Briefings in Bioinformatics, 2025, **26(1)**, bbae662
https://doi.org/10.1093/bib/bbae662
Problem Solving Protocol

scRGCL: a cell type annotation method for single-cell RNA-seq data using residual graph convolutional neural network with contrastive learning

Lin Yuan[1,2,3], Shengguo Sun[1,2,3], Yufeng Jiang[1,2,3], Qinhu Zhang[4], Lan Ye[5,*], Chun-Hou Zheng[6,*], De-Shuang Huang[4,7,*]

[1]Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), 3501 Daxue Road, 250353, Shandong, China
[2]Shandong Engineering Research Center of Big Data Applied Technology, Faculty of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), 3501 Daxue Road, 250353, Shandong, China
[3]Shandong Provincial Key Laboratory of Industrial Network and Information System Security, Shandong Fundamental Research Center for Computer Science, 3501 Daxue Road, 250353, Shandong, China
[4]Ningbo Institute of Digital Twin, Eastern Institute of Technology, 568 Tongxin Road, 315201, Zhejiang, China
[5]Cancer Center, The Second Hospital of Shandong University, 247 Beiyuan Street, 250033, Shandong, China
[6]Key Lab of Intelligent Computing and Signal Processing of Ministry of Education, School of Artificial Intelligence, Anhui University, 111 Jiulong Road, 230601, Anhui, China
[7]Institute for Regenerative Medicine, Medical Innovation Center and State Key Laboratory of Cardiology, Shanghai East Hospital, School of Life Sciences and Technology, Tongji University, 1239 Siping Road, 200123, Shanghai, China

*Corresponding authors. Lan Ye, Cancer Center, The Second Hospital of Shandong University, 247 Beiyuan Street, 250033, Jinan, Shandong, China. E-mail: sdeyyelan@email.sdu.edu.cn; Chun-Hou Zheng, Key Lab of Intelligent Computing and Signal Processing of Ministry of Education, School of Artificial Intelligence, Anhui University, 111 Jiulong Road, 230601, Hefei, Anhui, China. E-mail: zhengch99@126.com; De-Shuang Huang, Ningbo Institute of Digital Twin, Eastern Institute of Technology, 568 Tongxin Road, 315201, Ningbo, Zhejiang, China; Institute for Regenerative Medicine, Medical Innovation Center and State Key Laboratory of Cardiology, Shanghai East Hospital, School of Life Sciences and Technology, Tongji University, 1239 Siping Road, 200123, Shanghai, China. E-mail: dshuang@eias.ac.cn

Abstract

Cell type annotation is a critical step in analyzing single-cell RNA sequencing (scRNA-seq) data. A large number of deep learning (DL)–based methods have been proposed to annotate cell types of scRNA-seq data and have achieved impressive results. However, there are several limitations to these methods. First, they do not fully exploit cell-to-cell differential features. Second, they are developed based on shallow features and lack of flexibility in integrating high-order features in the data. Finally, the low-dimensional gene features may lead to overfitting in neural networks. To overcome those limitations, we propose a novel DL-based model, cell type annotation of single-cell RNA-seq data using residual graph convolutional neural network with contrastive learning (scRGCL), based on residual graph convolutional neural network and contrastive learning for cell type annotation of single-cell RNA-seq data. scRGCL mainly consists of a residual graph convolutional neural network, contrastive learning, and weight freezing. A residual graph convolutional neural network is utilized to extract complex high-order features from data. Contrastive learning can help the model learn meaningful cell-to-cell differential features. Weight freezing can avoid overfitting and help the model discover the impact of specific gene expression on cell type annotation. To verify the effectiveness of scRGCL, we compared its performance with six methods (three shallow learning algorithms and three state-of-the-art DL-based methods) on eight single-cell benchmark datasets from two species (seven in human and one in mouse). Experimental results not only show that scRGCL outperforms competing methods but also demonstrate the generalizability of scRGCL for cell type annotation. scRGCL is available at https://github.com/nathanyl/scRGCL.

Keywords: cell type annotation; residual graph neural network; contrastive learning; weight freezing; scRNA-seq

Introduction

Single-cell RNA sequencing (scRNA-seq) is a high-throughput technology that can analyze the transcriptomes of individual cells, providing a high-resolution view of cell-to-cell variation [1]. Cell type annotation is a critical step in analyzing scRNA-seq data, as the annotation results have important implications for downstream analyses, such as characterizing cellular heterogeneity and exploring cellular communication [2]. However, manual annotation is time-consuming and subjective. As an alternative, numerous computational methods have been developed for cell type annotation [3].

Recently, many shallow learning algorithms have been proposed to annotate cell types. CHETAH [4], one of the earliest computational methods, used hierarchical classification to link unassigned cells with relevant literature to annotate cell types. The performance of CHETAH depends on the existence of well-annotated reference datasets. SingleR [5], an impressive cell type annotation method, identified the cell type of an individual cell by comparing the similarity of its gene expression pattern with each classifier (samples in reference datasets). However, reference datasets cannot meet the growing demand for cell type annotation. scID [6] used a linear discriminant analysis framework to

Received: September 26, 2024. **Revised:** November 13, 2024. **Accepted:** December 4, 2024

identify cell types in scRNA-seq datasets. scID is only suitable for distinguishing cell types with distinct gene signatures because scID relies on gene signatures. CAEN [7] improved the feature extraction for scRNA-seq data. It encoded the categories using the rank of each gene sequence sample in each class, calculated the correlation coefficients between genes and categories, and identified the genes with the highest coefficient value as marker genes for cell type annotation. However, noise and redundant information in prior knowledge inevitably lead to biased results [8]. Shallow learning algorithms tend to deliver suboptimal results due to the complex structure of scRNA-seq datasets [9].

Large-scale scRNA-seq data provide researchers with unprecedented opportunities to apply deep learning (DL) approaches to annotate cell types. Automated cell type identification using neural networks (ACTINN) [10] utilized a multilayer perceptron (MLP) to extract high-order features and automatically annotated cell types. scDeepSort [11] constructed a cell–gene association graph and used a weighted graph neural network (GNN) for cell type annotation, achieving accurate cell type annotation results without reference datasets. scGraph [12] is a GNN-based method that integrated gene expression and gene interaction networks to overcome technical noise and annotate cell types. scMRA [13] built a knowledge graph to represent the features of cell types from multiple datasets and used a graph convolutional network (GCN) model to annotate cell types. CIForm [14] combined the transformer and patch concept to annotate cell types in large-scale scRNA-seq data. These studies have demonstrated that DL-based methods can effectively model the complex relationships between genes and cells and improve the performance of cell type annotation.

These methods have achieved remarkable results. However, there are several limitations to these methods. First, they do not fully exploit cell-to-cell differential features. Second, they are developed based on shallow features and lack of flexibility in integrating high-order features in the data. Finally, the low-dimensional gene features may lead to overfitting in neural networks.

In this paper, we are interested in integrating the graph convolutional neural network and contrastive learning to guide deep learning architecture to simultaneously learn high-order features and cell-to-cell differential features for cell type annotation. The integration of graph convolutional neural network and contrastive learning makes the model more robust while better distinguishing cell subtypes or cell types with fuzzy boundaries, improving the performance of the model in annotating cell types. To the best of our knowledge, this is the first time that the integration of graph convolutional neural network and contrastive learning has been applied to the task of cell type annotation. Different from the traditional dropout strategy, we add weight freezing to the fully connected (FC) layer to avoid overfitting. The proposed method presents a more flexible neural network architecture and is more effective for cell type annotation. Here, we name the proposed cell type annotation method based on residual graph convolutional neural network and contrastive learning as scRGCL. The structure of the scRGCL framework is presented in Fig. 1. First, scRGCL integrates cell-gene information matrix and gene interaction network information. Second, a graph representation module is applied to the integrated information to extract high-order features in the data. Third, scRGCL uses supervised contrastive learning to obtain important cell-to-cell differential features. Finally, scRGCL combines weighted cross-entropy with contrastive learning loss to assign different loss values to different classes during backward propagation, eliminating the impact of class-imbalanced data and improving the performance of cell type annotation. We compared the performance of scRGCL with six methods (three shallow learning algorithms and three state-of-the-art DL-based methods) on eight single-cell benchmark datasets. Experimental results not only demonstrate that scRGCL outperforms competing methods but also demonstrate the generalization performance of the model for cell type annotation.

Materials and methods

Single-cell RNA sequencing dataset preparation

To evaluate the performance of scRGCL, we used eight scRNA-seq datasets that have been widely used by cell type annotation methods for benchmarking their performances [14]. The eight benchmark datasets contain two species (human and mouse) and five tissues (colon, pancreas, lung, skin, and whole mouse), with cell numbers ranging from thousands to hundreds of thousands. The eight single-cell benchmark datasets are zhangT [15], TM [16], Xin [17], Lung1 [18], Colon [19], Baron Human [20], Lung2 [21], and T [22]. The details of these datasets are listed in Table 1. To benchmark performance on a large-scale mixed dataset containing multiple tissues and cell types, we obtained the Human Cell Landscape (HCL) dataset from the HCL project.

In the preprocessing of data, we first filtered out cells with counts <10 and cells with unclear or discrete cell type annotations. Subsequently, we normalized the cell data by dividing each gene expression value in each cell by the sum of total expression values and then multiplying by a scaling factor of 10e6. Assuming that the single-cell gene expression values follow a negative binomial distribution, we introduced pseudocounts and applied a log2 transformation to each expression value. We used pseudocounts to prevent log transformation of invalid data when the original expression value was zero.

Gene interaction networks

Gene interaction networks within cells influence cell-to-cell differential characteristics [23]. Considering gene interaction networks during modeling is of great value for an in-depth understanding of cell characteristics and accurate annotation of cell types [24]. In this article, the scRGCL model combined gene interactions and utilized the neighbor information of each gene to update cell embedding representations.

