apply the cell clustering method proposed in Section Cell Clustering Phase, corresponding to the combination of the cell clustering phase and step f in Fig. 3, to obtain s cell groups. The inner distance (denoted as D) of bicluster b_Γ is calculated as follow:

$$D_{b_\Gamma} = \frac{1}{s} \sum_{g=1}^{s} insim_g \tag{5}$$

where $insim_g$ denotes the inner distance of the gth cell group. Given a cell group g, n^* represents the number of cells within a cell group, m^* represents the number of genes within a cell cluster, and the inner distance (denoted as $insim$) of cell group g is calculated by the following formula:

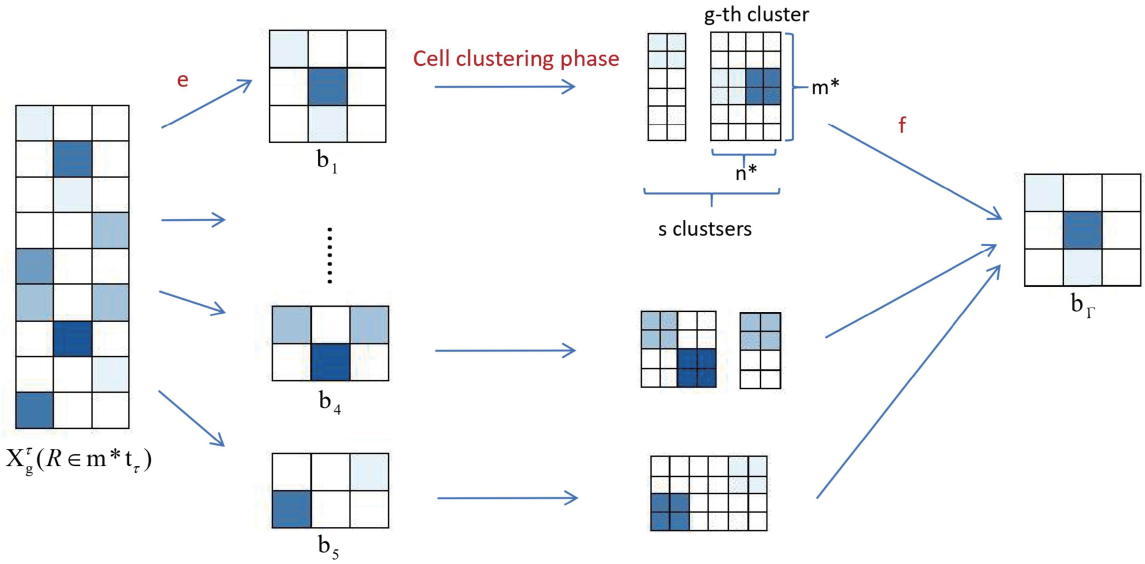

Figure 3 The workflow of gene clustering. Step e represents K-means clustering for genes. Cell clustering phase and step f jointly constitute the process of identifying key candidate biclusters in reverse strategy.

$$insim_g = \frac{1}{n^*}\sum_{i=1}^{n^*}\sum_{j=i+1}^{n^*}\sqrt{\sum_{q=1}^{m^*}(B_{i,g}-B_{j,g})^2} \quad (6)$$

where $B_{i,g}$ and $B_{j,g}$ represent the expression levels of the gth gene in cell i and cell j, respectively.

The smaller the inner distance of a bicluster, the more important that bicluster is considered. By comparing the inner distances of $b_1, b_2, \ldots, b_5$, the key candidate bicluster is identified as the one with the smallest inner bicluster distance, assumed here to be b_Γ. We demonstrate the significance of identifying the key biclusters in the subsequent Section 3.3.

3 Result

3.1 Datasets and evaluation metrics

In this study, we select five well-annotated scRNA-seq datasets: E-MTAB-3321, E-MTAB-2600, GSE45719, GSE52529, and GSE87544. These datasets exhibit diverse scales, comprising between hundreds and ten thousand cells and tens of thousands of genes. To measure the performance of the algorithms, we have adopted the Macro-F1 Score (hereafter referred to as *F1*), Adjusted Rand Index (*ARI*), and Normalized Mutual Information (*NMI*) as evaluation metrics. The detailed formulations of these metrics are provided in Chapter 1, available as supplementary data at *Bioinformatics* online.

