


A robust and scalable graph neural network for accurate single-cell classification

Yuansong Zeng , Zhuoyi Wei, Zixiang Pan, Yutong Lu and Yuedong Yang

Corresponding authors: Yuedong Yang, School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510000, China. Tel.: +86 020-37106046; Fax: +86 020-37106020; E-mail: yangyd25@mail.sysu.edu.cn. Yutong Lu, E-mail: yutong.lu@nssc-gz.cn

Abstract

Single-cell RNA sequencing (scRNA-seq) techniques provide high-resolution data on cellular heterogeneity in diverse tissues, and a critical step for the data analysis is cell type identification. Traditional methods usually cluster the cells and manually identify cell clusters through marker genes, which is time-consuming and subjective. With the launch of several large-scale single-cell projects, millions of sequenced cells have been annotated and it is promising to transfer labels from the annotated datasets to newly generated datasets. One powerful way for the transferring is to learn cell relations through the graph neural network (GNN), but traditional GNNs are difficult to process millions of cells due to the expensive costs of the message-passing procedure at each training epoch. Here, we have developed a robust and scalable GNN-based method for accurate single-cell classification (GraphCS), where the graph is constructed to connect similar cells within and between labelled and unlabeled scRNA-seq datasets for propagation of shared information. To overcome the slow information propagation of GNN at each training epoch, the diffused information is pre-calculated via the approximate Generalized PageRank algorithm, enabling sublinear complexity over cell numbers. Compared with existing methods, GraphCS demonstrates better performance on simulated, cross-platform, cross-species and cross-omics scRNA-seq datasets. More importantly, our model provides a high speed and scalability on large datasets, and can achieve superior performance for 1 million cells within 50 min.

Keywords: single-cell RNA sequencing, single-cell classification, batch effects, scalable graph neural network, virtual adversarial training

Introduction

Single-cell RNA sequencing (scRNA-seq) technologies promise to provide high-resolution insights into the complex cellular ecosystem [1–3] by measuring gene expression in millions of single cells from multiple samples [4–8]. Several large-scale single-cell projects, e.g. the human cell atlas (HCA), have been established as a result of the decreasing costs in scRNA-seq technologies [9, 10]. In scRNA-seq studies, an essential step is to identify the sequenced cells through the sequenced gene expression [11], which is usually obtained through cell clustering and subsequently manually identifying cell clusters through marker genes [12]. This process is time-consuming and subjective.

With the tremendous increase of well-annotated scRNA-seq datasets, it is feasible to transfer well-defined labels (cell types) of existing single-cell datasets to newly generated single-cell datasets [13, 14]. However, the knowledge transferring is challenging due to various

noises among scRNA-seq data (e.g. dropout; [15, 16]). In addition, batch effects exist between single-cell datasets because they are usually collected from different platforms [17, 18], tissues or species [19, 20]. Early methods were developed to search for similar cells in the reference datasets with well-defined labels. For example, scmap [21] measures the maximum similarity between well-annotated cells of reference data and unknown query data to annotate cell types. SingleR [22] measures the similarity by calculating the correlation between gene expression. CHETAH [23] identifies the unknown cells using the high cumulative density of each cell type correlation distribution. OnClass labels cells according to the most similar cells annotated in the Cell Ontology [24]. CelliD is a clustering-free statistical method for extracting the gene signatures of each cell from scRNA-seq data, and used the gene signatures as unique cell identity cards to align with annotated cells [25]. CelliD provides two modes: CelliD (C) by aligned

Yuansong Zeng is a PhD student at the School of Computer Science and Engineering at Sun Yat-Sen University. His research interests include deep learning, graph neural network, and single-cell RNA-seq data analysis.

Zhuoyi Wei is a graduate student at the School of Computer Science and Engineering at Sun Yat-Sen University. His research interests include deep learning and single-cell RNA-seq data analysis.

Zixiang Pan is a graduate student at the School of Computer Science and Engineering at Sun Yat-Sen University. His research interests include deep learning and bioinformatics.

Yutong Lu is a professor at the School of Computer Science and Engineering and the National Super Computer Center at Guangzhou, Sun Yat-sen University, China. Her research interests include parallel system management and high-speed communication.

Yuedong Yang is a professor at the School of Computer Science and Engineering and the National Super Computer Center at Guangzhou, Sun Yat-sen University, China. His research group emphasizes on developing HPC and AI algorithms for protein function prediction, multi-omics data integration and intelligent drug design. He is also responsible for constructing the HPC platform for biomedical applications based on the Tianhe-2 supercomputer.

Received: October 28, 2021. **Revised:** December 1, 2021. **Accepted:** December 11, 2021

© The Author(s) 2022. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oup.com

with cells and CellID(G) by aligned cell clusters in the reference dataset. Obviously, these methods consider only pairwise similarity and have ignored the nonlinear relations between annotated cells. For this reason, several methods train classifiers using the labeled datasets or reference atlas, and make predictions on query datasets. For example, scPred [26] trains a support vector machine by using the features obtained from singular value decomposition. SingleCellNet [27] applies an ensemble of boosted regression trees and a Random Forest classifier to annotate cells. Seurat [28] is a commonly used toolkit in single-cell studies, which applies a specialized method to transfer labels to unknown cell types. scClassify is a supervised classification framework for multilevel cell type annotations, relying on cell type hierarchies from single or multiple reference datasets [29]. scANVI [30] is a semi-supervised variant of scVI [31], which annotates cell types in the query dataset by leveraging all available cell state annotations. scNym is a semi-supervised deep learning classification framework that uses an adversarial neural network to transfer cell identity annotations from reference to query [32]. In addition, many methods [33–39] have been developed for different scenes, e.g. scMatch and scID. However, all these methods still exhibit limited performance, partially due to their ignorance of higher-order relations between cells.

In fact, the high-order representation and topological relations could be naturally learned by the graph neural network (GNN), and GNN have been proven with improved performance in scRNA-seq data analyses such as imputation and clustering [40–42]. ScGCN [43] is currently the only GNN method for annotating cells. The method is based on the GNN architecture proposed by Kipf and Welling [44], which relies on an expensive message-passing procedure to propagate information and has to include the full-batch during training. Thus, the huge costs of computations and memory prevent its applications to large datasets, especially with the arrival of datasets containing millions of cells [45, 46].

To solve the scalability of GNN, many studies have been proposed. For example, Chen *et al.* [47] proposed a scalable GNN model, which could be efficiently trained with mini-batches using GPU. One critical point is its approximation of the diffused information through the bidirectional propagation by the Generalized PageRank algorithm [48], which avoids iterative information diffusion in each training epoch. In addition, the use of mini-batch training reduces the requirement of large GPU memory from full-batch training. Thus, the method could be used on large graphs with billions of edges. Another issue for GNN is to accurately construct the cell graph among millions of cells. Traditional methods such as Cosine similarity, KNN, UMAP [49] and Annoy [50] (<https://github.com/spotify/annoy>) are widely used for constructing the cell graph by measuring the cell-to-cell similarity in single-cell RNA-seq data [51–53], but they do not take account of the batch effects between

datasets. To consider the batch effects, several methods captures the cell relations through scGCN [54] constructs the cell graph using CCA–MNN, a combination of canonical correlation analysis (CCA; [55]) and the mutual nearest neighbor (MNN; [56]). Conos [57] relies on multiple plausible inter-sample mappings to construct a graph connecting all measured cells. BBKNN [58] provides an extremely fast and scalable neighborhood construction method across all batches. The runtimes of BBKNN scale linearly with the increase in number of cells through integrating the approximate neighbor detection technique in algorithm Annoy.