The gene interaction networks used by scRGCL were mainly derived from STRINGDB [25], which is a widely used database for retrieving protein–protein interactions (PPIs). It contains data from other databases, experimental data, and results in the literature. In addition, we collected four human and one mouse gene interaction networks to evaluate the performance of scRGCL on various backbone networks. The details of the six networks are summarized in Table 2. HumanNet [26] is a functional gene network that integrated different omics data using a Bayesian statistical framework. We employed two versions of HumanNet, namely, HumanNet-CF and HumanNet-PI, representing cofunctional networks and PPI networks, respectively. FunCoup [27] is a whole-genome functional association network that utilized a unique redundant-weighted Bayesian integration to merge 10 different types of functional association data. GeneMANIA [28] generated gene networks by weighting multiple functional genomics datasets. The mouse gene interaction network was derived from STRINGDB [25].

In our proposed scRGCL model, the gene interaction networks were used as prior information for scRNA-seq datasets. When the

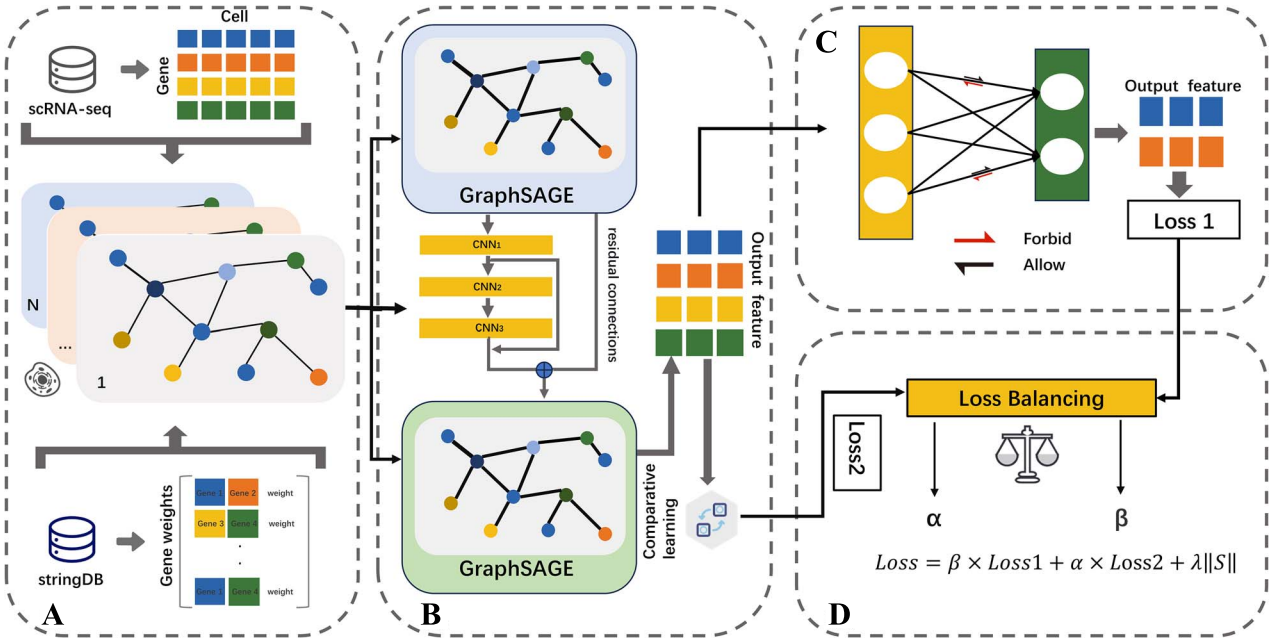

Figure 1. Schematic overview of scRGCL. (A) The input is the scRNA-seq data and gene interaction networks. (B) The residual graph convolutional neural network and contrastive learning. (C) Weight freezing module. (D) Classification module.

Table 1. Overview of datasets used in this study.

Dataset	Tissues	Species	#Genes	#Cells	#Cell types/subtypes	Source
zhangT	Colon	*Homo sapiens*	23 459	8530	20	[15]
TM	Cell Atlas	*Mus musculus*	19 791	54 865	55	[16]
Xin	Pancreas	*H. sapiens*	33 899	1459	4	[17]
Lung1	Lung	*H. sapiens*	29 634	180 069	7	[18]
Colon	Colon	*H. sapiens*	13 538	43 817	5	[19]
Baron Human	Pancreas	*H. sapiens*	17 499	8569	14	[20]
Lung2	Lung	*H. sapiens*	15 148	32 472	17	[21]
T	Skin	*H. sapiens*	16 291	43 817	5	[22]

Table 2. Overview of six gene interaction networks used in this study.

Dataset	Tissues	Species	#Genes
STRINGDB	*H. sapiens*	18 606	11 016 254
HumanNet-CF	*H. sapiens*	14 739	252 590
HumanNet-PI	*H. sapiens*	15 352	158 499
FunCoup	*H. sapiens*	18 081	5 036 826
GeneMANIA	*H. sapiens*	19 551	6 979 630
STRINGDB	*Mus musculus*	21 291	11 944 806

gene interaction network is applied to a single-cell benchmark dataset, only the gene–gene interaction pairs present in benchmark dataset are retained, while the remaining gene–gene interaction pairs in the gene interaction network are discarded. scRGCL constructed undirected gene networks based on gene interaction networks. scRGCL treated the edge from gene *A* to gene *B* as a pair of edges (i.e. an edge from gene *A* to gene *B* and an edge from gene *B* to gene *A*) and additionally assigned each gene an edge pointing to itself. The undirected gene network is conducive to aggregating the information of neighbor genes while retaining the gene's own information according to the weight. This gene network construction method can avoid excessive aggregation of neighbor information, ensuring that the gene's own information is retained and the node features in the network are distinguishable.

The scRGCL framework

scRGCL takes scRNA-seq data and gene interaction networks as input data and uses a deep neural network architecture to automatically annotate cell types. As illustrated in the Fig. 1, scRGCL mainly consists of five modules: (i) data input module; (ii) graph representation module; (iii) contrastive learning; (iv) weight freezing; and (v) classification module. In the data input module, we use scRNA-seq data and gene interaction networks (see sections Single-cell RNA Sequencing Dataset Preparation and Gene Interaction Networks). The graph representation module is utilized to extract complex high-order features from data. GraphSAGE, which includes residual connections and convolutional layers, can solve the problem of vanishing gradient and oversmoothing and improve the model's ability to extract

complex high-order features. Large-scale scRNA-seq data contain massive features and complex relationships between cells. Contrastive learning can help the model learn meaningful cell-to-cell differential features. Weight freezing can avoid overfitting and help the model discover the impact of specific gene expression on cell type annotation. In the classification module, scRGCL combines weighted cross-entropy with contrastive learning loss to assign different loss values to different classes during backward propagation, eliminating the impact of class-imbalanced data and improving the accuracy of cell type annotation. Next, we introduce the graph representation module, contrastive learning, weight freezing, and classification module.

Graph representation module

The graph representation module consists of two GraphSAGEs, a residual convolutional layer, and residual connections between GraphSAGEs. Considering gene interaction networks during modeling is of great value for an in-depth understanding of cell characteristics and accurate annotation of cell types. Compared with other feature extraction methods, the graph convolutional network can make full use of this interaction network to obtain effective cell type annotation information. However, the graph convolutional network may lead to over-smoothing. We introduce residual connections to solve this problem so that the residual graph convolutional network can not only utilize the graph structure information but also increase the network depth to capture more complex high-order features. In this section, we introduce GraphSAGE, residual convolutional layer (RCL), and residual connections between GraphSAGEs.

Graph networks can be used to model the interactions between genes. A graph represents a cell, nodes represent genes, and edges represent the relationships. We adopt a modified GraphSAGE [29] to update each node by aggregating information from neighbor nodes.

$$h_v^{k+1} \leftarrow \sigma\left(\mathrm{W}\cdot \mathrm{MEAN}\left(\left\{h_v^k\right\} \cup \left\{h_u^k, \forall u \in N(v)\right\} \delta\left(S_v\right)\right)\right) \tag{1}$$

where h_v^k represents k-th layer feature vector of node v, $N(v)$ represents the neighbor nodes of node v, $\sigma(\cdot)$ is the nonlinear activation function and W represents the trainable parameters, S_v is the edge importance score vector, and δ is a sigmoid function. Each gene is embedded as an 8D feature.

A residual network can pass input information directly to output through skip connections, addressing the issues of vanishing gradient and oversmoothing and improving model performance. Assuming x and y represent input and output, respectively, the general forward propagation formula for a residual layer [30] can be expressed as:

$$y = \mathsf{F}(x, w_i) + x \tag{2}$$

where $\mathsf{F}(x, w_i)$ represents the residual mapping. In this study, the residual convolutional layer is mainly composed of four convolutional neural network (CNN) layers:

$$\begin{cases} h_1 = \mathrm{CNN}_1(x) \\ h_2 = \mathrm{Dropout}\left(h_1\right) \\ h_3 = \mathrm{CNN}_2\left(h_2\right) \\ h_4 = \mathrm{Dropout}\left(h_3\right) \\ h_5 = \mathrm{CNN}_3\left(h_4\right) \\ h_6 = h_5 + h_1 \\ y = \mathrm{CNN}_4\left(h_6\right) \end{cases} \tag{3}$$

In order to obtain accurate node feature representation while avoiding vanishing gradient and oversmoothing, we use the residual convolutional layer to connect two GraphSAGE [31]. Assuming that the input and output of the network are $\hat{x}$ and $\hat{y}$, respectively, forward propagation can be defined as:

$$\begin{cases} f_1 = \mathrm{GraphSAGE}_1\left(\hat{x}\right) \\ f_2 = \mathrm{RCL}\left(\mathrm{Dropout}\left(f_1\right)\right) \\ f_3 = \mathrm{FC}\left(f_1 + f_2\right) \\ \hat{y} = \mathrm{GraphSAGE}_2\left(\mathrm{Dropout}\left(f_3\right)\right) \end{cases} \tag{4}$$

where $\mathrm{RCL}(\cdot)$ is the residual convolutional layer, $\mathrm{FC}(\cdot)$ is the fully connected layer, and $f_3 = \mathrm{FC}\left(f_1 + f_2\right)$ represents the residual connection between GraphSAGEs.