3.2 Hyperparameter tuning

Based on the results of the hyperparameter tuning experiments, the default settings for scDBic are as follows: the number of training epochs for the autoencoder is set to 100, the optimizer of the autoencoder is RMSprop, the number of nearest neighbors l is 20, and the value of k in the K-means gene clustering is 5. Detailed results of the hyperparameter tuning experiments can be found in Chapter 2, available as supplementary data at *Bioinformatics* online.

3.3 Deep autoencoder module and identifying key candidate biclusters are two critical steps in clustering scRNA-seq data

As discussed in Section 2.1.2 and Section 2.3.2, the autoencoder mitigates noise by mapping profiles to low-dimensional spaces, while key bicluster identification reveals cluster-specific genes. To validate these components, we compared three variants: “Walktrap” (baseline); “Walktrap + Autoencoder” (iterative autoencoder downscaling without gene clustering); and “Walktrap + Key Biclusters + PCA”. In the third variant, PCA reduces dimensions to 128—matching the autoencoder’s bottleneck—to ensure a fair comparison while utilizing our framework’s gene clustering strategy.

From Table 1, we observe that scDBic algorithm shows a 3.6%–77.7% improvement over the traditional Walktrap algorithm across the *F*1, *ARI*, and *NMI* metrics, which demonstrates the superiority of scDBic algorithm.

From the analysis of lines 4, 8, and 12 in Table 1, the “Walktrap + Key Biclusters + PCA” algorithm improves the results by 2.2%–71.9% compared to the traditional Walktrap method. In addition, the primary difference between scDBic and “Walktrap + Key Biclusters + PCA” lies in their dimensionality reduction techniques of deep autoencoder versus PCA. Among the 12 results from three evaluation metrics across four datasets, scDBic with autoencoder dimensionality reduction outperforms “Walktrap + Key Biclusters + PCA” in 11 results, with improvements ranging from 0.8% to 12.9%. Only the *F*1 score of the E-MTAB-3321 dataset underperformed by 3.1%, underscoring the superior adaptability of the autoencoder to scRNA-seq data.

Table 1 The performance of our method and Walktrap-based methods.[a]

Approaches	E-MTAB-3321	GSE45719	GSE52529	E-MTAB-2600	Metrics
scDBic	0.923	**0.856**	**0.642**	**0.982**	*F1*
Walktrap + Autoencoder	0.899	0.822	0.634	0.973	
Walktrap + Key Biclusters + PCA	**0.954**	0.801	0.634	0.947	
Walktrap	0.454	0.523	0.504	0.841	
scDBic	**0.981**	**0.638**	0.288	**0.958**	*ARI*
Walktrap + Autoencoder	0.971	0.560	**0.325**	0.934	
Walktrap + Key Biclusters + PCA	0.923	0.511	0.263	0.866	
Walktrap	0.204	0.387	0.235	0.639	
scDBic	**0.950**	**0.776**	0.313	**0.925**	*NMI*
Walktrap + Autoencoder	0.928	0.611	**0.345**	0.883	
Walktrap + Key Biclusters + PCA	0.896	0.742	0.299	0.796	
Walktrap	0.267	0.611	0.277	0.650	

[a] Bolding indicates the highest scores.

Similarly, according to lines 3, 7, and 11 in Table 1, the "Walktrap + Autoencoder" algorithm improves the results by up to 76.7% compared to the traditional Walktrap algorithm. Furthermore, comparing the scDBic algorithm to the "Walktrap + Autoencoder" algorithm, scDBic includes a step for identifying key candidate biclusters, while "Walktrap + Autoencoder" algorithm does not. In the GSE52529 dataset, "Walktrap + Autoencoder" performs better by 3.7% in *ARI* and 3.2% in *NMI*. However, scDBic shows improvements ranging from 1% to 16.5% over "Walktrap + Autoencoder" in the other datasets. These results demonstrate that identifying the key candidate biclusters can help improve the accuracy of the algorithm and highlight the necessity of this step.

