Unsupervised Gene-Cell Collective Representation Learning with Optimal Transport

Jixiang Yu[1], **Nanjun Chen**[1], **Ming Gao**[2,3], **Xiangtao Li**[4], **Ka-Chun Wong**[1,5,6,*]

[1]Department of Computer Science, City University of Hong Kong, Hong Kong SAR
[2]School of Management Science and Engineering, Key Laboratory of Big Data Management Optimization and Decision of Liaoning Province, Dongbei University of Finance and Economics, Dalian, China
[3]Center for Post-doctoral Studies of Computer Science, Northeastern University, Shenyang, China
[4]School of Artificial Intelligence, Jilin University, Jilin, China
[5]Shenzhen Research Institute, City University of Hong Kong, Shenzhen, China
[6]Hong Kong Institute for Data Science, City University of Hong Kong, Hong Kong SAR
{jixiang.yu, nanjuchen2-c}@my.cityu.edu.hk, gm@dufe.edu.cn, lixt314@jlu.edu.cn, kc.w@cityu.edu.hk

Abstract

Cell type identification plays a vital role in single-cell RNA sequencing (scRNA-seq) data analysis. Although many deep embedded methods to cluster scRNA-seq data have been proposed, they still fail in elucidating the intrinsic properties of cells and genes. Here, we present a novel end-to-end deep graph clustering model for **s**ingle-**c**ell transcriptomics data based on unsupervised **G**ene-**C**ell collective representation learning and **O**ptimal **T**ransport (scGCOT) which integrates both cell and gene correlations. Specifically, scGCOT learns the latent embedding of cells and genes simultaneously and reconstructs the cell graph, the gene graph, and the gene expression count matrix. A zero-inflated negative binomial (ZINB) model is estimated via the reconstructed count matrix to capture the essential properties of scRNA-seq data. By leveraging the optimal transport-based joint representation alignment, scGCOT learns the clustering process and the latent representations through a mutually supervised self optimization strategy. Extensive experiments with 14 competing methods on 15 real scRNA-seq datasets demonstrate the competitive edges of scGCOT.

Introduction

Single-cell RNA-sequencing (scRNA-seq) technology can characterize individual cellular states in a high-throughput manner (Simmons et al. 2023), which enables researchers to elucidate the heterogenity among individual cells. Over the past years, scRNA-seq has become an important tool as the robustness and accessibility of scRNA-seq assays are improved continuously. Cell type identification has become a vital task in scRNA-seq data analysis, as it can imply different biological processes, such as cellular differentiation, lineage commitment, and gene regulation (Wan, Chen, and Deng 2022). Based on the transcriptome similarity, clustering has been proved to be an effect leverage to define cell types in an unbiased way (Kiselev, Andrews, and Hemberg 2019). However, although many classical clustering algorithms are robust and universal for tabular data, cluster analysis of scRNA-seq data remains a domain-specific challenge, owing to its unique high dimensionality characteristics (Kiselev, Andrews, and Hemberg 2019).

Essentially, scRNA-seq data is very sparse and has a large number of zero elements (Grün, Kester, and Van Oudenaarden 2014; Kulkarni et al. 2019). An intuitive way to alleviate the problem is to leverage a transformation to learn a latent representation on the original scRNA-seq data, and conduct clustering in the latent space. Therefore, multiple approaches are proposed (Butler et al. 2018; Žurauskienė and Yau 2016; Wang et al. 2018). However, linear transformation methods, such as PCA, suffer from the noise in the scRNA-seq data; and are difficult to capture the cell-cell relationships. Classical non-linear techniques also have limitations; their parameters are required to be manually defined by user and can strongly affect the downstream tasks (Kiselev, Andrews, and Hemberg 2019).