Here, we present a scalable GNN learning model for cell annotations by constructing the graph via BBKNN, and pre-calculate the diffused features via the graph bidirectional propagation algorithm (GBP). Concretely, GBP propagates information among similar cells within and between labeled and unlabeled datasets, resulting in significant gains of speed and scalability of GNN while efficiently removing the batch effects. The integrated features from the GBP module are then inputted to a classification neural network to annotate cells for the query dataset. To better estimate the decision boundary between different cell types, we also use the virtual adversarial training (VAT) loss [59] to improve model generality. Our method was demonstrated to outperform other methods on both simulated datasets and real datasets across species, platforms and omics. More importantly, the model can be extended to large-scale datasets in a reasonable time scale.

Materials and methods

Datasets

We benchmarked our method through multiple simulated and real scRNA-seq datasets. The simulated datasets were generated by the R package ‘splatter’, and the real datasets were obtained from previous references. As shown in Table 1, the real scRNA-seq datasets included four paired cross-species datasets, eight cross-platform datasets (including a multiple-reference dataset), two paired cross-omics datasets, two paired unknown cell type datasets (tumor datasets), two paired minor cell type datasets and four paired benchmark datasets. The data preprocessing was detailed in Supplementary Note 1.

The architecture of GraphCS

This study proposed a robust and scalable GNN model to annotate cell types in a semi-supervised manner. As shown in Figure 1, the GraphCS model consists of three modules: graph construction, GBP and classification modules.

Graph construction module

The cell graph G is constructed by linking cells with similar gene expressions within and between the reference and query datasets. Here, we construct the graph G by

Table 1. Summary of the dataset pairs used in this study

Analysis	Reference										Query									
	Dataname	Species	Protocol	# of cells	# of genes	# of cell types	Dataname	Species	Protocol	# of cells	# of genes	# of cell types								
Simulation Cross-platform	Simulation	Mouse	Drop-seq	2000	10 000	4	Simulation	Mouse	Drop-seq	1000	10 000	4								
	Mouse retain	Mouse	Drop-seq	26 830	12 333	5	Mouse retain	Mouse	Drop-seq	43 603	12 333	5								
	Mouse brain	Human	Drop-seq	691 600	17 745	12	Mouse brain	Mouse	SPLiT-seq	141 606	17 745	11								
	PbmcBench	Human	10x Chromium(v2,v3)	13 028	33 694	9	PbmcBench	Human	Smart-seq2	526	33 694	7								
Cross-species	multiple_references (Segerstolpe, Mutaro, Wang)	Human	CEL-Seq2Smart-seq2SMARTer	3418	10 992	12	Baron human	Human	inDrop	8569	10 992	10								
	Baron mouse	Mouse	inDrop	1886	12 808	13	Baron human	Human	inDrop	7568	20 215	11								
Minor types	Baron human	Human	inDrop	8569	20 215	14	Baron mouse	Mouse	inDrop	1868	12 808	11								
	PBMC_30K	Human	10x	29 079	12 423	12	GSE99254	Human	Smart-seq2	5675	12 423	5								
cross-omics	GSE120575	Human	Smart-seq2	16 291	13 519	14	GSE148190	Human	10x	19 374	13 519	8								
	TM (kidney)	Mouse	10x	2781	12 543	8	kidney_sci-ATAC-seq	Mouse	sci-ATAC-seq	4187	12 543	4								
Tumor dataset	TM (lung)	Mouse	10x	5404	11 990	13	lung_sci-ATAC-seq	Mouse	sci-ATAC-seq	4103	11 990	6								
	GSE72056	Human	Smart-seq2	3280	17 860	7	GSE103322	Human	Smart-seq2	4507	17 860	5								
Benchmarks of HCL, MCA, and TM 10x	GSE118056	Human	10x	6043	15 976	8	GSE117988	Human	10x	10 082	15 976	5								
	MCA(Spleen)	Mouse	Microwell-seq	1970	7125	10	TM (Spleen)	Mouse	10x	3253	7125	4								
	HCL(AdultOmentum)	Human	Microwell-seq	12 785	14 406	12	MCA(AdultOmentum)	Mouse	Microwell-seq	2807	14 406	5								
	HCL(AdultBoneMarrow)	Human	Microwell-seq	8693	14 406	11	MCA(AdultBoneMarrow)	Mouse	Microwell-seq	7120	14 406	5								
	TM(Muscle)	Mouse	10x	3909	3935	6	MCA(Muscle)	Mouse	Microwell-seq	645	3935	4								

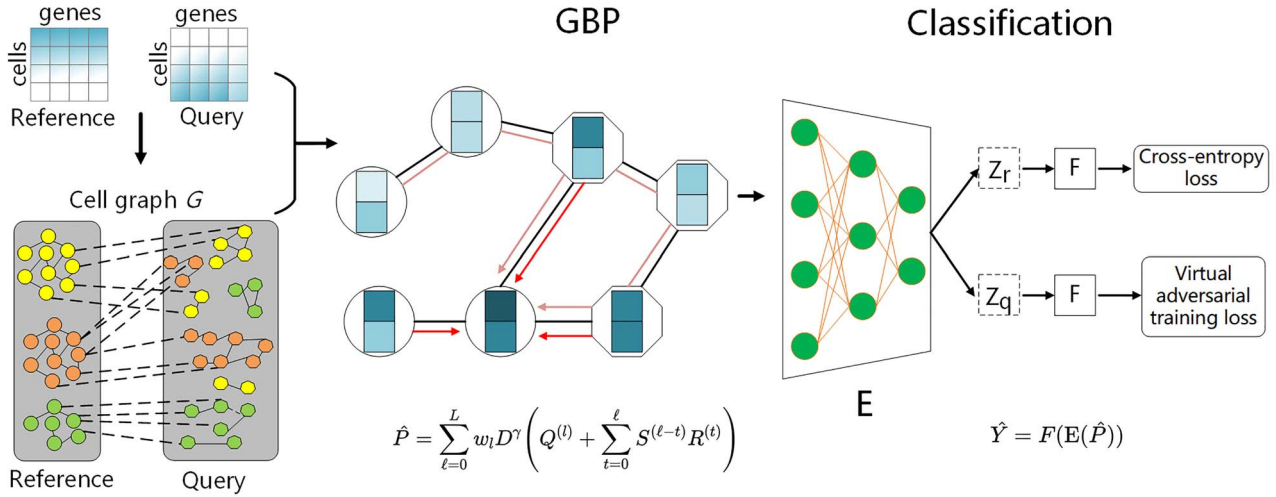


Figure 1. The schematic overview of GraphCS for cell type classification. GraphCS consists of graph construction, GBP and classification modules. The graph construction module constructs the cell graph according to gene expression similarity through the BKNN algorithm. Through the graph, the GBP module diffuses feature information among cells, which is then inputted into the classification module used to classify cells.