3.4 Comparing the performance of our algorithm with other clustering and biclustering algorithms

We compared scDBic with six state-of-the-art biclustering methods spanning various principles: graph-based [BiSNN-Walk (Shi and Huang 2017)], information-theoretic [QUBIC2 (Xie *et al.* 2020)], matrix factorization [SSLB (Moran *et al.* 2021)], statistical [GiniClust3 (Dong and Yuan 2020)], common-subsequence [Runibic (Orzechowski *et al.* 2018)], and deep learning [scDCABC (Tang and Lan 2024)]. We also evaluated 21 clustering algorithms categorized into: deep learning [ADClust (Zeng *et al.* 2022), scGNN2.0 (Gu *et al.* 2022), scSSA (Zhao *et al.* 2022), AttentionAE-sc (Li *et al.* 2023), DeepScena (Lei *et al.* 2023), scDECL (Gan *et al.* 2023), scDSC (Gan *et al.* 2022), SCLEGA (Liu *et al.* 2024)]; graph and spectral methods [Seurat (Hao *et al.* 2021), ScGSLC (Li *et al.* 2021a), MPSSC (Park and Zhao 2018)]; consensus and hierarchical clustering [SC3 (Kiselev *et al.* 2017), SCENA (Cui *et al.* 2021), scHFC (Wang *et al.* 2022)]; classical clustering paired with PCA/UMAP; and imputation-assisted methods [GE-Impute (Wu and Zhou 2022) with Seurat]. All methods were implemented using default parameters. A comprehensive summary of all compared methods is presented in Table 2, available as supplementary data at *Bioinformatics* online.

Three evaluation metrics, *F*1, *ARI*, and *NMI*, are employed to assess the clustering performance of these algorithms on five public datasets. The results of *F*1 are shown in Fig. 4, while the results and analysis of *ARI* and *NMI* are provided in Chapter 4, available as supplementary data at *Bioinformatics* online.

In terms of the *F1* metric, scDBic achieves the best performance across four of the five datasets. Specifically, on the E-MTAB-3321 dataset, the improvement range of scDBic is between 0% and 61.3%; for the GSE45719 dataset, the improvement range is between 8.8% and 66.8%; for the GSE52529 dataset, the improvement range is between 1.8% and 52.8%; and on the E-MTAB-2600 dataset, the improvement range is between 1.5% and 77.1%. Notably, since nearly half of the competing algorithms were unable to handle the large-scale GSE87544 dataset, to ensure fairness, the average *F1* scores were calculated based on the other four datasets. On the GSE87544 dataset, scDBic achieves an *F1* score of 0.503, ranking third. From an overall perspective, the average *F1* score of scDBic is 0.851, which is significantly higher than that of the other algorithms, exceeding their average *F1* scores by 0.064 to 0.569, indicating a notable improvement.

Overall, the performance of the scDBic algorithm is superior to that of traditional clustering and biclustering algorithms.

3.5 Category distribution analysis

We visualized the clustering results of scDBic and other baseline methods across all datasets involved in the experiments to evaluate the spatial structure of the clustering. For comparison, the analysis focused on two aspects: clustering boundaries and cell integrity. The results show that scDBic yields clearer and more distinct clustering boundaries than other methods. Moreover, scDBic preserves full cell coverage without the cell loss commonly observed in many biclustering methods, thereby maintaining better cell integrity. Detailed results can be found in Chapter 5, available as supplementary data at *Bioinformatics* online.

3.6 Time complexity analysis of scDBic

The time complexity of scDBic is approximately $O(n^2m)$, with detailed derivation provided in Chapter 6, available as supplementary data at *Bioinformatics* online. The time complexities of all baseline algorithms are summarized in the sixth column of Table 2, available as supplementary data at *Bioinformatics* online. As shown in the