In recent years, with the advent of deep learning, deep embedded clustering methods have been developed to avoid the shortcomings of the conventional methods and to facilitate cell type identification in scRNA-seq data (Lopez et al. 2018; Wang and Gu 2018; Chen et al. 2020; Tian et al. 2019; Tran et al. 2021). These methods basically utilize an autoencoder to learn the latent representation of scRNA-seq data. Among those methods, one of the most representative one is scDeepCluster (Tian et al. 2019); it is a model-based deep autoencoder, combining the model of DCA (Eraslan et al. 2019) and the clustering algorithm of DEC (Xie, Girshick, and Farhadi 2016) together by means of Kullback–Leibler divergence to realize cell type identification in an end-to-end manner. However, these autoencoder-based algorithms focus only on studying the data itself, neglecting the relationships between cells, thus learns representations ineffectively. To cover this shortage, some graph neural network-based methods, such as GraphSCC (Zeng et al. 2020), scGAE (Luo et al. 2021), scDSC (Gan et al. 2022), scGNN (Wang et al. 2021), are developed to capture the cell relations and topological structure information. These methods integrate graph neural nework and autoencoder to simultaneously capture the complex relationship among cells and the intrinsic properties of cells. However, they only models the cell correla-

*Corresponding author

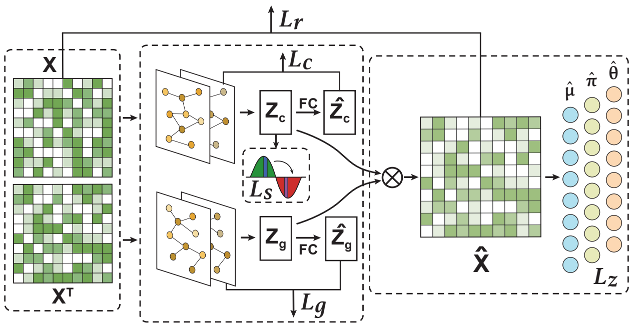

Figure 1: Overview of scGCOT. The overall model consists of two graph autoencoders to learn the latent representations of cells and genes, a ZINB-based decoder using the reconstructed count matrix to learn the inherent structure and properties of scRNA-seq data, and adopts OT to optimize the clustering process.

tions and may lack the latent representations of genes.

Given the drawbacks of the previous works, we propose a deep graph clustering algorithm based on unsupervised gene-cell collective representation learning and optimal transport (OT) theory, named scGCOT, as shown in Fig. 1. scGCOT learns cell correlations and gene correlations by attention-based graph autoencoders simultaneously, and learns to reconstruct the original gene expression count matrix. Subsequently, a ZINB model is formed in a seamless way via the reconstructed count matrix to capture the essential properties of scRNA-seq data, by estimating three main parameters of the ZINB distribution and maximizing its log likelihood. Then, a mutually supervised self-optimization learning strategy is leveraged to conduct the joint representation alignment, where the OT theory is used to align the clustering distribution and the auxiliary distribution.

The main contributions of our work are listed as follows:

- A deep graph embedded model named scGCOT is proposed for cell type identification on scRNA-seq data.
- Gene correlations are also taken into consideration in scGCOT. To the best of our knowledge, scGCOT is the first architecture that combines gene correlations and cell correlations together.
- The ZINB distribution is estimated in a seamless way to capture the essential properties of the data.
- The optimal transport theory is used for joint representation alignment, which can optimize the cluster assignment continuously.
- scGCOT is compared with a dozen of competing methods on 15 real scRNA-seq datasets. The experiment results demonstrate that scGCOT outperforms all of the other baselines.