BKNN with default parameters, which provides a fast and scalable neighborhood construction method across all batches. Briefly, for each cell c , three most similar cells are selected with the lowest Euclidean distances from each of N_b batches (including the batch itself). The connected cell graph is then inputted into UMAP for recalculating connectivity scores, through which neighbored cells are trimmed so that each cell contains at most $30N_b$ neighbors (edges). The constructed cell graph is then input to the following GBP module.

GBP module

To acquire high scalability, GNN is estimated through the Generalized PageRank algorithm, which is further approximated by the GBP Algorithm.

Generalized PageRank algorithm

To acquire high scalability of GNN, the feature propagation is pre-calculated through Generalized PageRank matrix as:

$$P = \sum_{\ell=0}^L w_{\ell} (D^{\gamma-1} A D^{-\gamma})^{\ell} \cdot X \quad (1)$$

where w_{ℓ} is the weight of the ℓ th order convolution matrix, A and D are the adjacency matrix and diagonal degree matrix of graph G , respectively, X is the feature matrix and γ is the convolution coefficient. This strategy has been proven to well estimate feature propagation [60], and we followed the study to set $w_{\ell} = \alpha(1 - \alpha)^{\ell}$ for constant decay factor $\alpha \in (0, 1)$.

The GBP algorithm

To reduce the time complexity, the Generalized PageRank is further approximated with the GBP that combines the Monte-Carlo Propagation and Reverse Push Propagation. GBP has been proven to provide accurate unbiased estimator [47]. Concretely, we use the following formula as an unbiased estimator for the Generalized PageRank

matrix P defined in Equation (1).

$$\hat{P}^{n \times d} = \sum_{\ell=0}^L w_{\ell} D^{\gamma} \cdot \left(Q^{(\ell)} + \sum_{t=0}^{\ell} S^{(\ell-t)} R^{(t)} \right) \quad (2)$$

where n is the total number of cells in the reference and the query data, and d is the size of gene features. Q and R are respectively the reserve matrix and the residue matrix originated from the Reverse Push Propagation algorithm, and S records the fraction of random walks from the Monte-Carlo propagation. The detailed information and proof of Equation (2) can be found in ref. [47].

Classification module

The feature matrix $\hat{P}^{n \times d}$ obtained from the GBP module was input to our classification module to make predictions of cell types. Here, the module is composed of the neural network classification, based on which the virtual adversarial loss is added to improve the generality.

Neural network classification: Our classification module contains a neural network feature extractor E with multiple hidden layers and a label predictor F with a *Softmax* output layer. The input of classification module includes reference gene expression matrix $X_r = [x_1^r, \dots, x_{m_r}^r] \in \hat{P}^{m_r \times d}$ with the corresponding labels $Y_r = \{y_i^r\}_{i=1}^{m_r}$ and query gene expression matrix $X_q = [x_1^q, \dots, x_{m_q}^q] \in \hat{P}^{m_q \times d}$. We optimize the classification module using the following standard cross-entropy loss:

$$L_{CE} = -\frac{1}{m_r} \sum_{i=1}^{m_r} y_{i,r}^T F(E(x_i^r)) \quad (3)$$

where $y_{i,r} \in \mathbb{R}^{CL \times 1}$ is one-hot encoded vector of y_i^r and CL is the number of class.

Virtual adversarial training: VAT is applied to improve the generalization of the classification module by

incorporating the information of data distribution from query data. VAT is a data augmentation technique without prior label information [61]. VAT tries to make predictions invariant to small perturbation by minimizing the distance between the input and a perturbed version of the input. Then the model is robust to small noises or perturbations in the inputs. We compute VAT's loss function as the following:

$$L_{\text{VAT}}(X_q, \theta) = D_{\text{KL}}[p(Y_q|X_q, \theta), p(Y_q|X_q + r_{\text{vat}}, \theta)] \quad (4)$$

$$r_{\text{vat}} = \arg \max_{\Delta x: \|\Delta x\|_2 \leq \epsilon} D_{\text{KL}}[p(Y_q|X_q, \theta), p(Y_q|X_q + \Delta x)] \quad (5)$$

where r_{vat} optimizes the difference between the model output of the non-perturbed input and the perturbed input, θ is parameter of the model, Δx is a Gaussian noise and Y_q is predicted by the label predictor F . The hyper-parameter ϵ is the norm constraint for the adversarial direction, and we set ϵ to 0.1 following the previous study [59]. The output distribution is parameterized as $p(Y_q|X_q, \theta)$, and $D_{\text{KL}}[\bullet, \bullet]$ is Kullback–Leibler divergence.

So, the total loss function of classification module as the following:

$$L_{\text{overall}} = L_{\text{CE}} + \lambda L_{\text{VAT}} \quad (6)$$

where λ is the hyper-parameters (set as 0.1) to balance the contribution of VAT to the total loss function.

Hyper-parameters setting

The GraphCS was implemented in PyTorch and C++. For GBP module, we set $\alpha = 0.05$ and $\gamma = 0.5$ for all datasets. For the classification module, the dimensions of hidden layers were set to [256, 256]. The training batch size was generally set as 128, whereas the sizes was increased for large datasets (1024 and 4096 for above 10 000 and 50 000 cells, respectively) to further reduce the training time of each epoch on large datasets. The models were optimized through the Adam optimizer with a learning rate of 0.001, a maximum of 1000 epochs, and early stopping with a patience of 20 epochs. Since the performance of our model was affected by the constructed cell connections (inter-edges) by BBKNN between two batches, we decided the final number of inter-edges through an empirical parameter `edge_ratio`, the ratio of selected inter-edges relative to the number of cells in the larger batch. The `edge_ratio` was set as 2 in default and 0.5 for difficult datasets with large batch effects (e.g. cross-omics data) by trials and tests. All results reported in this paper were conducted on Ubuntu 16.04.7 LTS with Intel® Core (TM) i7-8700K CPU @ 3.70 GHz and 256GB memory, with the Nvidia Tesla P100 (16G).

Benchmarking classification methods

To evaluate the performance, we compared GraphCS with other tools including: Seurat V3, `scmap`, `scPred`, `CHETAH`, `SingleR`, `SingleCellNet`, `scGCN`, `onClass`, `scClassify`, `scNym`, `scANVI` and `CelliD`. For Seurat V3, we applied

both the Principal component analysis (PCA)-based and Canonical correlation analysis (CCA)-based version to evaluate whether the aligned data was benefit for classification. We used the default hyper-parameters recommended in the origin paper for the competing methods.

Evaluation metrics: We evaluated the classification performance for all methods using the accuracy, the proportion of correctly annotated cells. For each dataset, we considered the cell type annotations provided by the original dataset as the ground truth.