The graph network of the graph representation module can learn the primary features of scRNA-seq data, while the residual module can learn high-order and complex features. The graph convolutional neural network with residual connections has powerful feature representation capability.

Contrastive learning

Contrastive learning obtains anchor samples by sampling multiple positive and negative samples and then uses the loss function to guide positive samples to approach the anchor samples while pushing negative samples away and finally helping the model learn features that are common to the same type of cells but not shared by other types of cells. Contrastive learning can help the model more accurately discover subtle differences between cells and obtain feature representations with more biological meanings. Contrastive learning compares the differences between different cell types, making the model more sensitive to the gene expression data of cells under different biological conditions, helping the model to identify cell subtypes. Models containing contrastive learning not only learn the characteristics of samples but also focus on the relative relationship between samples, thereby optimizing the classification boundaries, making the features learned by the model richer and the model more robust.

We adopt the SupCon [32] loss function, which helps the model learn meaningful and discriminative feature representations. The loss function is defined as follows:

$$L_{CON} = -\sum_{i=1}^{B} \frac{1}{M_{y_i} - 1} \sum_{j=1}^{B} l_{i \neq j} l_{y_i = y_j} \ln \left[\frac{\exp\left(s_{i,j}/t\right)}{\exp\left(s_{i,j}/t\right) + \sum_{k=1}^{M} l_{y_i \neq y_k} \exp\left(s_{i,k}/t\right)} \right] \tag{5}$$

where M represents a mini-batch size, y_i and y_j represent the label of the anchor sample i and the sample j, respectively. M_{y_i} represents the number of cells labeled type Y. $l_{i \neq j}$, $l_{y_i = y_j}$, and $l_{y_i \neq y_j}$ are indicator functions. $l_{i \neq j} = 1$ if $i \neq j$ and $l_{i \neq j} = 0$ otherwise. $l_{y_i = y_j}$ and $l_{y_i \neq y_j}$ have the same situation.

$$s_{i,j} = v_i^T v_j / \|v_i\| \|v_j\| \tag{6}$$

where $s_{i,j}$ is cosine similarity between sample i and j, and v_i and v_j represent the high-order feature vectors of the sample i and j, respectively.

Weight freezing

The dense connections of the FC layer increase the overfitting probability when using low-dimensional feature vectors of genes.

Studies [33, 34] have shown that gene expression patterns are cell-type specific, and specific cell types are closely associated with expression patterns of specific genes. Instead of using low-dimensional feature vectors of all genes, the model needs to focus on low-dimensional feature vectors of specific genes that have an important influence on cell type annotation. Using FC layers means that each neuron will affect the result of cell type annotation. However, this may not be consistent with the prior knowledge of cell type annotation. In this section, we introduce weight freezing to replace the dense connections with sparse connections to improve the influence of key genes on cell type annotation. Sparse connections are more consistent with the characteristics of gene expression data in scRNA-seq data and help the model more accurately simulate the relationship between gene expression and cell type annotation.

Weight freezing does not change the forward propagation of the FC layer but freezes some learnable parameters during the backward propagation [35]:

$$W_n = W_n - M \odot \left(\eta \cdot \left(\tilde{z}_n - z_n\right) x_n^T\right) \tag{7}$$

where W_n is a learnable parameter that is iteratively adjusted based on the gradient calculated by the back-propagation algorithm, M is a mask matrix, $\tilde{z}_n$ represents the predicted value, z_n represents the target value, η is the learning rate of the optimizer, x_n^T represents the input vector of the FC layer, and $\odot$ denotes element-wise multiplication.

Weight freezing and dropout are both regularization methods to prevent neural networks from overfitting. Dropout affects both forward and backward propagation, while weight freezing only affects backward propagation. Neurons affected by dropout lose all decision-making ability, but neurons affected by weight freezing only lose part of the decision-making ability. Weight freezing masks specific parameters instead of directly changing neurons to create sparse connections like dropout.

Classification module

The cross-entropy loss is used for training:

$$L_{CE} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{m=1}^{M}\left[y_n^m \cdot \log x_n^m + \left(1 - y_n^m\right) \cdot \log\left(1 - x_n^m\right)\right] \tag{8}$$

where N and M are the number of samples and cell types, respectively. y_n^m is 1 if the n-th sample belongs to the m-th cell type and 0 otherwise. x_n^m is the predicted probability of the n-th sample belonging to the m-th cell type.

The edge importance score vector S is also added to the final loss function. After stabilizing the model using R-DROP [36], we perform joint optimization using the contrastive learning loss. α represents the weighting factor of the contrastive learning loss, and L_{CON} represents the loss function of contrastive learning. The final loss function is defined as follows:

$$L = L_{CE} + \alpha L_{CON} + \lambda \|S\| \tag{9}$$

We use the Adam [37] optimizer with an initial learning rate of 0.01 and a weight decay of 10e-5. First, we initialize the model weights using cosine annealing with warm restart [38]. Then, when the F1 metric stops improving, the model is trained with a strategy of reducing the learning rate by a factor of 0.1.

Evaluation strategies

The two methods (k-fold cross-validation and independent dataset testing) are used to evaluate the model performance. Four evaluation metrics are used to assess the model, namely, the AUROC (area under the ROC curve), AUPRC (area under the precision–recall curve), ACC (accuracy), and F1-Score. These evaluation metrics are defined as follows:

$$\begin{cases} \mathrm{TPR} = \dfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\ \mathrm{FPR} = \dfrac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} \\ \mathrm{Precision} = \dfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \\ \mathrm{Recall} = \dfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \\ \mathrm{ACC} = \dfrac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \\ \mathrm{F1} = \dfrac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \end{cases} \tag{10}$$

We compared scRGCL with CHETAH [4], SingleR [5], scID [6], ACTINN [10], scGraph [12], and CIForm [14] to test the performance of scRGCL. These methods include three shallow learning algorithms (CHETAH, SingleR, scID) and three state-of-the-art DL-based methods (ACTINN, scGraph, CIForm). CHETAH (2019) is one of the earliest computational methods that annotated cell types by linking queried scRNA-seq data with relevant literature. SingleR (2019) identified the cell type of a single cell by comparing the similarity of its gene expression pattern with each classifier. scID (2020) used a linear discriminant analysis framework to identify cell types. ACTINN (2020) utilized MLP to extract high-order features and automatically annotated cell types. scGraph (2022) used the GNN to annotate cell types combined with gene expression and gene interaction information. CIForm (2023) combined transformer and patch concept to annotate cell types. All experiments were done on an NVIDIA RTX 3090 GPU. To avoid overfitting, we applied the R-drop regularization technique, set the dropout to 0.3, the batch size to 64 and the learning rate to 0.001, and used Adam as the optimizer for the model.

Results

Ablation experiments

To investigate the impact of key modules on scRGCL, we constructed four variants of scRGCL, (i) (w/o) RGNN; (ii) (w/o) Contrastive Learning; (iii) (w/o) Weight Freezing; (iv) Weight Freezing- > Drop out. (w/o) RGNN represents that scRGCL does not use the residual graph neural network and contains a layer of GraphSAGE. (w/o) Contrastive Learning represents that scRGCL does not contain contrastive learning. (w/o) Weight Freezing represents that scRGCL does not contain the Weight Freezing. Weight Freezing- > Drop out replaces the Weight Freezing module with drop out. We trained the four variant models using the same hyperparameters and compared the performance of scRGCL with these models using eight single-cell benchmark datasets.

As shown in Fig. 2A, compared with scRGCL, the F1 and ACC of (w/o) RGNN, (w/o) Contrastive Learning, (w/o) Weight Freezing, and Weight Freezing- > Drop out decreased by 3.24% and 3.89%, 1.08% and 1.40%, 0.83% and 0.56%, and 1.30% and 0.94%, respectively. Experimental results show that residual graph neural network, contrastive learning module, and Weight Freezing module all contribute to cell type annotation.

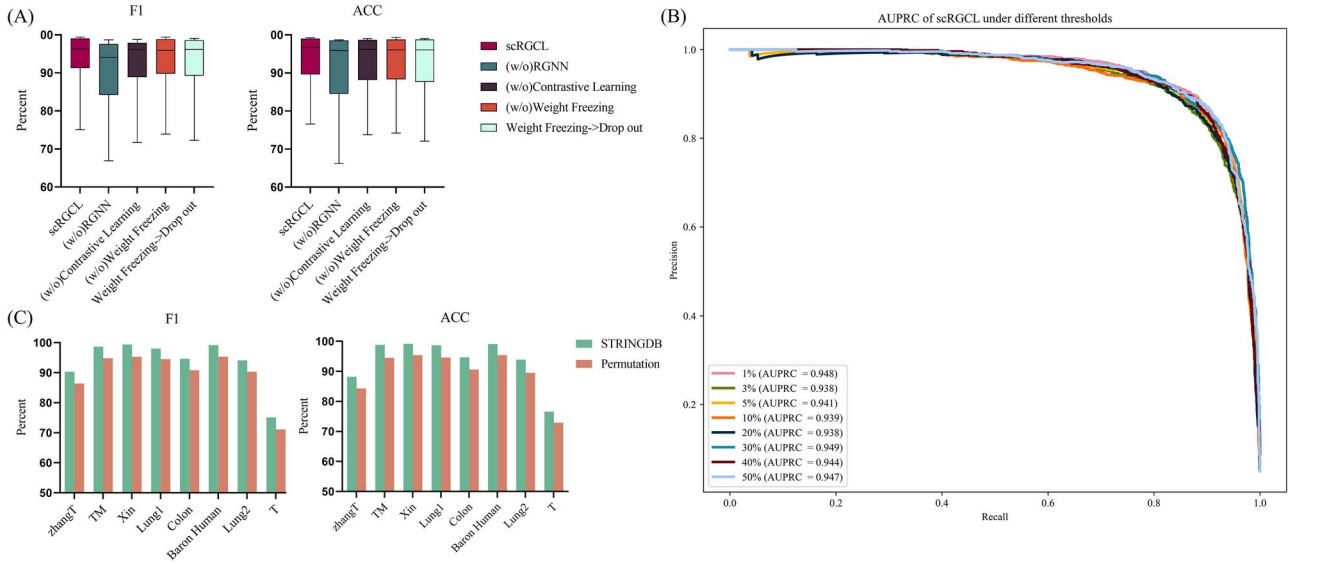

Figure 2. (A) Ablation experiment results. (B) AUPRCs of scRGCL in gene interaction networks at different thresholds. (C) Comparison of F1 and ACC of scRGCL under a real gene interaction network and a randomly generated gene interaction network.