Related Works

Deep embedded clustering methods ususally levearge a deep autoencoder to learn a feature representation in the latent space via minimizing the mean square error (MSE) of the raw data and the reconstructed data. For example, scDeepCluster combines DCA (Eraslan et al. 2019) and DEC (Xie, Girshick, and Farhadi 2016) to learn the latent feature representation and cluster assignment simultaneously. scVI (Lopez et al. 2018) uses stochastic optimization in a deep autoencoder to aggregate information and to approximate the distributions that underlies the observed expression values. scziDesk (Chen et al. 2020) adopts a weighted soft K-means algorithm for data points in the latent space to achieve high quality cell clustering. DESC (Li et al. 2020) leverages a deep autoencoder to learn the nonlinear mapping from the original feature space to the low dimensional feature space iteratively. Generally, these methods integrates feature representation learning and cluster assignment to enhance the final clustering performance. However, they only consider the gene expression information, without explicitly characterizing the relationship and structural information among cells.

To this end, multiple deep graph embedded clustering methods are proposed to learn the structural information and correlations between cells. For instance, scGNN (Wang et al. 2021) inserts a Gaussian mixture model into the graph neural networks to capture various heterogeneous gene expression patterns. To better learn the cell topology, scTAG (Yu et al. 2022) adopts the topological adaptive graph convolutional networks (Du et al. 2017) to extract the structural information of scRNA-seq data at different scales. scDFC (Hu et al. 2023) applies the graph attention network (Veličković et al. 2017) to reduce the noise effect brought in to the cell graph. As a novel approach, contrastive learning is introduced for cell clustering by scNAME (Wan, Chen, and Deng 2022) and scDCCA (Wang et al. 2023a). Overall, existing state-of-the-art graph embedded clustering methods extract the cell correlations to realize a better understanding of gene expression.

Methodology

Pre-processing

We take the original scRNA-seq gene expression count matrix $\mathbf{X}^o \in \mathbb{R}^{M \times N}$ as the input, where M and N are the numbers of cells and genes. We first filter out the genes that are not expressed in cells. Considering the different range of count values for each cell, we normalize the expression for each cell to transform the discrete value to continuous using the following equation:

$$\mathcal{M}\left(\mathbf{X}_{ij}^o\right) = \ln\left(\text{median}(\mathbf{X}^o)\frac{\mathbf{X}_{ij}^o}{\sum_{n=1}^{N}\mathbf{X}_{in}^o}\right), \quad (1)$$

where $\text{median}(\mathbf{X}^o)$ is the median of the filtered gene expression matrix. We normalize the expression for $\mathbf{X}_{ij}^o$ using the sum of N genes expressions for cell i. Then we adopt the scannpy package (Wolf, Angerer, and Theis 2018) to rank the top n highly-variable genes using the normalized dispersion values to get the final processed scRNA-seq gene expression matrix. For datasets with less than one thousand cells, we follow scGAC (Cheng and Ma 2022) to conduct a network enhancement technique to reduce the noise.

Cell and Gene Graphs

Previous deep learning methods for scRNA-seq data clustering only leverages the cell correlations without considering the gene correlations. Therefore, we try to extract gene correlation representations along with the ordinary cell correlation representations. Given the preprocessed scRNA-seq gene expression data $\mathbf{X} \in \mathbb{R}^{M \times n}$, where $\mathbf{X}_{ij}$ denotes the expression count of j-th highly-variable gene in i-th cell, we first construct the cell correlation graph. Similar to previous works, the KNN algorithm is employed to construct the cell graph $\mathcal{G}_c$ and each node in the graph represents a cell. Specifically, an edge between node a and node b exists if a is b's neighbor within top-K shortest distance. Subsequently, we use the same method to construct the gene correlation graph $\mathcal{G}_g$ based on $\mathbf{X}^{\mathrm{T}}$, where $(\mathbf{X}^{\mathrm{T}})_{ij}$ represents the expression count of i-th gene in the j cell. The reason to construct a gene correlation graph on $\mathbf{X}^{\mathrm{T}}$ is biologically intuitive: some genes may be expressed across multiple types of cells and some may not, and thus $\mathbf{X}^{\mathrm{T}}$ can be treated as a cell expression profile for each gene.