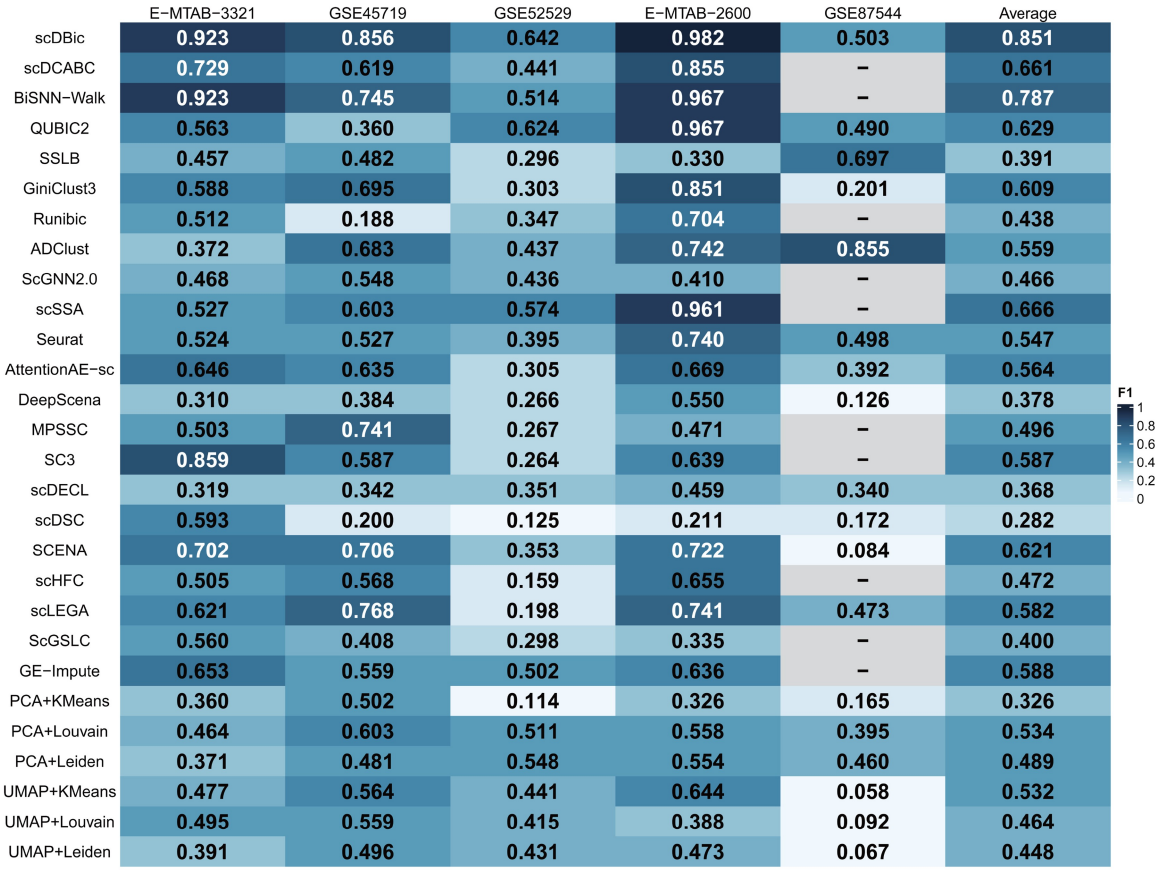

Figure 4 *F1* scores of our algorithm scDBic and 27 state-of-the-art algorithms on five scRNA-Seq datasets.

fourth columns of Table 2, available as supplementary data at *Bioinformatics* online, scDBic ranks at a middle level among all clustering and biclustering methods in terms of runtime and completes the biclustering task within an acceptable timeframe. Combined with its strong accuracy and robustness across multiple datasets, scDBic demonstrates a well-balanced performance in terms of efficiency, stability, and practical applicability.

3.7 The biological significance of biclusters

The biclusters identified by scDBic algorithm not only reveal the cell groups of the tissue but also identify the key genes of cell groups. Here, we analyze E-MTAB-3321 as an example. The analysis of KEGG pathways for the GSE45719, GSE52529, and E-MTAB-2600 datasets is presented in Chapter 7, available as supplementary data at *Bioinformatics* online.

The E-MTAB-3321 dataset comprises 124 mouse embryonic cells and 41 480 genes (Goolam *et al.* 2016). It involves isolation of individual cells from consecutive developmental stages of pre-implantation mouse embryos and sequencing of their transcriptomes using the Smart-seq2 single-cell RNA sequencing protocol across five stages: 2-cell, 4-cell, 8-cell, 16-cell, and 32-cell stages.

Mubeen Goolam's research (Goolam *et al.* 2016) shed light on the significant impact of gene expression heterogeneity on the cellular fate during the 4-cell stage of mouse embryonic development. He observed that the Sox21 gene displayed proximal binding to Oct4 and regulated the expression of the differentiation master regulator Cdx2, exhibiting the strongest heterogeneity during the 4-cell stage. This suggests a pivotal role of the Sox21 gene in orchestrating cellular differentiation during the early stages of embryonic development. Further studies reveal that the expression of Sox21 is regulated by CARM1, which can methylate arginine 26 of histone H3 (H3R26). Therefore, the CARM1 gene is also a key gene in 4-cell stage.