The effects of gene interaction network on scRGCL

scRGCL incorporates the gene interaction network to model gene expression in cells, leveraging the neighbor information of each gene to update cell embedding representations. In this section, we investigated the effects of gene interaction networks on scRGCL at different thresholds. First, we adopted the threshold strategy of scGraph [12] and systematically filtered the STRINGDB PPI network using eight different thresholds, retaining the top 1%, 3%, 5%, 10%, 20%, 30%, 40%, and 50% of the gene interaction pairs with the highest scores. Subsequently, we evaluated the performance of scRGCL on the zhangT using eight STRINGDB backbone networks.

As shown in Fig. 2B and Supplementary Table 1, when the threshold exceeds 10%, the computation time of the model increases dramatically while its performance barely changes. This suggests that although the model uses more gene interaction information, low-scoring interaction pairs may contain less useful information and provide limited help for cell annotation. The F1, ACC, AUROC, and AUPRC of scRGCL using different STRINGDB backbone networks on the zhangT dataset are listed in Supplementary Table 2. Next, we evaluated the performance of scRGCL on seven human datasets (zhangT, Xin, Lung1, Colon, Baron Human, Lung2, and T) using 1%, 3%, and 5% of the STRINGDB network to determine the optimal threshold. As shown in Table 3, the maximum and minimum values of F1 and ACC differ by 0.005 and 0.004, respectively. This indicates that scRGCL exhibits robust performance in gene interaction networks at different thresholds. Considering that the top 1% network is the most condensed network, we selected the top 1% network as the gene interaction network of scRGCL.

Furthermore, we used five human gene interaction networks (STRINGDB, HumanNet-CF, HumanNet-PI, FunCoup, and GeneMANIA) to evaluate the performance of scRGCL on different gene interaction networks. FunCoup and GeneMANIA retained the top 1% interaction pairs to construct the gene interaction network due to the presence of numerous ambiguous edges. The experimental results are listed in Table 4. Although these networks differ in functional types, the number of nodes, and edges, scRGCL performs essentially the same in terms of F1 and ACC. The experimental results show that scRGCL exhibits strong robustness across different gene interaction networks. STRINGDB is a widely used important tool for studying gene interaction networks, which contains experimental data, results from PubMed, data from other databases, and results predicted by bioinformatics methods. Compared with other gene interaction networks (HumanNet-CF, HumanNet-PI, FunCoup, and GeneMANIA), the gene interaction network from STRINGDB contains more accurate and comprehensive information. In future research, we recommend using the STRINGDB gene interaction network, and researchers can evaluate the performance of their proposed model using 1%, 3%, and 5% of the STRINGDB network to determine the optimal threshold.

To verify the effectiveness of a real gene interaction network, we also evaluated the performance of scRGCL containing randomly generated gene interaction networks. We randomly generated 10 networks and calculated the average F1 and average ACC of scRGCL containing these networks on each dataset. As shown in Fig. 2C, compared with scRGCL containing a real gene interaction network, the F1 and ACC of scRGCL containing a randomly generated network decreased by 0.039 and 0.040, respectively. The results are listed in Supplementary Table 3. The experimental result demonstrated the effectiveness of the real gene interaction network.

The impact of contrastive learning on model

PCA (principal component analysis) [39], LSTM (long short-term memory) [40], and AutoEncoder [41] are widely used feature learning methods. To compare the performance of contrastive learning and these feature learning methods, we constructed three variants of scRGCL, specifically, (i) CL- > PCA; (ii) CL- > LSTM; and (iii) CL- > AutoEncoder. CL- > PCA represents replacing contrastive learning with PCA. CL- > LSTM represents replacing contrastive learning with LSTM. CL- > AutoEncoder represents replacing contrastive learning with AutoEncoder. We trained the three variant models using the same hyperparameters and compared the performance of scRGCL with these models using eight single-cell benchmark datasets. As shown in Fig. 3A and Supplementary Table 4, in terms of ACC and F1, scRGCL outperforms the three variants on all datasets. Experimental

Table 3. F1s and ACCs of scRGCL on seven human datasets using 1%, 3%, and 5% of the STRINGDB network.

Methods	zhangT	Xin	Lung1	Lung2	Colon	Baron Human	T	Average
				F1				
STRINGDB-1%	0.903	0.994	0.980	0.941	0.946	0.992	0.751	0.930
STRINGDB-3%	0.896	0.988	0.978	0.938	0.940	0.990	0.747	0.925
STRINGDB-5%	0.901	0.992	0.981	0.936	0.945	0.983	0.750	0.927
				ACC				
STRINGDB-1%	0.882	0.992	0.987	0.939	0.947	0.991	0.766	0.929
STRINGDB-3%	0.877	0.986	0.982	0.937	0.943	0.990	0.762	0.925
STRINGDB-5%	0.880	0.992	0.984	0.928	0.946	0.983	0.764	0.925

Table 4. F1s and ACCs of scRGCL on seven human datasets containing five human gene interaction networks.

Backbones	zhangT	Xin	Lung1	Lung2	Colon	Baron Human	T	Average
				F1				
STRINGDB	0.903	0.994	0.980	0.941	0.946	0.992	0.751	0.930
HumanNet-CF	0.894	0.986	0.967	0.934	0.939	0.987	0.747	0.922
HumanNet-PI	0.898	0.989	0.973	0.937	0.934	0.977	0.738	0.921
FunCoup	0.901	0.993	0.984	0.938	0.942	0.984	0.749	0.927
GeneMANIA	0.904	0.981	0.978	0.940	0.948	0.993	0.734	0.925
				ACC				
STRINGDB	0.882	0.992	0.987	0.939	0.947	0.991	0.766	0.929
HumanNet-CF	0.870	0.985	0.981	0.933	0.931	0.986	0.741	0.918
HumanNet-PI	0.876	0.983	0.984	0.935	0.936	0.987	0.747	0.921
FunCoup	0.885	0.990	0.979	0.931	0.944	0.992	0.762	0.926
GeneMANIA	0.883	0.976	0.989	0.938	0.946	0.989	0.767	0.927

results show that contrastive learning can help scRGCL learn meaningful and discriminative feature representations. PCA is a linear dimensionality reduction method. Using PCA for feature learning in the model cannot fully capture the complex nonlinear relationships in the data, resulting in poor cell classification results. The performance of LSTM depends on the sequential input of data and cannot effectively process high-dimensional sparse gene expression data in cells. The AutoEncoder may focus on the common features in cells when reconstructing data while ignoring the subtle differences and features of type-specific cells that are important in the classification task.

To study the impact of contrastive learning on models when using mixed datasets, we collected two human datasets 10X_3-rep1 and Scipio-rep2 from the literature [42] (https://cellxgene.cziscience.com/collections/398e34a9-8736-4b27-a9a7-31a47a67f446). 10X_3-rep1 contains 7750 cells, 21 231 genes, and 10 cell types, and Scipio-rep2 contains 4425 cells, 14 912 genes, and 10 cell types. After preprocessing the two datasets, we performed gene alignment (obtaining the common genes in the two datasets) and then mixed the two datasets. We then compared the performance of scRGCL and three variant models and six competing methods on the mixed dataset. As shown in Fig. 3B and Supplementary Table 5, scRGCL outperforms the three variant models and six competing methods. Experimental results show that contrastive learning enhances feature representation and cell type separation across various datasets.

The impact of noise or sparse data on model robustness

To evaluate the performance of scRGCL on noise or sparse data, we constructed five types of data based on the eight benchmark datasets. Specifically, (i) GaussianNoise (10%); (ii) GaussianNoise (20%); (iii) Sparse(10%); (iv) Sparse(30%); (v) Sparse(50%). GaussianNoise(10%) represents that in each benchmark dataset, we randomly selected 10% of the genes in each cell and added Gaussian noise to generate noise data. GaussianNoise(20%) represents that 20% of the genes in each cell were randomly selected. Sparse(10%) represents that in each benchmark dataset, we randomly selected 10% of the genes in each cell and set their expression values to zero. Sparse(30%) and Sparse(50%) indicate that 30% and 50% of the genes in each cell were randomly selected, respectively. We compared the performance of scRGCL on the benchmark datasets and these five types of datasets. As shown in Fig. 3C and Supplementary Table 6, scRGCL showed a slight yet nonsignificant performance decrease. Experimental results demonstrated that scRGCL is robust to noise and sparse data.

Performance comparison of methods for annotating cell types

In this section, we compared the performance of scRGCL with CHETAH [4], SingleR [5], scID [6], ACTINN [10], scGraph [12], and CIForm [14]. The seven methods were tested on eight single-cell benchmark datasets (zhangT [15], TM [16], Xin [17], Lung1 [18], Colon [19], Baron Human [20], Lung2 [21], and T [22]) from two species (seven in human and one in mouse). We calculated the F1 and ACC of these methods on the eight datasets using five-fold cross-validation.