Attention-based Graph Autoencoders

Since scRNA-seq data are inherently noisy (Grün, Kester, and Van Oudenaarden 2014), it is reasoned that many edges in the correlation graphs may be incorrectly connected. Therefore, an attention mechanism is needed to ensure that accurate information of the graph structure is obtained. To capture the precise intrinsic structure and node features in the cell graph and the gene graph, we develop two graph attention autoencoders based on Graph Transformer (Shi et al. 2020) for cell and gene representation learning. The gene expression matrix $\mathbf{X}$ and the cell expression profile $\mathbf{X}^{\mathrm{T}}$, together with their corresponding adjacency matrices, $\mathbf{A}_c$ and $\mathbf{A}_g$, are used as inputs.

For cell graph autoencoder learning, considering the graph encoder has L layers, we assume each node has a hidden dimension of d_l, which means the input data of the l-th hidden layer is $\mathbf{x}_i^{(l)} \in \mathbb{R}^{d_l}$. Then the cell graph encoding process is defined as follows:

$$\mathbf{y}_i^{(l)} = \mathbf{W}_1 \mathbf{x}_i^{(l)} + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} \mathbf{W}_2 \mathbf{x}_j^{(l)}, \tag{2}$$

where $\mathbf{y}_i^{(l)}$ represents the l-th output feature map of the cell graph encoder; $\mathbf{W}_1$ and $\mathbf{W}_2$ are the learnable weights of the encoder layer; and the attention coefficient α_{ij} is computed via dot product attention:

$$\alpha_{i,j} = \mathrm{softmax}\left(\frac{(\mathbf{W}_3 \mathbf{x}_i^{(l)})^{\mathrm{T}} (\mathbf{W}_4 \mathbf{x}_j^{(l)})}{\sqrt{d_l}} \right), \tag{3}$$

where $\mathbf{W}_3$ and $\mathbf{W}_4$ are learnable weights of the attention calculation. After each graph attention calculation, a nonlinear activation function is applied to the output $\mathbf{y}_i^{(l)}$:

$$\mathbf{x}_i^{(l+1)} = \sigma(\mathbf{y}_i^{(l)}), \tag{4}$$

where $\sigma(\cdot) = \max(0, x)$ denotes the ReLU function. To simplify the equations, we abbreviate the cell graph encoder calculation as:

$$\mathbf{Z}_c = f_E^1(\mathbf{X}). \tag{5}$$

Ideally, the intrinsic correlations of cells are preserved in the latent embedding space, therefore, we define the decoder part by simply combining a fully-connected layer and a self inner product, as formulated below:

$$\hat{\mathbf{Z}}_c = \mathbf{W}_c \mathbf{Z}_c + \mathbf{b}_c, \tag{6}$$

$$\hat{\mathbf{A}}_c = \phi(\hat{\mathbf{Z}}_c^{\mathrm{T}} \hat{\mathbf{Z}}_c), \tag{7}$$

where $\hat{\mathbf{A}}_c$ is the reconstructed adjacency matrix of the cell correlation graph; $\phi(\cdot)$ is the sigmoid activation function. To constrain the learning process, the MSE loss between the reconstructed adjacency matrix $\hat{\mathbf{A}}_c$ and the original adjacency matrix of the cell graph $\mathbf{A}_c$ is calculated as the reconstruction loss:

$$\mathcal{L}_c = ||\mathbf{A}_c - \hat{\mathbf{A}}_c||_2^2. \tag{8}$$

Similarly, we deploy another gene graph autoencoder with the same structure as the cell graph autoencoder for gene representation learning, denoted as f_E^2. Then the gene latent space given by the autoencoder is formulated as:

$$\mathbf{Z}_g = f_E^2(\mathbf{X}^{\mathrm{T}}). \tag{9}$$

And the reconstructed adjacency matrix of the gene correlation graph is obtained in the same way as the cell decoder:

$$\hat{\mathbf{Z}}_g = \mathbf{W}_g \mathbf{Z}_g + \mathbf{b}_g, \tag{10}$$

$$\hat{\mathbf{A}}_g = \phi(\hat{\mathbf{Z}}_g^{\mathrm{T}} \hat{\mathbf{Z}}_g). \tag{11}$$