Both the Sox21 and CARM1 genes appear in the representative 4-cell bicluster generated by the scDBic algorithm. Additionally, KEGG pathway enrichment analysis of this bicluster (Fig. 5) shows that the top three significantly enriched pathways—"Neuroactive ligand-receptor interaction," "Cytokine-cytokine receptor interaction," and "MAPK signaling pathway"—are closely related to mouse embryonic development. They jointly regulate critical processes such as cell growth, immune response, and fate determination. These results confirm that scDBic successfully identifies key genes defining specific cell subpopulations.

4 Discussion

In this study, we propose scDBic, a specialized deep learning-based biclustering algorithm tailored for scRNA-seq data analysis. Conceptually, scDBic optimizes the deep learning component by employing a streamlined autoencoder architecture. Compared to complex probabilistic models, our design avoids heavy parameter estimation while effectively extracting key gene expression features. This optimization, combined with the active logic gate and reverse selection strategy, enables scDBic

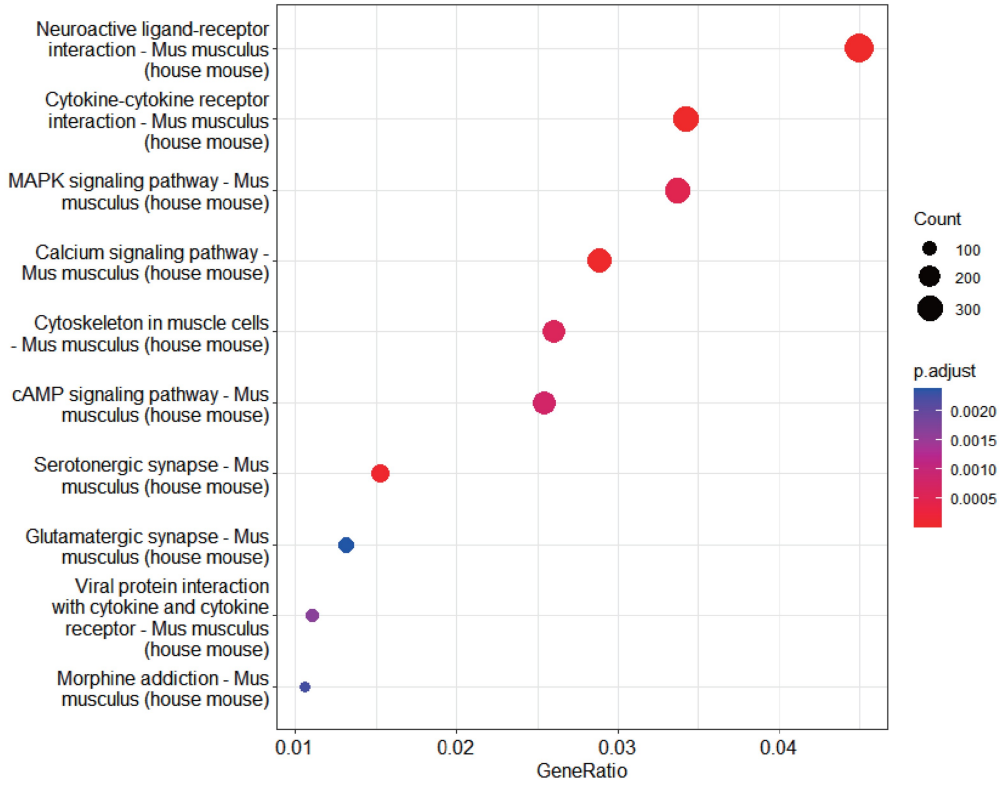

Figure 5 KEGG pathway enrichment results of biclusters identified by scDBic in the E-MTAB-3321 dataset.

to accurately identify representative biclusters. Experimental results across diverse datasets demonstrate that scDBic outperforms six biclustering and twenty-one clustering methods. While our recursive strategy entails a moderate computational cost compared to simple matrix factorization, it offers a superior trade-off by ensuring high clustering accuracy and biological interpretability. Looking ahead, we acknowledge the emergence of Single-cell Large Language Models (scLLMs) which offer powerful universal representations. Future work will explore integrating scLLM-derived embeddings into the scDBic framework to leverage the strengths of both paradigms for uncovering intrinsic patterns in massive transcriptomic datasets.