As shown in Fig. 4, scRGCL outperforms competing methods on all datasets. CIForm and scGraph are the two best-performing methods among the six competing methods. The detailed F1s and ACCs are summarized in Table 5. The average F1 of scRGCL is 0.937, and the average F1 of CIForm and scGraph are 0.896

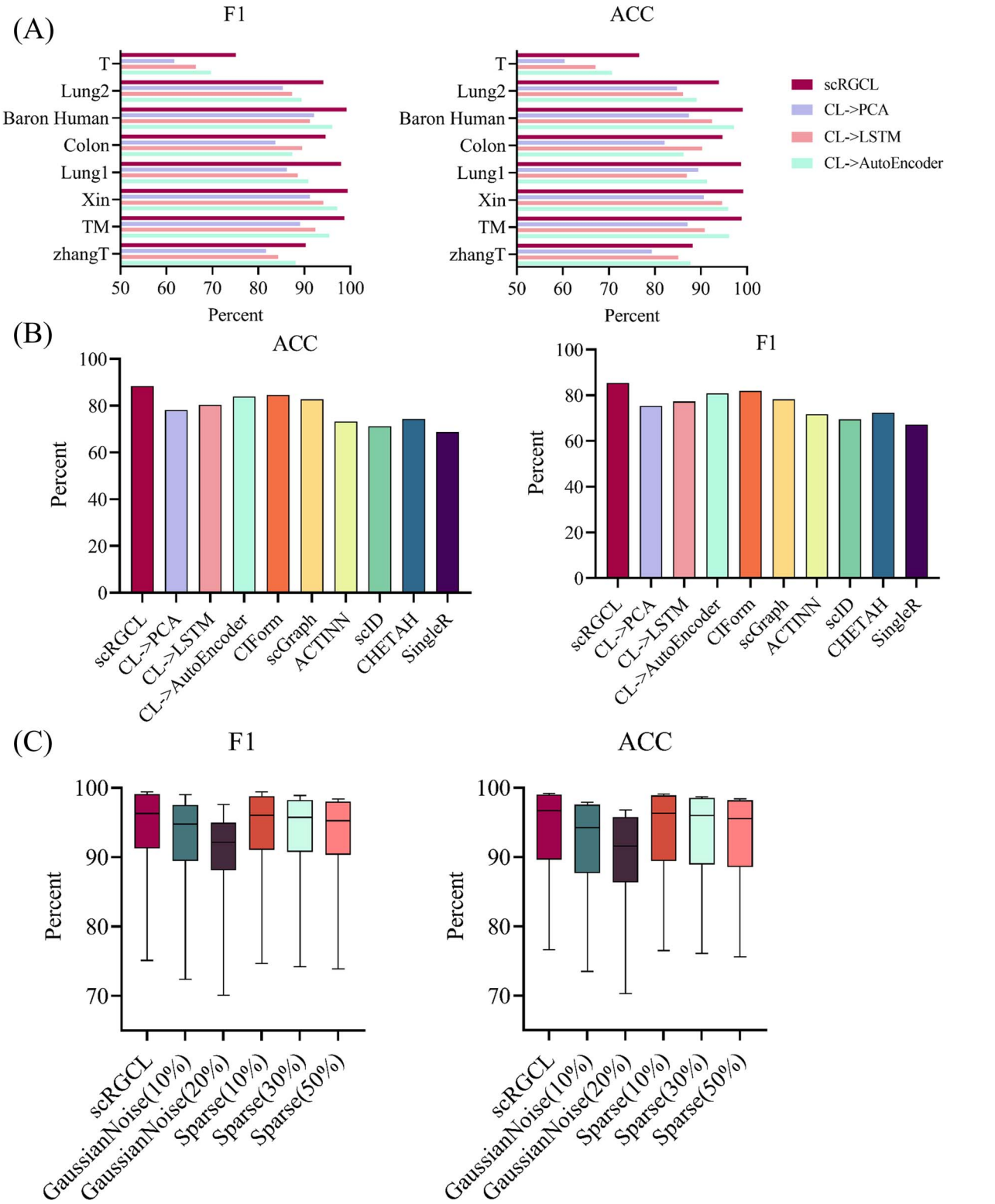

Figure 3. (A) F1s and ACCs of SCRGCL and three variant models (CL- > PCA, CL- > LSTM, CL- > AutoEncoder) on eight single-cell benchmark datasets. (B) Performance comparison of scRGCL with three variant models and six competing methods on a mixed dataset. (C) F1s and ACCs of scRGCL on original data, noise data, and sparse data.

and 0.875, respectively. The average ACC of scRGCL is 0.937, and the average ACC of CIForm and scGraph are 0.898 and 0.891, respectively. Compared with CIForm and scGraph, scRGCL improves F1 by 0.041 and 0.062, and ACC by 0.039 and 0.046, respectively. In the T dataset containing five cell types, scRGCL improved F1 and ACC by 0.090 and 0.095, respectively. It should be noted that scRGCL can also effectively annotate cell types on small-scale datasets containing multiple cell types. For instance, scRGCL outperformed CIForm and scGraph in the zhangT dataset containing 20 T-cell subtypes. The F1 of scRGCL is 0.903, exceeding

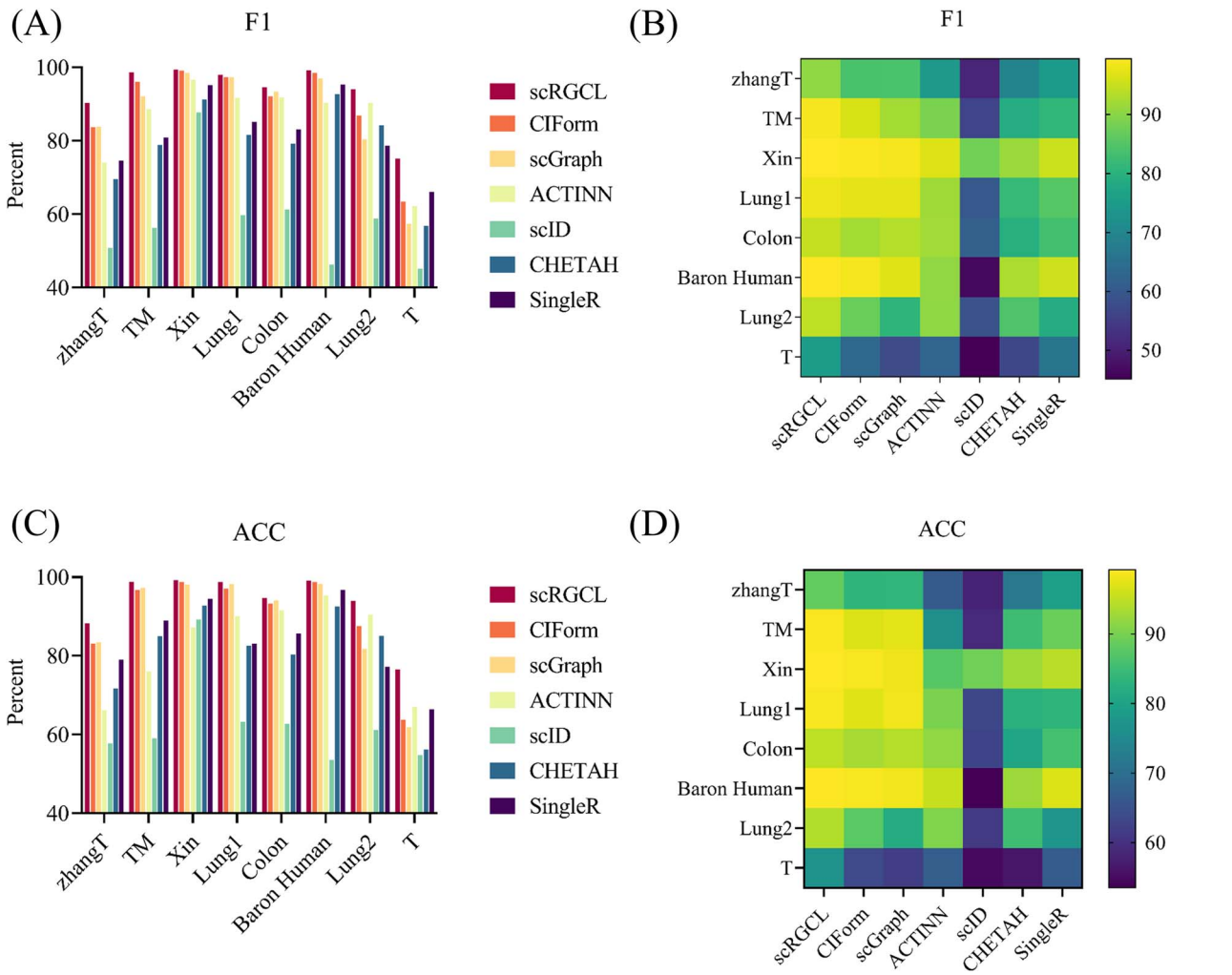

Figure 4. (A) Comparison of F1 between scRGCL and six competing methods on eight single-cell benchmark datasets. (B) F1 heatmap of scRGCL and six competing methods on eight single-cell benchmark datasets. (C) Comparison of ACC between scRGCL and six competing methods on eight single-cell benchmark datasets. (D) ACC heatmap of scRGCL and six competing methods on eight single-cell benchmark datasets.

the 0.837 and 0.839 of the two best models (CIForm and scGraph). Similarly, the ACC of scRGCL is 0.882, while the ACC of the best two models (CIForm and scGraph) are 0.831 and 0.834, respectively. Figure 5 shows that scRGCL accurately annotates twenty cell types in the zhangT dataset.