Then the reconstruction loss of the gene autoencoder is calculated to constrain the learning process:

$$\mathcal{L}_g = ||\mathbf{A}_g - \hat{\mathbf{A}}_g||_2^2. \tag{12}$$

ZINB-based Gene-Cell Collective Representation Learning Decoder

In order to capture the global structure of the gene-cell expression count matrix $\mathbf{X}$, we integrate a decoder in scGCOT using the latent feature embedding $\mathbf{Z}_g$ and $\mathbf{Z}_c$ collectively to better obtain the structure of scRNA-seq data. We first let the latent representation $\mathbf{Z}_g$ and $\mathbf{Z}_c$ reconstruct the original gene expression matrix $\mathbf{X}$ collectively by their inner product and compute the reconstruction loss:

$$\hat{\mathbf{X}} = \mathbf{Z}_c^{\mathrm{T}} \mathbf{Z}_g, \tag{13}$$

$$\mathcal{L}_r = ||\mathbf{X} - \hat{\mathbf{X}}||_2^2. \tag{14}$$

scRNA-seq gene expression count matrix basically conforms to three characteristics: 1) discrete; 2) variance greater than the mean; 3) many entries have a zero value. These characteristics can be modeled as a zero-inflated negative binomial (ZINB) distribution. Formally, ZINB is formulated with a mean value μ and a dispersion parameter θ of the negative binomial distribution, along with an additional coefficient π that represents the weight of the probability of the dropout events in scRNA-seq data:

$$\begin{aligned} \mathrm{ZINB}(\mathbf{X}|\pi, \mu, \theta) = \pi \delta_0(\mathbf{X}) + (1 - \pi) \times \frac{\Gamma(\mathbf{X} + \theta)}{\mathbf{X}! \Gamma(\theta)} \\ \times \left(\frac{\theta}{\theta + \mu} \right)^{\theta} \times \left(\frac{\mu}{\theta + \mu} \right)^{\mathbf{X}}, \end{aligned} \tag{15}$$

where μ and θ represent the mean and dispersion in the negative binomial distribution, respectively; π denotes the weight of the zero components. To capture the characteristics of scRNA-seq data, we assume the reconstructed $\hat{\mathbf{X}}$ also follows a ZINB distribution. The three parameters μ, θ, and π of the ZINB distribution is estimated by constructing three parallel output fully-connected layers:

$$\hat{\pi} = \mathrm{sigmoid}(\mathbf{W}_\pi \hat{\mathbf{X}}), \tag{16}$$

$$\hat{\theta} = \exp(\mathbf{W}_\theta \hat{\mathbf{X}}), \tag{17}$$

$$\hat{\mu} = \exp(\mathbf{W}_\mu \hat{\mathbf{X}}), \tag{18}$$

where $\mathbf{W}_\pi$, $\mathbf{W}_\theta$, and $\mathbf{W}_\mu$ are the corresponding weight of the three parameter estimators; and the selection of the activation function depends on the range and the definition of the three parameters: since π denotes the weight of the point mass at zero, which is in the range of $[0, 1]$, sigmoid is chosen for $\hat{\pi}$ estimation; given the non-negative essence of the paramter μ and θ, we use the exponential function. Subsequently, the loss function of the ZINB parameter estimation is defined as the negative log-likelihood of the ZINB distribution:

$$\mathcal{L}_z = -\log(\mathrm{ZINB}(\mathbf{X}|\hat{\pi}, \hat{\theta}, \hat{\mu})). \tag{19}$$