Author contributions

Xiaoqi Tang (Data curation [lead], Formal analysis [lead], Investigation [equal], Methodology [equal], Software [lead], Visualization [lead], Writing—original draft [equal], Writing—review & editing [equal]), Caihua Liu (Funding acquisition [equal], Project administration [equal], Writing—original draft [equal]), and Chaowang Lan (Conceptualization [lead], Formal analysis [equal], Funding acquisition [equal], Project administration [equal], Software [supporting], Supervision [lead], Writing—review & editing [lead])

Supplementary material

Supplementary material is available at *Bioinformatics* online.

Conflicts of interest

None declared.

Funding

This work was supported and funded by the Guangxi Science and Technology Base and Talent Special Fund [GuikeAD21220099] and the Humanities and Social Sciences Youth Foundation, Ministry of Education of the People's Republic of China [21YJC870009].

Data availability

scDBic is available at https://github.com/Xiaoqi-Tang/scDBic and archived on Zenodo (DOI: 10.5281/zenodo.18676401). Public datasets are accessible via BioStudies (E-MTAB-3321: https://www.ebi.ac.uk/biostudies), Teichmann Lab (E-MTAB-2600: https://espresso.teichlab.sanger.ac.uk), and GEO (GSE45719, GSE52529, GSE87544: https://www.ncbi.nlm.nih.gov/geo).

References

Bandyopadhyay S, Mallik S. Integrating multiple data sources for combinatorial marker discovery: a study in tumorigenesis. *IEEE/ACM Trans Comput Biol Bioinform* 2018;**15**:673–87.

Celiker H, Gore J. Cellular cooperation: insights from microbes. *Trends Cell Biol* 2013;**23**:9–15.

Cui Y, Zhang S, Liang Y *et al.* Consensus clustering of single-cell RNA-seq data by enhancing network affinity. *Brief Bioinform* 2021; **22**:bbab236.

Dong R, Yuan G-C. GiniClust3: a fast and memory-efficient tool for rare cell type identification. *BMC Bioinformatics* 2020;**21**:158.

Eberhardt CS, Kissick HT, Patel MR *et al.* Functional HPV-specific PD-1 + stem-like CD8 T cells in head and neck cancer. *Nature* 2021; **597**:279–84.

Gan Y, Chen Y, Xu G *et al.* Deep enhanced constraint clustering based on contrastive learning for scRNA-seq data. *Brief Bioinform* 2023;**24**:bbad222.

Gan Y, Huang X, Zou G *et al.* Deep structural clustering for single-cell RNA-seq data jointly through autoencoder and graph neural network. *Brief Bioinform* 2022;**23**:bbac018.

Goolam M, Scialdone A, Graham SJ *et al.* Heterogeneity in Oct4 and Sox2 targets biases cell fate in 4-cell mouse embryos. *Cell* 2016; **165**:61–74.

Gu H, Cheng H, Ma A *et al.* scGNN 2.0: a graph neural network tool for imputation and clustering of single-cell RNA-seq data. *Bioinformatics* 2022;**38**:5322–5.

Hao Y, Hao S, Andersen-Nissen E *et al.* Integrated analysis of multimodal single-cell data. *Cell* 2021;**184**:3573–87.e29.

Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. *Science* 2006;**313**:504–7.

Hussain SF, Ramazan M. Biclustering of human cancer microarray data using co-similarity based co-clustering. *Expert Syst Appl* 2016;**55**:520–31.

Khan Y, Hammarström D, Ellefsen S *et al.* Normalization of gene expression data revisited: the three viewpoints of the transcriptome in human skeletal muscle undergoing load-induced hypertrophy and why they matter. *BMC Bioinformatics* 2022;**23**:241.

Kiselev VY, Kirschner K, Schaub MT *et al.* SC3: consensus clustering of single-cell RNA-seq data. *Nat Methods* 2017;**14**:483–6.

Lei T, Chen R, Zhang S *et al.* Self-supervised deep clustering of single-cell RNA-seq data to hierarchically detect rare cell populations. *Brief Bioinform* 2023;**24**:bbad335.

Li J, Jiang W, Han H *et al.* ScGSLC: an unsupervised graph similarity learning framework for single-cell RNA-seq data clustering. *Comput Biol Chem* 2021a;**90**:107415.

Li S, Guo H, Zhang S *et al.* Attention-based deep clustering method for scRNA-seq cell type identification. *PLoS Comput Biol* 2023; **19**:e1011641.