The comparison methods have their own strengths and weaknesses for cell type annotation. Specifically, CIForm uses the traditional Transformer, which has strong cell annotation capability and is robust to batch effects of datasets. However, CIForm does not fully utilize the association information between genes, cannot explain the biological significance of gene embedding in the encoder, and is sensitive to noise in high-dimensional sparse scRNA-seq data. scGraph effectively utilizes the interaction relationship of genes through graph neural networks, improves the interpretability of cell type annotation results, and is robust to batch effects of datasets. However, over-smoothing is prone to occur as the model depth increases, and the model cannot effectively learn high-order information between genes, making it difficult to identify cell subtypes. ACTINN automatically learns features in the data, reducing manual feature selection and making it suitable for the analysis of larger datasets. However, ACTINN is insensitive to subtle differences in gene expression and may have poor identification of rare cell types. scID can integrate multiple data to provide a more comprehensive analysis of cell characteristics and reduce dependence on reference datasets. However, scID is only suitable for distinguishing cell types with distinct gene signatures. CHETAH supports in-depth analysis of complex cell populations and provides accurate cell annotation by comparing with reference gene expression data. However, the performance of CHETAH depends on the existence of well-annotated reference datasets. SingleR does not require predefined cell type labels and can automatically perform cell annotation based on reference datasets. However, SingleR is highly dependent on accurate and comprehensive reference datasets. The reference datasets cannot meet the growing demand for cell type annotation.

To verify whether the annotation results of scRGCL are consistent with biological findings, we compared the true labels of the Colon and Baron Human datasets with the annotation results of scRGCL, CIForm, and scGraph. We used scRGCL to extract high-order features of each single cell and applied UMAP for dimensionality reduction and visualization. As shown in Fig. 6, compared with the real label UMAP graph (Fig. 6A), scRGCL did not identify the very few B cells and CD4 T cells next to the Myeloid cells. scRGCL accurately identified Myeloid cells dispersed among B cells and also accurately annotated complex cell types such as CD4 T Cell, CD8 T Cell, and ILC. CIForm annotated almost all Myeloid cells dispersed in B cells as B cells and has good

Table 5. F1s and ACCs of scRGCL and six competing methods on eight single-cell benchmark datasets.

Methods	zhangT	TM	Xin	Lung1	Colon	Baron Human	Lung2	T	Average
					F1				
scRGCL	0.903	0.987	0.994	0.980	0.946	0.992	0.941	0.751	0.937
CIForm	0.837	0.961	0.991	0.973	0.921	0.985	0.869	0.634	0.896
scGraph	0.839	0.921	0.985	0.973	0.934	0.969	0.804	0.574	0.875
ACTINN	0.741	0.886	0.966	0.917	0.919	0.904	0.903	0.622	0.857
scID	0.508	0.563	0.877	0.597	0.613	0.463	0.588	0.451	0.583
CHETAH	0.695	0.789	0.913	0.816	0.792	0.927	0.842	0.568	0.793
SingleR	0.746	0.809	0.952	0.852	0.831	0.953	0.787	0.661	0.824
					ACC				
scRGCL	0.882	0.988	0.992	0.987	0.947	0.991	0.939	0.766	0.937
CIForm	0.831	0.967	0.987	0.971	0.932	0.987	0.875	0.637	0.898
scGraph	0.834	0.973	0.981	0.982	0.941	0.983	0.818	0.619	0.891
ACTINN	0.662	0.761	0.872	0.901	0.916	0.953	0.904	0.671	0.830
scID	0.578	0.590	0.892	0.632	0.627	0.535	0.611	0.548	0.627
CHETAH	0.717	0.850	0.927	0.826	0.803	0.925	0.851	0.562	0.808
SingleR	0.790	0.889	0.945	0.831	0.857	0.968	0.772	0.664	0.840

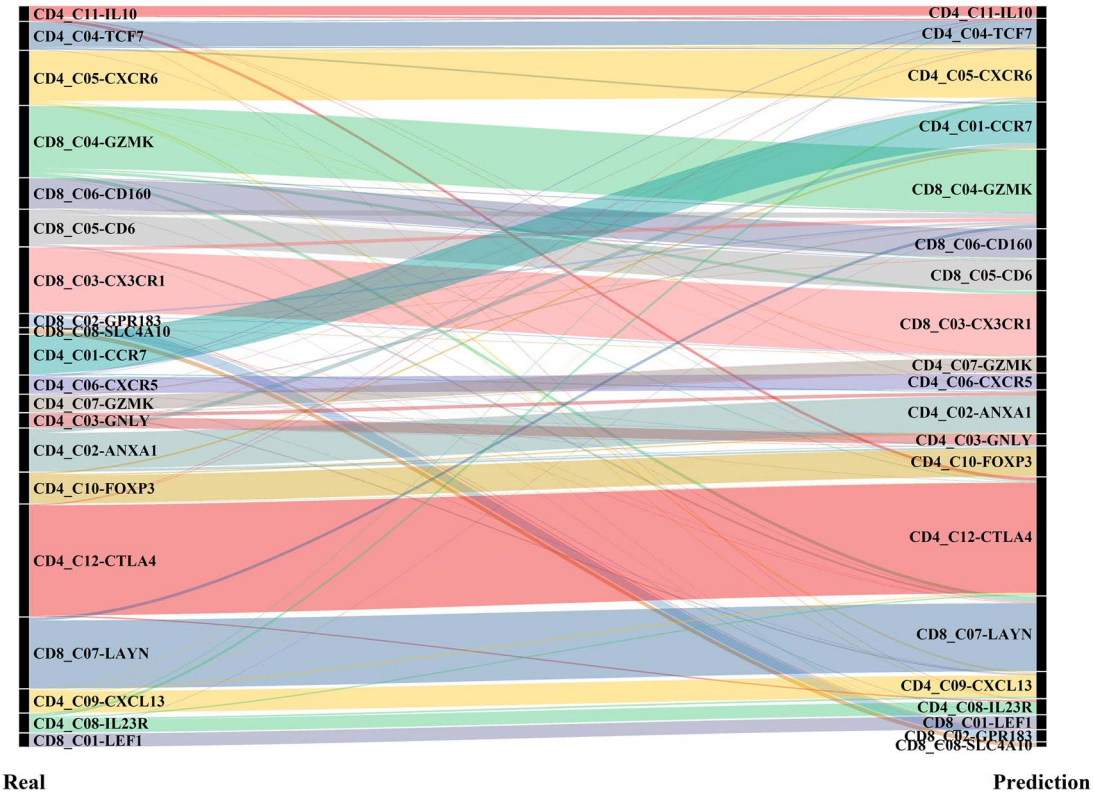

Figure 5. scRGCL annotation results for twenty cell types in the zhangT dataset.

annotation results for CD4 T Cell, CD8 T Cell, and ILC. scGraph cannot accurately distinguish CD4 T cells, CD8 T Cells, and ILC. As shown in Fig. 7, scRGCL accurately annotated each cell type and cell types with fewer numbers, such as mast, can also be accurately identified. CIForm identified many acinar cells as alpha cells or beta cells, and scGraph annotated many ductal cells as gamma cells or alpha cells. The above results show that the annotation results of scRGCL are consistent with the true labels and perform better than CIForm and scGraph.

Evaluating the generalization performance of scRGCL

In this section, several experiments were performed to investigate the generalization performance of scRGCL. To analyze the scalability of scRGCL, we recorded the runtime of scRGCL on eight benchmark datasets (Supplementary Table 7). Among these eight datasets, the smallest dataset is Xin, which contains 1459 cells, and the largest dataset is Lung1, which contains 180 069 cells. The runtime of scRGCL is essentially proportional to the number of cells, and its runtime is acceptable even for large-scale datasets.

To verify whether the pretrained scRGCL can accurately annotate cell types on another independent datasets, we collected three human pancreas datasets (Baron Human [20], Muraro [43], and Segerstolpe [44]). We trained the scRGCL model on the Baron Human dataset and applied the scRGCL to the Muraro dataset and Segerstolpe dataset. For the Muraro dataset, the accuracy of the scRGCL in identifying alpha cell, beta cell, ductal cell, delta cell, and gamma cell was 0.969, 0.957, 0.932, 0.985, and