Results

Performance on simulated datasets

To investigate the performance of GraphCS under different magnitudes of batch effects, we generated the simulated scRNA-seq data by setting different values of 'batch.facScale' through the R package 'Splatter'. As shown in Figure 2A, the accuracies of all methods decreased with the increase of batch.facScale since higher batch.facScale represented larger batch effects, i.e. higher annotating difficulty. Overall, our method consistently achieved stable and the best performance with the accuracies only slightly changed from 1.0 to 0.97 when increasing batch.facScale from 0.2 to 1.6. By comparison, `scGCN`, the second-best method, had significant drop in accuracies when batch.facScale was greater than 1.0, and a sharp drop from 0.93 to 0.88 when increasing batch.facScale from 1.4 to 1.6. The accuracies of `SingleR` and `scPred` were larger than 0.9 when the value of batch.facScale was less than 0.4, but their accuracies significantly dropped afterwards and were only 0.45 and 0.4, respectively when batch.facScale = 1.6. For two Seurat methods, `Seurat-PCA` was more sensitive from the batch.facScale value. `Seurat-PCA` had higher accuracies than `Seurat-CCA` at batch.facScale of <1.0, but lower accuracies at greater batch.facScale values. This is likely because `Seurat-CCA` overcorrected the batch effects at small batch.facScale values. `scNym` performed better than `scANVI` in terms of average accuracy, but both of them were worse than `scGCN`. By comparison, GraphCS always outperformed the competing methods in different magnitudes of batch effects. The superior performance showed that GraphCS could effectively reduce performance degradation brought by batch difference.

Performance on real datasets

We further evaluated the performance of GraphCS on different types of real datasets. For the cross-platform datasets, we tested on seven paired cross-platform datasets. As shown in Figure 2B, the average accuracy of GraphCS (mean Acc=89%) was 2% higher than the second-ranked method `scNym` (mean Acc=87%) and consistently outperformed other competing methods. `Seurat-PCA`, `SingleCellNet`, `CelliD (C)`, `SingleR`, `scANVI`, `scGCN` and `scmap` ranked the 3rd, 4th, 5th, 6th, 7th, 8th and 9th in terms of the average accuracy, respectively. In

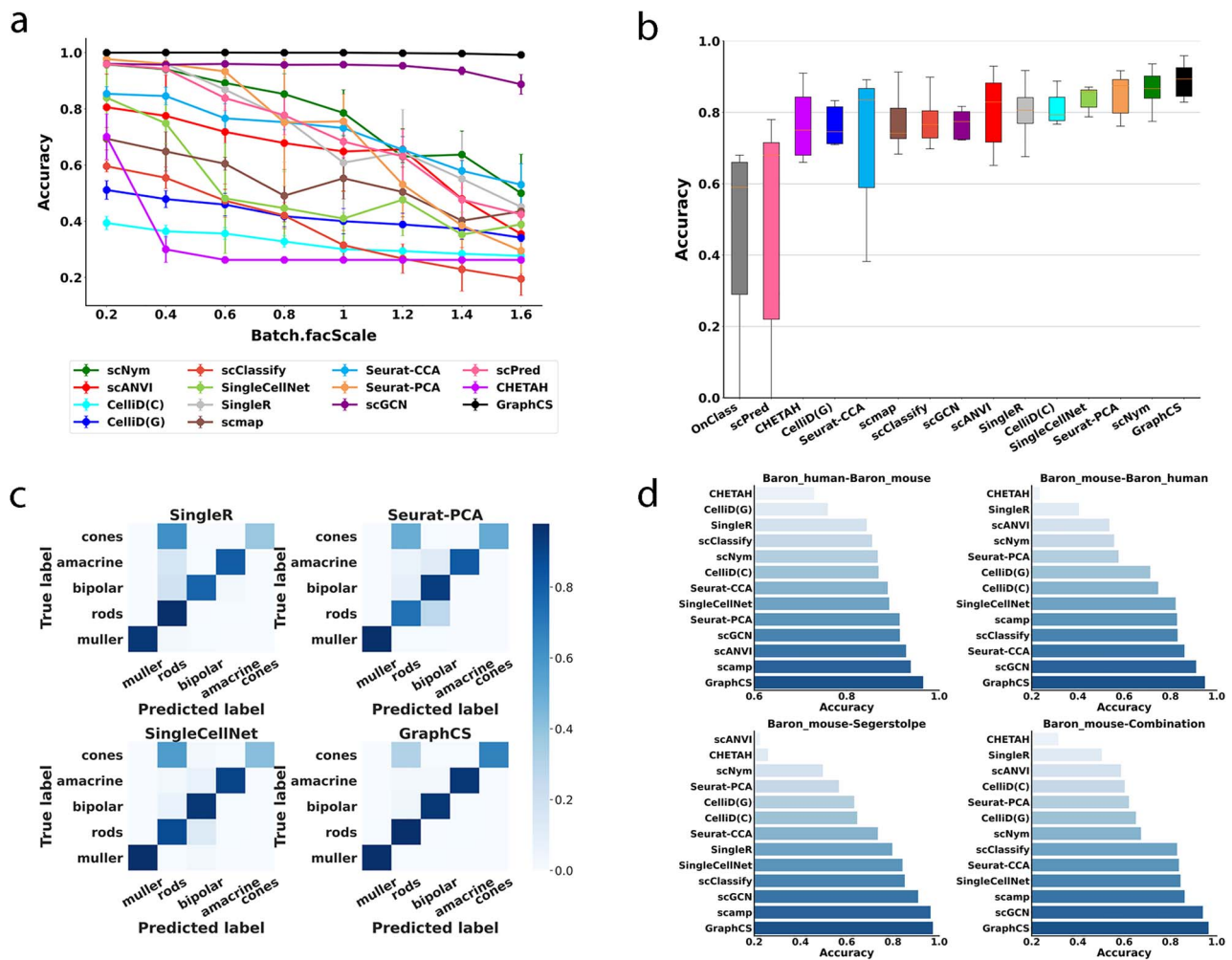


Figure 2. The performance of GraphCS on simulated, cross-platform and cross-species datasets: **(A)** the average and mean square error values of cell type prediction accuracy on five groups of simulated scRNA-seq data at different batch.facScale values; **(B)** the boxplots of cell type prediction accuracy of all methods based on the cross-platform datasets; **(C)** the accuracy matrix of each cell type identified by different methods on the mouse retina dataset and **(D)** the performance of GraphCS on four paired cross-species datasets. Baron_human-Baron_mouse represents the Baron human pancreas dataset as the reference to annotate the Baron mouse pancreas dataset. The rest results represent that the Baron mouse pancreas dataset as the reference to annotate respectively the Baron, Segerstolpe and the combination human pancreas datasets (the combination contained five human pancreas datasets, including Baron et al., Wang et al., Xin et al., Muraro et al. and Segerstolpe et al.). Each bar represents the accuracy of each method.

comparison with Seurat-PCA, Seurat-CCA did not benefit from aligning and integrating the datasets. CHETAH, scmap, CellID (G) and scGCN achieved similar average accuracy. Though scGCN took a similar technique to ours, the average accuracy of scGCN was lower than GraphCS. It is likely because the scGCN constructed graphs containing fewer edges (averagely one to two times lower than ours) and did not fully utilized the advantages of GNN. ScPred and onClass achieved much lower performance than other methods. To highlight the comparison regarding specific cell types, we used the heatmap to show the accuracy of each cell type annotated by different methods on the mouse retina dataset. As shown in Figure 2C and Supplementary Figure S1. CellID(G) and scmap incorrectly assigned most of bipolar cells. SingleR, SingleCellNet, Seurat-PCA, CHETAH, CellID(C), scANVI, scClassify and scNym incorrectly assigned most of cones cells. In contrast, our method correctly discriminated most cell types. In addition, we performed

an experiment by using multiple reference datasets (Supplementary Figure S2), and our model achieved comparable accuracy with CellID and outperformed other methods.