OT-based Joint Representation Alignment

To enhance the representation of cluster assignment, we introduce a novel approach, named mutually supervised self-optimization strategy, which takes the advantage of OT. This strategy aligns the cluster distribution and the auxiliary distribution continuously, thereby optimizing the clustering process. In our framework, we consider the clustering distribution as $\mathbb{Q}$ and the auxiliary distribution as $\mathbb{P}$ same as (Xie, Girshick, and Farhadi 2016). To quantify the clustering distribution of the latent embedding $\mathbf{Z}_c$, we define q_{iu}, an element in $\mathbb{Q}$, as:

$$q_{iu} = \frac{\left(1 + \|z_i - \mu_u\|^2\right)^{-1}}{\sum_r \left(1 + \|z_i - \mu_r\|^2\right)^{-1}}, \tag{20}$$

In our approach, each cell i is represented by its latent representation z_i, while μ_u represents the cluster centroid representation obtained from the pseudo-labels generated by spectral clustering, given the cluster number r. To ensure the prominence of high-confidence data points, an auxiliary distribution $\mathbb{P}$ is introduced based on q_{iu}, which is defined as follows:

$$p_{iu} = \frac{q_{iu}^2 / \sum_i q_{iu}}{\sum_r \left(q_{ir}^2 / \sum_i q_{ir}\right)}. \tag{21}$$

Additionally, we denote the divergence between the distributions $\mathbb{Q}$ and $\mathbb{P}$ as an entropically regularized OT problem, formulated as follows:

$$OT(\mathbf{M}, \mathbb{Q}, \mathbb{P})^\epsilon = \min_\gamma \langle \gamma, \mathbf{M} \rangle_F + \epsilon \cdot \sum_{i,j} \gamma_{i,j} \log(\gamma_{i,j}), \quad \text{subject to } \gamma \mathbf{1} = \mathbb{Q}; \gamma^{\mathrm{T}} \mathbf{1} = \mathbb{P}; \gamma \geq 0, \tag{22}$$

Dataset	Cell	Gene	Types	Ref.
p3cl	2609	17561	3	(Dong et al. 2021)
Muraro	2122	19046	9	(Muraro et al. 2016)
Qx_L._M.	3909	23341	6	(Schaum et al. 2018)
Qs_Diaph.	870	23341	5	(Schaum et al. 2018)
Qs_Heart	4365	23341	8	(Schaum et al. 2018)
Qs_L._M.	1090	23341	6	(Schaum et al. 2018)
Qs_Lung	1676	23341	11	(Schaum et al. 2018)
Young	5685	28205	11	(Young et al. 2018)
Adam	3660	23797	8	(Adam, Potter, and Potter 2017)
Plasschaert	6977	14561	8	(Plasschaert et al. 2018)
Chen	12089	23284	46	(Chen et al. 2017)
Xin	1601	39851	4	(Xin et al. 2016)
Pollen	301	21721	11	(Pollen et al. 2014)
Guo	272	8772	7	(Guo et al. 2015)
Human	1289	8772	5	(Petropoulos et al. 2016)

Table 1: Summary of the fifteen datasets used in this study

where $\mathbf{M}$ denotes the transport cost matrix from the distribution $\mathbb{Q}$ to $\mathbb{P}$, defined here simply as the Euclidean distance, and the amount of transport from $\mathbb{Q}$ to $\mathbb{P}$ is γ; ϵ is the regularization parameter that controls the entropic regularization term. OT aims to minimize the expected transport cost subject to the marginal constraints. Finally, we utilize the Sinkhorn divergence (Genevay, Peyré, and Cuturi 2018) to align the two finite discrete distributions. The Sinkhorn divergence, denoted as $\mathcal{L}_s$, is defined as follows:

$$\mathcal{L}_s = \mathrm{S}_\epsilon(\mathbb{Q}, \mathbb{P}) := OT(\mathbf{M}, \mathbb{Q}, \mathbb{P})^\epsilon - \frac{1}{2}\left(OT(\mathbf{M}, \mathbb{Q}, \mathbb{Q})^\epsilon + OT(\mathbf{M}, \mathbb{P}, \mathbb{P})^\epsilon\right). \tag{23}$$

Here, the auxiliary distribution $\mathbb{P}$ is based on $\mathbb{Q}$, and $\mathbb{Q}$ is expected to be aligned with $\mathbb{P}$ to prioritize high-confidence data points. We continuously optimize this alignment until maximum number of iterations.