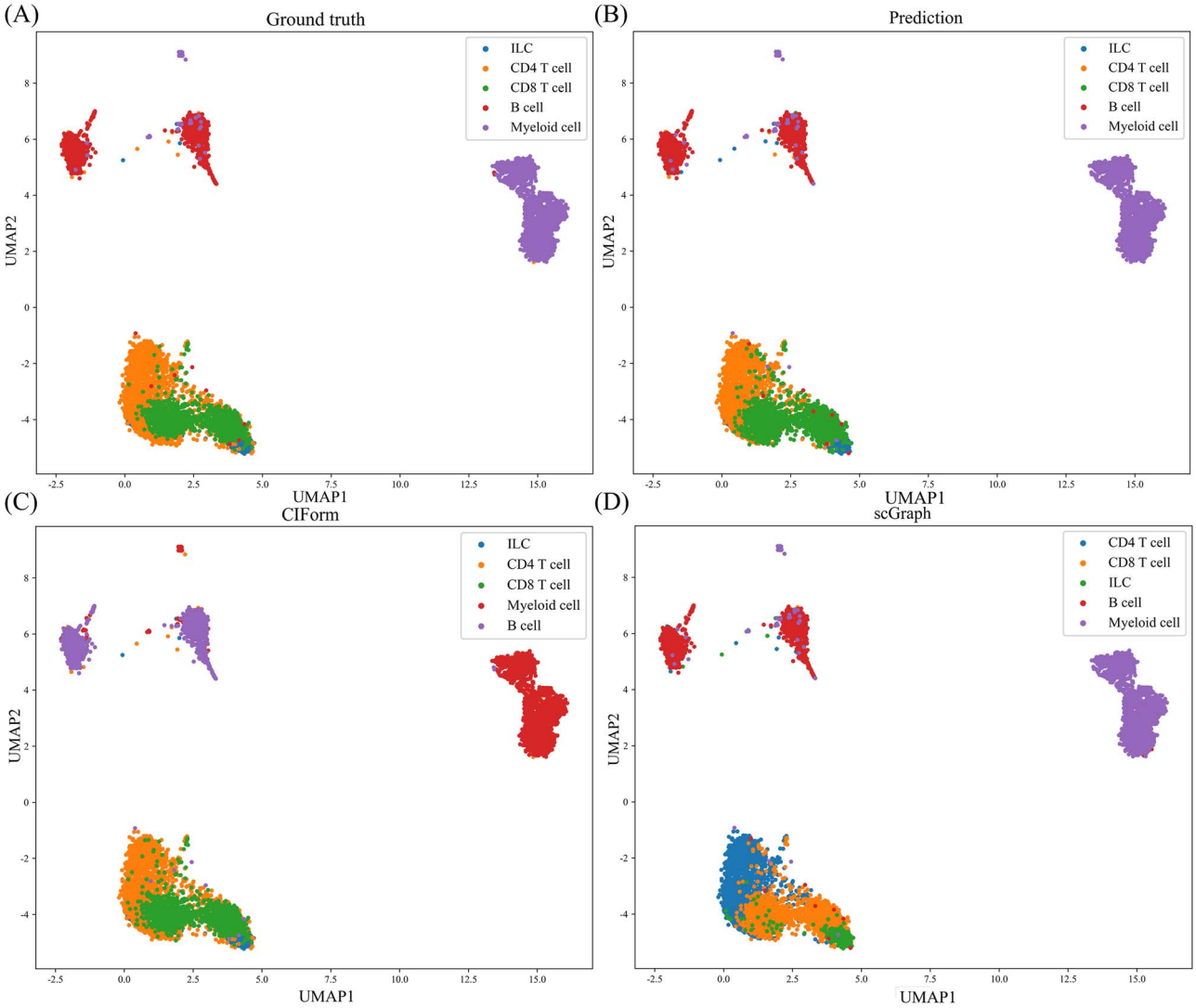

Figure 6. (A) The true cell type labels for the colon dataset. (B) The annotation results of scRGCL for the Colon dataset. (C) The annotation results of CIForm for the Colon dataset. (D) The annotation results of scGraph for the Colon dataset.

0.986, respectively. For the Segerstolpe dataset, the accuracy of the scRGCL in identifying the alpha cell, beta cell, ductal cell, delta cell, and gamma cell was 0.983, 0.965, 0.907, 0.988, and 0.981, respectively.

Furthermore, we trained scRGCL on a large-scale mixed dataset containing multiple tissues and cell types from the HCL [45] database and applied the trained scRGCL to an independent Lung2 dataset [18]. If a single cell was identified as having scores for all cell types below the threshold 0.1, we deleted the cell. As shown in Supplementary Table 8, compared to the performance of scRGCL trained and tested on the Lung2 dataset, the F1 and ACC of scRGCL decreased by 0.043 and 0.037, respectively. The F1 and ACC of CIForm decreased by 0.055 and 0.061, respectively. The F1 and ACC of scGraph decreased by 0.058 and 0.059, respectively. Although the performance of all three methods has decreased, compared with CIForm and scGraph, our proposed scRGCL still has a powerful cell type annotation capability on large-scale mixed datasets.

To investigate the performance of scRCGL in the face of technical noise generated by different sequencing technologies, we used the Colon [19] dataset, which sequenced samples using smart-seq2 [46] and 10x [47] protocols, respectively. The scRGCL showed robustness with ACC values of 0.944 and 0.947 and F1 values of 0.947 and 0.946 for smart-seq2 and 10x sequencing technologies, respectively.

We also investigated the performance of scRGCL using the mouse dataset. We used the STRINGDB mouse PPI network as the gene interaction network of scRGCL and compared the performance of scRGCL and competing methods on the TM dataset (see Table 1). The results of scRGCL and competing methods on the TM dataset (the second column of Table 5) demonstrated that scRGCL still outperformed competing methods even after integrating the PPI networks of other species. This suggests that the scRGCL is not limited to human cells in cell type annotation, and scRGCL can be applied to cell type annotation tasks in different species by replacing gene interaction networks. The above experimental results demonstrated the generalizability of the scRGCL in automatic cell type annotation.

Conclusions and discussion

In this paper, we proposed a new DL-based model, scRGCL, based on the residual graph convolutional neural network and contrastive learning for cell type annotation. By incorporating

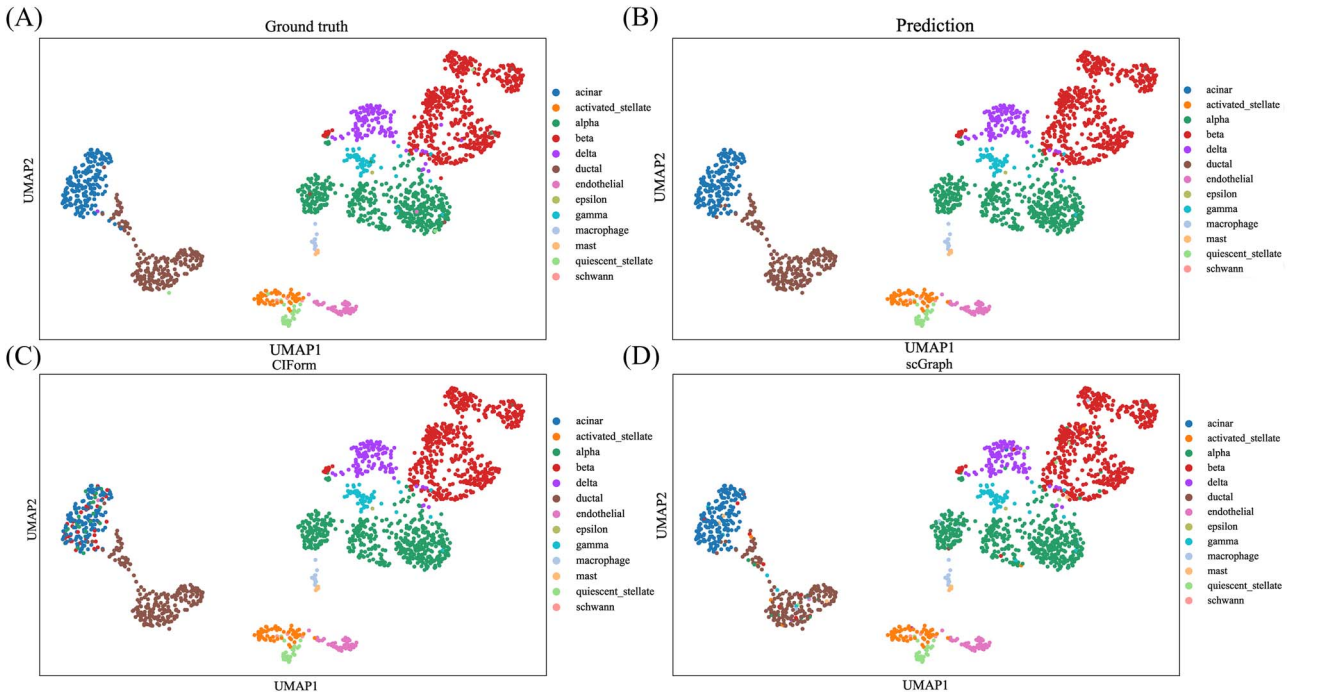

Figure 7. (A) The true cell type labels for the Baron Human dataset. (B) The annotation results of scRGCL for the Baron Human dataset. (C) The annotation results of CIForm for the Baron Human dataset. (D) The annotation results of scGraph for the Baron Human dataset.

a residual graph convolutional neural network and leveraging contrastive learning, scRGCL effectively learns high-order features and cell-to-cell differential features. Experimental results show that scRGCL outperforms state-of-the-art methods and exhibits superior performance in cell type annotation. Experimental results demonstrated the powerful predictive ability of scRGCL in deciphering complex cellular relationships, improving the performance of cell type annotation methods. Experimental results demonstrated the generalization performance of scRGCL.

The interactions between genes often form complex regulatory networks that regulate the biological processes of cells. Many genes have similar expression patterns under specific conditions, indicating that they may be involved in the same biological process. Modeling co-expression networks can help the model identify these genes and their potential functions, but this information cannot be obtained by shallow neural networks. In this article, we constructed a residual graph neural network to increase the depth of the network, which can help the model obtain the complex relationships between genes and use these relationships as high-order features to annotate cell types.

For larger datasets, incorporating more efficient graph neural network architectures or employing pruning techniques could help scale the model more effectively for even larger datasets. We can also select highly expressed genes and limit the number of genes to reduce the computational demands.

Integrating scRNA-seq data and scATAC-seq data can help the model simultaneously analyze gene expression and open chromatin in cells, allowing the model to learn more meaningful features, identify the drivers of differential gene expression, reveal heterogeneity between cells, and improve the accuracy of cell annotation. Integrating multiple gene interaction networks can help the model learn more features and improve the model's performance. Further improvement with contrastive learning would be needed to improve the performance of scRGCL in identifying rare cell types using multi-omics data. We will try to add appropriate discrimination methods in the contrastive learning module to select representative positive and negative samples from multi-omics data to further improve the ability of scRGCL to annotate rare cell types.

Key Points

- There are several limitations of existing methods. First, they do not fully exploit cell-to-cell differential features. Second, they are developed based on shallow features and lack of flexibility in integrating high-order features in the data. Finally, the low-dimensional gene features may lead to overfitting in neural networks. There is an urgent need to develop an effective method to solve this problem.
- We propose a new deep learning (DL)–based model, called scRGCL, which uses a deep network architecture including a residual graph convolutional neural network, contrastive learning, and weight freezing to annotate cell types.
- To verify the effectiveness of scRGCL, we compared its performance with six methods on eight single-cell benchmark datasets from two species (seven in human and one in mouse). Experimental results not only show that scRGCL not only outperforms competing methods but also demonstrates the generalizability of scRGCL for cell type annotation.

Supplementary data

Supplementary data are available at *Briefings in Bioinformatics* online.