For the cross-species datasets, we evaluated all methods on four paired cross-species datasets. We did not include scPred since it raised exceptions on cross-species datasets. As shown in Figure 2D, GraphCS achieved an average accuracy of 0.96, respectively 4 and 7% higher than those by the second-ranked method scGCN (0.92) and the third-ranked method scmap (0.89). The left methods are ordered as: SingleCellNet, Seurat-CCA, scClassify, CellID(C), CellID(G), Seurat-PCA, scNym, SingleR, scANVI and CHETAH. Specifically, in the combination dataset with only seven T cell and 13 Schwann types, GraphCS could still annotate them accurately (Supplementary Figure S3A). As shown in the Sankey diagram (Supplementary Figure S3B), the much smaller number of cells in the reference data than the

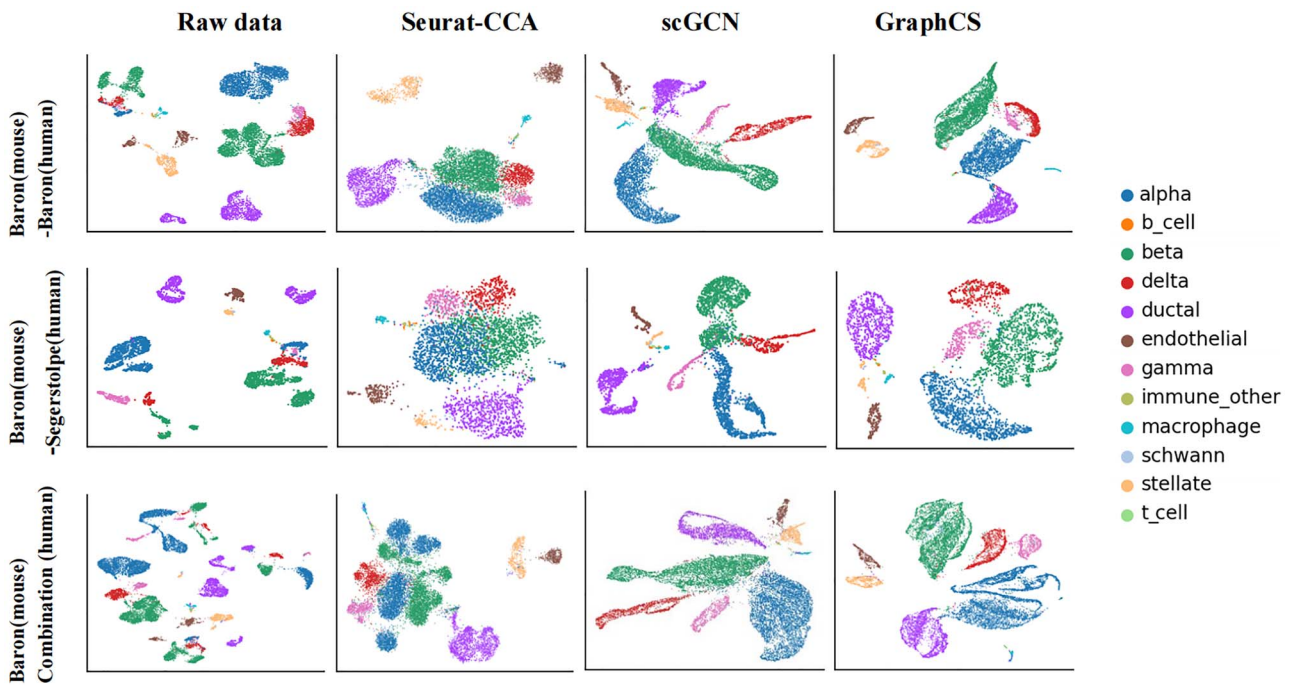


Figure 3. UMAP visualization of three paired cross-species datasets, based on the aggregated data by different methods. The Baron mouse pancreas dataset is the reference for all query datasets. First row: the Baron human pancreas dataset as the query data. Second row: the Segerstolpe human pancreas dataset as the query data. Third row: we combined datasets Baron *et al.*, Wang *et al.*, Xin *et al.*, Muraro *et al.* and Segerstolpe *et al.* as the query data.

query data suggests the capability of our model in small reference data.

To interpret our method, we visualized the cells in the aggregated reference-query data of cross-species. We compared Seurat-CCA, scGCN and GraphCS since they provided the aggregated data and took account of batch effects between datasets. As shown in Figure 3, cells in the raw data were not separated well due to the substantial noise and batch effects. For example, in the dataset Baron (mouse)-Baron (human), beta cells were separated into two clusters, whereas alpha and delta cells gathered together. Although Seurat and scGCN could discriminate most of the cell populations on all cross-species datasets, they could not explicitly distinguish a few cell types, such as beta and delta cells. By comparison, GraphCS could clearly separate most of the cell populations in all scenarios, indicating its ability to deal with strong batch effects between species.

To evaluate whether GraphCS can classify these unknown cell types, we trained it on two paired tumor datasets, where malignant cells were included in the query data but not in the reference data. This is practically important since the query data may contain novel cell types not appearing in the reference dataset, and classifiers should identify the novel unknown cells, or output low classification confidence scores for unknown cell types. We followed the previous study [62] to evaluate all methods. Specifically, we evaluated each method by its precision (the percentage of correctly annotated cells for known cell types) under a given FPR, the percentage of falsely assigned unknown cells. For this purpose,

we selected a threshold of predicted score output by each method, so that 1-FPR of unknown cell types are annotated as unknown, and all cells with scores below the threshold are defined as unknown. The precision was defined as the number of cells assigned with correct known cell types and with scores above the threshold, divided by the total number of cells with known cell types. As shown in Supplementary Figure S4, at the FPR of 0.05, GraphCS, together with scGCN and Seurat, achieved the highest average precision (~68%). The next-level methods are scANVI, SingleCellNet, scNym, scPred and CHATAH with an average precision of 0.60, 0.56, 0.43 and 0.37, respectively. SingleR and scmap performed worst in terms of precision. The results indicated the advantage of our model in identifying unknown cell types. The trend was similar at the FPR of 0.1. We did not include onClass because it could not run without a pretrained model, or CelliD and scClassify because they did not return the confidence scores.

To evaluate the performance for transferring labels across different types of omics, we performed experiments on two cross-omics datasets. As recommended in Seurat, we first converted the peak matrix of the scATAC-seq data to a ‘gene activity matrix’ by adding the counts in the gene body +2 kb upstream, representing a synthetic scRNA-seq dataset to leverage for annotation. As shown in Figure 4, our model achieved the highest accuracy of 0.95 that was 5% higher than the second-ranked method scGCN. SingleR, scClassify and CelliD (C) achieved slightly lower but reasonable accuracies of 0.88, 0.85 and 0.74, respectively. scPred, scNym and

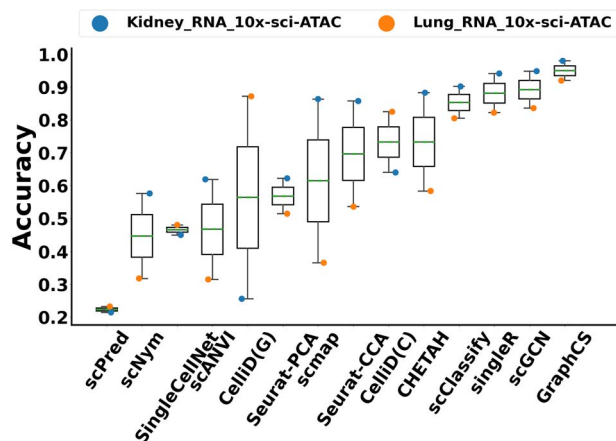


Figure 4. The performance of different methods on cross-omics datasets.