Overall Training Process

The overall training process of scGCOT contains two stages, namely, embedding learning stage, and representation alignment and cluster assignment stage. In the embedding learning stage, the loss function for training is defined as:

$$\mathcal{L}_1 = \lambda_1 \mathcal{L}_c + \lambda_2(\mathcal{L}_g + \mathcal{L}_r) + \lambda_3 \mathcal{L}_z, \tag{24}$$

where λ_1, λ_2, λ_3 are the weight coefficients for each loss. After scGCOT learns the latent embedding for certain epochs, it turns to the representation alignment and cluster assignment stage, in which the loss functions are combined following (Kendall, Gal, and Cipolla 2018):

$$\mathcal{L}_2 = \lambda_1^{'} \mathcal{L}_c + \frac{1}{2\sigma_1^2}\mathcal{L}_z + \frac{1}{2\sigma_2^2}\mathcal{L}_s + \log \sigma_1 \sigma_2, \tag{25}$$

where σ_1 and σ_2 are two parameters to balance the loss $\mathcal{L}_z$ and $\mathcal{L}_s$.

Experiments and Results

Data Source and Baselines

As shown in Table 1, 15 real datasets are selected for evaluation. These datasets range in size from hundreds to thousands and come from different platforms. Meanwhile, the

	Methods	Muraro	Human	Qs_Diaph.	Qs_Heart	Chen	Qs_Lung	Young	Adam	Plass.	Qx_L._M.	Qs_L._M.	p3cl
NMI↑	K-means****	0.3683	0.5916	0.1669	0.2645	0.4512	0.2346	0.1780	0.1326	0.3897	0.2821	0.1256	0.5338
	Spectral***	0.8266	0.5995	0.3380	0.4647	0.6319	0.4136	0.3671	0.1000	0.5412	0.8442	0.3337	0.9350
	DCA****	0.7301	0.5146	0.6358	0.5840	0.7182	0.6708	0.6643	0.7261	0.4921	0.7262	0.6173	0.4797
	scDeepCluster****	0.7356	0.4695	0.7864	0.6667	0.6774	0.6840	0.5953	0.6493	0.5916	0.8022	0.7061	0.9397
	GraphSCC****	0.6326	0.5951	0.7100	0.8321	0.7167	0.6872	0.6377	0.6429	0.7747	0.8177	0.9287	0.9544
	scziDesk****	0.7806	0.6082	0.9051	0.8545	0.6899	*0.8068*	0.7325	0.8164	0.6469	0.9239	*0.9520*	0.9653
	scGAE****	0.5789	0.4271	0.3526	0.3094	-	0.3971	0.2986	0.0795	0.2902	0.4909	0.1279	0.3350
	scDSC****	0.7707	0.5644	0.9363	0.8700	0.6594	0.7124	0.5705	0.3100	0.7333	0.7994	0.8647	0.8541
	scGAC****	0.5519	0.5577	0.9355	*0.9058*	-	0.7872	0.4822	0.2880	0.7344	*0.9670*	0.9407	0.9342
	scNAME**	**0.8783**	*0.6163*	*0.9570*	0.8868	0.7454	0.8003	*0.7653*	0.8459	0.7453	0.9505	**0.9584**	*0.9677*
	CAPKM++2.0***	0.7467	0.5978	0.9334	0.8319	*0.7598*	0.7722	0.7630	0.6891	**0.8590**	0.8867	0.7294	0.9609