Li S, Zhang Z, Li X *et al.* An iteration model for identifying essential proteins by combining comprehensive ppi network with biological information. *BMC Bioinformatics* 2021b;**22**:430.

Liu Z, Liang Y, Wang G *et al.* scLEGA: an attention-based deep clustering method with a tendency for low expression of genes on single-cell RNA-seq data. *Brief Bioinform* 2024;**25**:bbae371.

Moran GE, Ročková V, George EI. Spike-and-slab lasso biclustering. *Ann Appl Stat* 2021;**15**:148–73.

Orzechowski P, Panszczyk A, Huang X *et al.* runibic: a bioconductor package for parallel row-based biclustering of gene expression data. *Bioinformatics* 2018;**34**:4302–4.

Park S, Zhao H. Spectral clustering based on learning similarity matrix. *Bioinformatics* 2018;**34**:2069–76.

Rodriguez-Baena DS, Perez-Pulido AJ, Aguilar-Ruiz JS. A biclustering algorithm for extracting bit-patterns from binary datasets. *Bioinformatics* 2011;**27**:2738–45.

Rosenberg AB, Roco CM, Muscat RA *et al.* Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* 2018;**360**:176–82.

Satija R, Farrell JA, Gennert D *et al.* Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol* 2015;**33**:495–502.

Serin A, Vingron M. Debi: discovering differentially expressed biclusters using a frequent itemset approach. *Algorithms Mol Biol* 2011;**6**:1–12.

Shi F, Huang H. Identifying cell subpopulations and their genetic drivers from single-cell RNA-seq data using a biclustering approach. *J Comput Biol* 2017;**24**:663–74.

Sun B, Xun Z, Zhang N *et al.* Single-cell rna sequencing in cancer research: discovering novel biomarkers and therapeutic targets for immune checkpoint blockade. *Cancer Cell Int* 2023;**23**:313.

Tang X, Lan C. scDCABC: A deep biclustering method integrating denoising, dimensionality reduction, and clustering for scRNA-seq data. In: *Proceedings of the 2024 13th International Conference on Bioinformatics and Biomedical Science.* 2024, 45–52. Hong Kong, China. New York, NY, USA: Association for Computing Machinery; 2024.

Torres-Jimenez J, Perez-Torres JC. A greedy algorithm to construct covering arrays using a graph representation. *Inf Sci* 2019;**477**:234–45.

Trapnell C, Cacchiarelli D, Grimsby J *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat Biotechnol* 2014;**32**:381–6.

Vincent P, Larochelle H, Bengio Y *et al.* Extracting and composing robust features with denoising autoencoders. In: *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland. New York, NY, USA: Association for Computing Machinery; 2008. p. 1096–1103.

Wang J, Xia J, Tan D *et al.* scHFC: a hybrid fuzzy clustering method for single-cell RNA-seq data optimized by natural computation. *Brief Bioinform* 2022;**23**:bbab588.

Wu X, Zhou Y. GE-Impute: graph embedding-based imputation for single-cell RNA-seq data. *Brief Bioinform* 2022;**23**:bbac313.

Xie J, Girshick R, Farhadi A. Unsupervised deep embedding for clustering analysis. In: *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA. Brookline, MA, USA: PMLR; 2016. p. 478–487.

Xie J, Ma A, Zhang Y *et al.* QUBIC2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale RNA-seq data. *Bioinformatics* 2020;**36**:1143–9.

Xu C, Su Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics* 2015;**31**:1974–80.

Xue Y, Li T, Liu Z *et al.* A new approach for the deep order preserving submatrix problem based on sequential pattern mining. *Int J Mach Learn Cyber* 2018;**9**:263–79.

Zeng Y, Wei Z, Zhong F *et al.* A parameter-free deep embedded clustering method for single-cell RNA-seq data. *Brief Bioinform* 2022;**23**:bbac172.

Zhao J-P, Hou T-S, Su Y *et al.* scSSA: a clustering method for single cell RNA-seq data based on semi-supervised autoencoder. *Methods* 2022;**208**:66–74.

Zheng GXY, Terry JM, Belgrader P *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat Commun* 2017; **8**:14049.

© The Author(s) 2026. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.
Bioinformatics, 2026, 42, 1–9
https://doi.org/10.1093/bioinformatics/btag095
Original Paper