SingleCellNet did not perform well on cross-omics datasets with accuracies below 0.5. The results have shown that our model performed well on cross-omics datasets.

To evaluate the performance of our model in the classification of minor cell types, we compared it with competing methods on two paired datasets of three to four T cell subtypes. As shown in [Supplementary Figure S5](#), GraphCS performed the best by 1 and 2% higher than the second-ranked method Seurat-PCA in terms of average accuracies of all cells and T cell subtypes for all datasets. To further examine the ability of our model when fewer cell types are in the query dataset than the reference dataset, we employed the reference dataset GSE120575 containing 14 cell types (including four T cell types CD8Tcm, CD4Tn, CD8Tem and CD8Tex) and the query dataset GSE148190 containing eight cell types (including four T cell types CD8Tcm, CD4Tn, CD8Tem and CD8Tex). By recursively keeping only one T cell type and removing other T cells on the query dataset, the four experiments by our model produced an average accuracy of 0.68 for all T cell types, essentially the same as 0.66 by the direct test on GSE148190. The results indicate that our model can apply to reference datasets containing many cell types. We further evaluated our model on four paired datasets from HCL, MCA and TM. As shown in [Supplementary Figure S6](#), our model consistently performed the best with average accuracies at least 2% higher than other methods. These results indicated the efficiency and robustness of our model.

We also investigated the contribution of each component in GraphCS through ablation studies. The GBP module made the biggest contribution because its removal of batch effects between datasets, and the VAT module made small but significant contribution. Both modules are critical to our model. The results were detailed in [Supplementary Note 2](#).

Running time evaluation

To evaluate the runtimes of all methods and their scalability with the increase in the number of cells, we

sampled the mouse brain dataset in a stratified way (i.e. preserving population frequencies) to 6, 12, 36, 60, 96 and 120% of the original number of 833 206 cells and selected the top 2000 highly variable genes as the input features. As shown in [Figure 5](#), dramatic differences of runtimes could be observed between these methods with increases in the number of cells. GraphCS was faster than all other methods except scmap. GraphCS showed a high scalability with about linear growth of runtimes with the number of cells: 1008s for 500K cells and 2669s for 1000K cells. scNym and scANVI achieved similar time costs to GraphCS. This was six times faster than CHETAH. Seurat-PCA was close to our method in speed for dataset with 50K cells, but the runtime dramatically increased for large datasets: 42 times slower than our method when processing 1 million cells. Seurat-CCA was consistently slower than Seurat-PCA, and SingleCellNet was the slowest. Although GraphCS was two times slower than scmap, GraphCS consistently achieved average accuracies of 20% higher than scmap. In addition, under the default parameters, scmap could not process the dataset with >800K cells. scGCN cannot either process datasets with >500K cells because current GPU memory cannot support the full-batch training. In contrast, our model could deal with large datasets with > 1 million cells because the GBP module used in our model supports training with mini-batches. When the number of cells was less than 300K, scGCN was averagely 10 times slower than GraphCS mainly because their used CCA-MNN for graph construction is significantly slower than our used BKNN. The results demonstrated that our model could be extended to large-scale datasets in linear time complexity.

Discussion

With the tremendous increase of scRNA-seq datasets, it is feasible to transfer well-defined labels of existing single-cell datasets to newly generated single-cell datasets. In this study, we proposed a robust and scalable graph-based artificial intelligence model, which enables training the well-labeled single-cell data to annotate new data through robust knowledge transferring. We have demonstrated that GraphCS achieves significant improvements compared with 14 existing annotation methods in terms of performance and efficiency using the simulated, cross-platform, cross-species and cross-omics scRNA-seq datasets. Meanwhile, our model can be extended to large dataset in linear time complexity.

Although several commonly used cell annotation algorithms, such as Seurat and SingleR, also possess knowledge transferring functionalities, our model achieved superior results in terms of both performance and efficiency. Though another method, scGCN is also using the GNN method to annotate cells, the method relies on an expensive message-passing procedure to propagate information in each training epoch and thus has to include the full-batch during training. The method

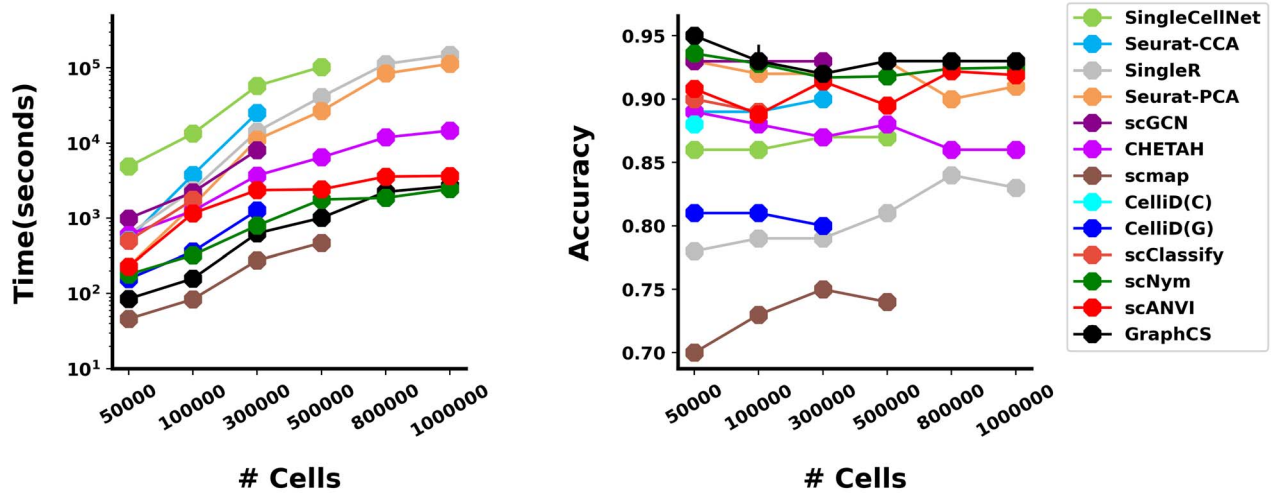


Figure 5. Comparison of different methods for the running time (left) and cell-type classification accuracy (right) on variably sized datasets.

inherently requires huge memory costs, preventing its applications to large datasets on GPU. In contrast, GraphCS pre-computes the information propagation via the approximate Generalized PageRank algorithm and trains the model with mini-batches, enabling linear complexity for a high speed and scalability on millions of cells. Meanwhile, we use VAT to incorporate the information of data distribution from unlabeled data to improve the generality of our model. We have demonstrated that our model outperformed scGCN in terms of average accuracy on all datasets and was 10 times faster than it on large datasets. In addition, the GBP module of GraphCS propagates feature information among similar cells based on the cell graph, resulting in batch effects removal due to similar cells within and between reference and query datasets sharing similar gene expressions.