Funding

This work was partly supported by grants from the National Natural Science Foundation of China, Nos. 62333018, 62472239, 62372255, U22A2039, 62073231, 62002189, partly supported by the Joint Project of National Natural Science Foundation of China and Russian Science Foundation (W2412087), and supported by the Natural Science Foundation of Ningbo City under Grant No.2023J199, and supported by Key Research and Development (Digital Twin) Program of Ningbo City under Grant Nos.2023Z219, 2023Z226, and supported by the Key Project of Science and Technology of Guangxi (Grant no. 2021AB20147), Guangxi Natural Science Foundation (Grant nos. 2021JJA170204 & 2021JJA170199), and supported by the University Synergy Innovation Program of Anhui Province (No. GXXT-2021-039), and supported by the Natural Science Foundation of Shandong Province, China (Nos. ZR2021MH104, ZR2024MF011, ZR2020QF038), and supported by the Young Taishan Scholars Program, Shandong (tsqn201909178), and supported by the Cultivation Fund of the Second Hospital of Shandong University (2023JX16), and supported by the Shandong Medical Association Qilu medical special project (YKH2022K02112), and supported by the Shandong Province Key Research and Development Program-International Scientific and Technological Cooperation Project (2024KJHZ029), and supported by the Ability Improvement Project of Science and Technology SMES in Shandong Province (2023TSGC0279), and supported by the Youth Innovation Team of Colleges and Universities in Shandong Province (2023KJ329).

Conflict of interest: No competing interest is declared.

Data availability

The source code and data are available at https://github.com/nathanyl/scRGCL.

References

1. 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.
2. Clarke ZA, Andrews TS, Atif J. *et al.* Tutorial: guidelines for annotating single-cell transcriptomic maps using automated and manual methods. *Nat Protoc* 2021;**16**:2749–64.
3. Forcato M, Romano O, Bicciato S. Computational methods for the integrative analysis of single-cell data. *Brief Bioinform* 2021;**22**:bbaa042.
4. de Kanter JK, Lijnzaad P, Candelli T. *et al.* CHETAH: a selective, hierarchical cell type identification method for single-cell RNA sequencing. *Nucleic Acids Res* 2019;**47**:e95–5.
5. Aran D, Looney AP, Liu L. *et al.* Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nat Immunol* 2019;**20**:163–72.
6. Boufea K, Seth S, Batada NN. scID uses discriminant analysis to identify transcriptionally equivalent cell types across single-cell RNA-seq data with batch effect. *iScience* 2020;**23**.
7. Zhou Y, Zhang L, Xu J. *et al.* Category encoding method to select feature genes for the classification of bulk and single-cell RNA-seq data. *Stat Med* 2021;**40**:4077–89.
8. Chu S-K, Zhao S, Shyr Y. *et al.* Comprehensive evaluation of noise reduction methods for single-cell RNA sequencing data. *Brief Bioinform* 2022;**23**:bbab565.
9. Chen J, Xu H, Tao W. *et al.* Transformer for one stop interpretable cell type annotation. *Nat Commun* 2023;**14**:223.
10. Ma F, Pellegrini M. ACTINN: automated identification of cell types in single cell RNA sequencing. *Bioinformatics* 2020;**36**:533–8.
11. Shao X, Yang H, Zhuang X. *et al.* scDeepSort: a pre-trained cell-type annotation method for single-cell transcriptomics using deep learning with a weighted graph neural network. *Nucleic Acids Res* 2021;**49**:e122–2.
12. Yin Q, Liu Q, Fu Z. *et al.* scGraph: a graph neural network-based approach to automatically identify cell types. *Bioinformatics* 2022;**38**:2996–3003.
13. Yuan M, Chen L, Deng M. scMRA: a robust deep learning method to annotate scRNA-seq data with multiple reference datasets. *Bioinformatics* 2022;**38**:738–45.
14. Xu J, Zhang A, Liu F. *et al.* CIForm as a transformer-based model for cell-type annotation of large-scale single-cell RNA-seq data. *Brief Bioinform* 2023;**24**:bbad195.
15. Zhang L, Yu X, Zheng L. *et al.* Lineage tracking reveals dynamic relationships of T cells in colorectal cancer. *Nature* 2018;**564**:268–72.
16. Schaum N, Karkanias J, Neff NF. *et al.* Single-cell transcriptomics of 20 mouse organs creates a tabula Muris: the tabula Muris consortium. *Nature* 2018;**562**:367.
17. Xin Y, Kim J, Okamoto H. *et al.* RNA sequencing of single human islet cells reveals type 2 diabetes genes. *Cell Metab* 2016;**24**:608–15.
18. Kim N, Kim HK, Lee K. *et al.* Single-cell RNA sequencing demonstrates the molecular and cellular reprogramming of metastatic lung adenocarcinoma. *Nat Commun* 2020;**11**:2285.
19. Zhang L, Li Z, Skrzypczynska KM. *et al.* Single-cell analyses inform mechanisms of myeloid-targeted therapies in colon cancer. *Cell* 2020;**181**(442–459):e429.
20. Baron M, Veres A, Wolock SL. *et al.* A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst* 2016;**3**(346–360):e344.
21. Vieira Braga FA, Kar G, Berg M. *et al.* A cellular census of human lungs identifies novel cell states in health and in asthma. *Nat Med* 2019;**25**:1153–63.
22. Sade-Feldman M, Yizhak K, Bjorgaard SL. *et al.* Defining T cell states associated with response to checkpoint immunotherapy in melanoma. *Cell* 2018;**175**(998–1013):e1020.
23. Costanzo M, VanderSluis B, Koch EN. *et al.* A global genetic interaction network maps a wiring diagram of cellular function. *Science* 2016;**353**:aaf1420.
24. Dixon SJ, Costanzo M, Baryshnikova A. *et al.* Systematic mapping of genetic interaction networks. *Annu Rev Genet* 2009;**43**:601–25.
25. Szklarczyk D, Kirsch R, Koutrouli M. *et al.* The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Res* 2023;**51**:D638–46.
26. Kim CY, Baek S, Cha J. *et al.* HumanNet v3: an improved database of human gene networks for disease research. *Nucleic Acids Res* 2022;**50**:D632–9.
27. Persson E, Castresana-Aguirre M, Buzzao D. *et al.* FunCoup 5: functional association networks in all domains of life, supporting directed links and tissue-specificity. *J Mol Biol* 2021;**433**:166835.
28. Franz M, Rodriguez H, Lopes C. *et al.* GeneMANIA update 2018. *Nucleic Acids Res* 2018;**46**:W60–4.
29. Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. *Adv Neural Inf Proces Syst* 2017;30.
30. He K, Zhang X, Ren S. *et al.* Deep residual learning for image recognition. In: Bajcsy R (ed). *Proceedings of the IEEE conference on*

computer vision and pattern recognition, IEEE Computer Society and the Computer Vision Foundation, Las Vegas, Nevada, USA, pp. 770–8, 2016.
31. Chen Z, Wang Z, Yang Y. *et al.* ResGraphNet: GraphSAGE with embedded residual module for prediction of global monthly mean temperature. *Artif Intell Geosci* 2022;**3**:148–56.
32. Khosla P, Teterwak P, Wang C. *et al.* Supervised contrastive learning. *Adv Neural Inf Process Syst* 2020;**33**:18661–73.
33. Perez RK, Gordon MG, Subramaniam M. *et al.* Single-cell RNA-seq reveals cell type–specific molecular and genetic associations to lupus. *Science* 2022;**376**:eabf1970.
34. Yazar S, Alquicira-Hernandez J, Wing K. *et al.* Single-cell eQTL mapping identifies cell type–specific genetic control of autoimmune disease. *Science* 2022;**376**:eabf3041.
35. Miao Z, Zhao M. *Weight Freezing: A Regularization Approach for Fully Connected Layers with an Application in EEG Classification*, arXiv preprint arXiv:2306.05775 2023.
36. Wu L, Li J, Wang Y. *et al.* R-drop: regularized dropout for neural networks. *Adv Neural Inf Proces Syst* 2021;**34**:10890–905.
37. Kingma DP. *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980 2014.
38. Loshchilov I, Hutter F. *Sgdr: Stochastic gradient descent with warm restarts*, arXiv preprint arXiv:1608.03983 2016.
39. Abdi H, Williams LJ. Principal component analysis. *Wiley Interdiscip Rev Comput Stat* 2010;**2**:433–59.
40. Greff K, Srivastava RK, Koutnik J. *et al.* LSTM: a search space odyssey. *IEEE Trans Neural Netw Learn Syst* 2016;**28**:2222–32.
41. Liou C-Y, Cheng W-C, Liou J-W. *et al.* Autoencoder for words. *Neurocomputing* 2014;**139**:84–96.
42. De Simone M, Hoover J, Lau J. *et al.* Comparative analysis of commercial single-cell RNA sequencing technologies. *bioRxiv* 2024;2024.2006. 2018.599579.
43. Muraro MJ, Dharmadhikari G, Grün D. *et al.* A single-cell transcriptome atlas of the human pancreas. *Cell Syst* 2016; **3**(385–394):e383.
44. Segerstolpe Å, Palasantza A, Eliasson P. *et al.* Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell Metab* 2016;**24**:593–607.
45. Han X, Zhou Z, Fei L. *et al.* Construction of a human cell landscape at single-cell level. *Nature* 2020;**581**:303–9.
46. Picelli S, Faridani OR, Björklund ÅK. *et al.* Full-length RNA-seq from single cells using Smart-seq2. *Nat Protoc* 2014;**9**: 171–81.
47. Mimitou EP, Cheng A, Montalbano A. *et al.* Multiplexed detection of proteins, transcriptomes, clonotypes and CRISPR perturbations in single cells. *Nat Methods* 2019;**16**:409–12.

Briefings in Bioinformatics, 2025, **26(1)**, bbae662
https://doi.org/10.1093/bib/bbae662
Problem Solving Protocol