With the rapid development of single-cell filed, several large-scale single-cell projects such as MCA, HCL and some other atlases [63–65] are established, where millions of sequenced cells have been well annotated. The tremendous accumulation of well-annotated scRNA-seq datasets can be used as high-quality reference datasets covering more cell types. Because of the scalable, fast and accurate performance, GraphCS is useful for transferring labels from these large scRNA-seq reference datasets to newly generated scRNA-seq data in a reasonable time. Second, our model is proven able to annotate cross-species datasets, which is useful to utilize similar well-studied species datasets for annotating new species. Third, with the decreasing scRNA-seq costs and international collaborations, extremely large datasets emerge for important problems such as the mouse brain [66, 67] and covid-19 datasets [68]. Such datasets require annotation methods that can remove batch effects and be scalable. On the other hand, even with the large-scale single-cell projects, novel cell types might still happen in the query dataset. In this scenario, the

predicted confidence scores for cell types aid to detect the unknown cell types, as proved in tumor datasets.

In spite of the superior performances, GraphCS can be improved in several aspects. First, our model ignores the relations between genes, which has been shown to improve the imputation of scRNA-seq data [40]. Second, the performance of our model is influenced by the constructed cell graph, and a high-quality graph can improve performance. Thus, the model may be useful for spatial transcriptomic data analysis [69, 70], where cells could be naturally connected through the provided spatial coordinates. Third, for unknown cell types, currently we have no good way to decide the optimal threshold, and users have to make decisions from prior knowledge. On the other hand, users can merge different datasets to reduce unknown cell types in the query dataset, as our model is shown not influenced by abundant cell types in the reference datasets. In conclusion, this study provided a new robust and scalable model to utilize known reference datasets for accurately classifying single cells in query datasets. This method will be particularly useful with the rapidly increasing annotated single-cell datasets.

Key Points

- With the launch of several large-scale single-cell projects, millions of sequenced cells have been annotated and it is promising to transfer labels from the annotated datasets to newly generated datasets. Graph neural network (GNN) is robust to learn cell relations. Vanilla GNNs need to pass information along the whole graph at each training epoch, which are difficult to process millions of cells due to the expensive costs of computations and memory.
- We proposed a scalable GNN-based method for accurate single-cell classification (GraphCS) by pre-calculating the diffused information via the

approximate Generalized PageRank algorithm (GBP), enabling sublinear complexity in computations.

- The used GBP algorithm in GraphCS enables training GNN networks through mini-batch, enabling training of large datasets on GPU with limited memory.
- GraphCS further reduces batch effects through the virtual adversarial training technique.
- GraphCS demonstrates superior performance on simulated, cross-platform, cross-species and cross-omics scRNA-seq datasets. The model could process 1 million cells within 50 min.

Supplementary Data

Supplementary data are available online at <https://academic.oup.com/bib/advance-article/doi/10.1093/bib/bbab570/6501353>

Code availability

All source code used in our experiments have been deposited at <https://github.com/biomed-AI/GraphCS>. The scRNA-seq datasets that support the findings of this study are available here: <https://drive.google.com/drive/folders/1ST0T90HcxCKuxOTmOvqCI-IyE2IY6YvM>.

Funding

This study has been supported by the National Key R&D Program of China (2020YFB0204803), National Natural Science Foundation of China (61772566), Guangdong Key Field R&D Plan (2019B020228001 and 2018B010109006), Introducing Innovative and Entrepreneurial Teams (2016ZT06D211), Guangzhou S&T Research Plan (20200-7030010).

References

1. 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**(4):346, e4–60.
2. Puram SV, Tirosh I, Parkh AS, et al. Single-cell transcriptomic analysis of primary and metastatic tumor ecosystems in head and neck cancer. *Cell* 2017;**171**(7):1611, e24–24.
3. Athanasiadis EI, Botthof JG, Andres H, et al. Single-cell RNA-sequencing uncovers transcriptional states and fate decisions in haematopoiesis. *Nat Commun* 2017;**8**(1):1–11.
4. Azizi E, Carr AJ, Plitas G, et al. Single-cell map of diverse immune phenotypes in the breast tumor microenvironment. *Cell* 2018;**174**(5):1293, e36–308.
5. Cusanovich DA, Hill AJ, Aghamirzaie D, et al. A single-cell atlas of in vivo mammalian chromatin accessibility. *Cell* 2018;**174**(5):1309, e18–24.
6. Muraro MJ, Dharmadhikari G, Grün D, et al. A single-cell transcriptome atlas of the human pancreas. *Cell Syst* 2016;**3**(4):385, e3–94.
7. T. M. Consortium. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature* 2018;**562**(7727):367–72.
8. Buenrostro JD, Corces MR, Lareau CA, et al. Integrated single-cell analysis maps the continuous regulatory landscape of human hematopoietic differentiation. *Cell* 2018;**173**(6):1535, e16–48.
9. Jaitin DA, Kenigsberg E, Keren-Shaul H, et al. Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. *Science* 2014;**343**(6172):776–9.
10. Gierahn TM, Wadsworth MH, II, Hughes TK, et al. Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput. *Nat Methods* 2017;**14**(4):395–8.
11. Lähnemann D, Köster J, Szczurek E, et al. Eleven grand challenges in single-cell data science. *Genome Biol* 2020;**21**(1):31. [10.1186/s13059-020-1926-6](https://doi.org/10.1186/s13059-020-1926-6).
12. Luecken MD, Theis FJ. Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol Syst Biol* 2019;**15**(6):e8746.
13. Mezger A, Klemm S, Mann I, et al. High-throughput chromatin accessibility profiling at single-cell resolution. *Nat Commun* 2018;**9**(1):1–6.
14. Buenrostro JD, Wu B, Litzenburger UM, et al. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature* 2015;**523**(7561):486–90.
15. Svensson V. Droplet scRNA-seq is not zero-inflated. *Nat Biotechnol* 2020;**38**(2):147–50.
16. Vieth B, Ziegenhain C, Parekh S, et al. powsimR: power analysis for bulk and single cell RNA-seq experiments. *Bioinformatics* 2017;**33**(21):3486–8.
17. Brennecke P, Anders S, Kim JK, et al. Accounting for technical noise in single-cell RNA-seq experiments. *Nat Methods* 2013;**10**(11):1093.
18. Lun AT, Marioni JC. Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data. *Biostatistics* 2017;**18**(3):451–64.
19. Regev A, Teichmann SA, Lander ES, et al. Science forum: the human cell atlas. *Elife* 2017;**6**:e27041.
20. Schaum N, et al. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris: the Tabula Muris Consortium. *Nature* 2018;**562**(7727):367.
21. Kiselev VY, Yiu A, Hemberg M. scmap: projection of single-cell RNA-seq data across data sets. *Nat Methods* 2018;**15**(5):359–62.
22. 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**(2):163–72.
23. 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**(16):e95–5.
24. Wang S, Pisco AO, McGeever A, et al. Leveraging the Cell Ontology to classify unseen cell types. *Nat Commun* 2021;**12**(1):5556.
25. Cortal A, Martignetti L, Six E, et al. Gene signature extraction and cell identity recognition at the single-cell level with cell-ID. *Nat Biotechnol* 2021;**39**(9):1095–102.
26. Alquicira-Hernandez J, Sathe A, Ji HP, et al. scPred: accurate supervised method for cell-type classification from single-cell RNA-seq data. *Genome Biol* 2019;**20**(1):1–17.
27. Tan Y, Cahan P. SingleCellNet: a computational tool to classify single cell RNA-Seq data across platforms and across species. *Cell Syst* 2019;**9**(2):207, e2–13.
28. Butler A, Hoffman P, Smibert P, et al. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* 2018;**36**(5):411–20. [10.1038/nbt.4096](https://doi.org/10.1038/nbt.4096).
29. Lin Y, Cao Y, Kim HJ, et al. scClassify: sample size estimation and multiscale classification of cells using single and multiple reference. *Mol Syst Biol* 2020;**16**(6):e9389.

30. Xu C, Lopez R, Mehlman E, et al. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Mol Syst Biol* 2021;**17**(1):e9620.
31. R. Lopez, J. Regier, M. B. Cole, et al. "Deep generative modeling for single-cell transcriptomics," *Nat Methods*, vol. **15**, no. 12, pp. 1053–8, Dec 2018, doi: [10.1038/s41592-018-0229-2](https://doi.org/10.1038/s41592-018-0229-2).
32. Kimmel JC, Kelley DR. Semisupervised adversarial neural networks for single-cell classification. *Genome Res* 2021;**31**(10):1781–93.
33. Zhang AW, O'Flanagan C, Chavez EA, et al. Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling. *Nat Methods* 2019;**16**(10):1007–15.
34. Hou R, Denisenko E, Forrest AR. scMatch: a single-cell gene expression profile annotation tool using reference datasets. *Bioinformatics* 2019;**35**(22):4688–95.
35. Pliner HA, Shendure J, Trapnell C. Supervised classification enables rapid annotation of cell atlases. *Nat Methods* 2019;**16**(10):983–6.
36. Johnson TS, Wang T, Huang Z, et al. LAMBDA: label ambiguous domain adaptation dataset integration reduces batch effects and improves subtype detection. *Bioinformatics* 2019;**35**(22):4696–706.
37. Ma F, Pellegrini M. ACTINN: automated identification of cell types in single cell RNA sequencing. *Bioinformatics* 2020;**36**(2):533–8.
38. Boufeua 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**(3):100914.
39. Zhou X, Chai H, Zeng Y, et al. scAdapt: virtual adversarial domain adaptation network for single cell RNA-seq data classification across platforms and species. *Brief Bioinform* 2021;**22**(6):bbab281.
40. Rao J, Zhou X, Lu Y, et al. Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks. *IScience* 2021;**24**(5):102393.
41. Zeng Y, Zhou X, Rao J, et al. Accurately clustering single-cell RNA-seq data by capturing structural relations between cells through graph convolutional network. In: *BIBM. Virtual Event, South Korea: IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2020*, 519–22.
42. Wang J, Ma A, Chang Y, et al. scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nat Commun* 2021;**12**(1):1–11.
43. Song Q, Su J, Zhang W. scGCN is a graph convolutional networks algorithm for knowledge transfer in single cell omics. *Nat Commun* 2021;**12**(1):1–11.
44. Kipf T, Welling M. Semi-supervised classification with graph convolutional networks. In: *ICLR. Toulon, France: International Conference on Learning Representations (ICLR), 2017*.
45. Cao J, Spielmann M, Qiu X, et al. The single-cell transcriptional landscape of mammalian organogenesis. *Nature* 2019;**566**(7745):496–502.
46. Datlinger P, Rendeiro AF, Boenke T, et al. Ultra-high throughput single-cell RNA sequencing by combinatorial fluidic indexing. *BioRxiv* 2019.
47. Chen M, Wei Z, Ding B, et al. Scalable graph neural networks via bidirectional propagation. *arXiv* 2020. arXiv:201015421.
48. Li P, Chien E, Milenkovic O. Optimizing generalized pagerank methods for seed-expansion community detection[J]. *Advances in Neural Information Processing Systems* 2019;**32**:11710–21.
49. L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:180203426*, 2018.
50. Sehanobish A, Ravindra NG, van Dijk D. Self-supervised edge features for improved graph neural network training. *arXiv preprint arXiv:200704777* 2020.
51. Wang B, Zhu J, Pierson E, et al. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nat Methods* 2017;**14**(4):414–6.
52. Hie B, Bryson B, Berger B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nat Biotechnol* 2019;**37**(6):685–91.
53. Szubert B, Cole JE, Monaco C, et al. Structure-preserving visualization of high dimensional single-cell datasets. *Sci Rep* 2019;**9**(1):1–10.
54. Song Q, Su J, Zhang W. scGCN is a graph convolutional networks algorithm for knowledge transfer in single cell omics. *Nat Commun* 2021;**12**(1):1–11.
55. Dudoit S, Fridlyand J, Speed TP. Comparison of discrimination methods for the classification of tumors using gene expression data. *J Am Stat Assoc* 2002;**97**(457):77–87.
56. Haghverdi L, Lun ATL, Morgan MD, et al. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol* 2018;**36**(5):421–7. [10.1038/nbt.4091](https://doi.org/10.1038/nbt.4091).
57. Barkas N, Petukhov V, Nikolaeva D, et al. Joint analysis of heterogeneous single-cell RNA-seq dataset collections. *Nat Methods* 2019;**16**(8):695–8.
58. Polański K, Young MD, Miao Z, et al. BBKNN: fast batch alignment of single cell transcriptomes. *Bioinformatics* 2020;**36**(3):964–5.
59. Miyato T, Maeda SI, Koyama M, et al. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Trans Pattern Anal Mach Intell* 2018;**41**(8):1979–93.
60. Klicpera J, Bojchevski A, Günnemann S. Predict then propagate: graph neural networks meet personalized pagerank. In: *ICLR. New Orleans, LA, USA: International Conference on Learning Representations (ICLR), 2019*.
61. Ouali Y, Hudelot C, Tami M. An overview of deep semi-supervised learning. *arXiv preprint arXiv:200605278* 2020.
62. Li C, Liu B, Kang B, et al. SciBet as a portable and fast single cell type identifier. *Nat Commun* 2020;**11**(1):1818.
63. He S, Wang LH, Liu Y, et al. Single-cell transcriptome profiling of an adult human cell atlas of 15 major organs. *Genome Biol* 2020;**21**(1):1–34.
64. Park J-E, Botting RA, Domínguez Conde C, et al. A cell atlas of human thymic development defines T cell repertoire formation. *Science* 2020;**367**(6480).
65. Wilk AJ, Rustagi A, Zhao NQ, et al. A single-cell atlas of the peripheral immune response in patients with severe COVID-19. *Nat Med* 2020;**26**(7):1070–6.
66. Saunders A, Macosko EZ, Wysoker A, et al. Molecular diversity and specializations among the cells of the adult mouse brain. *Cell* 2018;**174**(4):1015, e16–30.
67. 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**(6385):176–82.
68. Ren X, Wen W, Fan X, et al. COVID-19 immune features revealed by a large-scale single-cell transcriptome atlas. *Cell* 2021;**184**(7):1895, e19–913.
69. Crosetto N, Bienko M, Van Oudenaarden A. Spatially resolved transcriptomics and beyond. *Nat Rev Genet* 2015;**16**(1):57–66.
70. Qian X, Harris KD, Hauling T, et al. Probabilistic cell typing enables fine mapping of closely related cell types in situ. *Nat Methods* 2020;**17**(1):101–6.