	scDFC****	0.5261	0.5086	0.8280	0.7365	-	0.6613	0.4664	0.3736	-	0.7968	0.7240	0.9333
	scBGEDA****	0.8315	0.5864	0.9412	0.8964	0.6089	0.7773	0.7355	*0.8492*	0.6409	0.9631	0.9446	0.9651
	scDCCA****	0.7524	0.5124	0.2135	0.4261	0.6858	0.3770	0.4946	0.2822	0.6426	0.8475	0.2250	0.8436
	scGCOT (Ours)	*0.8428*	**0.6265**	**0.9613**	**0.9216**	**0.7661**	**0.8326**	**0.7888**	**0.8513**	*0.8325*	**0.9693**	0.9493	**0.9685**
	Methods	Muraro	Human	Qs_Diaph.	Qs_Heart	Chen	Qs_Lung	Young	Adam	Plass.	Qx_L._M.	Qs_L._M.	p3cl
ARI↑	K-means****	0.1786	0.5034	0.0794	0.1477	0.2771	0.0913	0.0584	0.0177	0.2508	0.1130	0.0457	0.3840
	Spectral****	0.6436	0.4315	0.2773	0.3345	0.2630	0.2493	0.2253	0.0368	0.3541	0.8976	0.2409	0.9603
	DCA****	0.4579	0.1863	0.2866	0.2113	0.4801	0.2623	0.3774	0.5530	0.1939	0.4094	0.2647	0.1523
	scDeepCluster****	0.6371	0.3238	0.7174	0.5125	0.3280	0.4504	0.4749	0.5595	0.3918	0.7202	0.5545	0.9705
	GraphSCC****	0.4506	0.4641	0.5661	0.8874	*0.6805*	0.5027	0.4352	0.4257	0.8305	0.7870	0.9608	0.9791
	scziDesk***	0.7037	0.4650	0.9272	0.8323	0.2935	0.7112	0.6371	0.7884	0.4867	0.9541	0.9745	0.9841
	scGAE****	0.1578	0.1526	0.1067	0.0351	-	0.1092	0.0422	0.0100	0.0231	0.0925	0.0291	0.0388
	scDSC***	0.7491	**0.4750**	0.9704	0.9360	0.5916	0.6994	0.3796	0.0000	0.7903	0.6839	0.9228	0.9279
	scGAC****	0.3481	0.3734	0.9648	*0.9403*	-	0.6342	0.2788	0.1149	0.8317	*0.9836*	0.9600	0.9645
	scNAME**	**0.8962**	0.4650	*0.9772*	0.8681	0.4908	0.6757	0.6130	*0.8306*	0.7315	0.9469	**0.9782**	*0.9852*
	CAPKM++2.0***	0.6567	0.4581	0.9547	0.8476	0.6744	0.6612	0.6334	0.5168	**0.8974**	0.8862	0.5357	0.9824
	scDFC****	0.3304	0.2695	0.9022	0.7077	-	0.6146	0.3203	0.2224	-	0.7493	0.6410	0.9641
	scBGEDA***	*0.8897*	*0.4747*	0.9667	0.9399	0.3096	*0.7277*	*0.6522*	0.8202	0.6761	0.9753	*0.9715*	0.9846
	scDCCA****	0.5550	0.3762	0.2138	0.3803	0.3947	0.2852	0.3709	0.1578	0.5150	0.7580	0.1660	0.9102
	scGCOT (Ours)	0.8811	0.4685	**0.9812**	**0.9581**	**0.7066**	**0.7887**	**0.7191**	**0.8475**	*0.8659*	**0.9865**	0.9707	**0.9855**

Table 2: The NMI and ARI comparison of scGCOT and 14 other baseline methods. The bold font indicates the best performance on one specific dataset, and the italic font indicates the second best performance. The significance levels are obtained using one-sided Wilcoxon signed-rank test. - indicates the algorithm cannot be run on a single RTX 4090 with 24 GB GPU memory (out of memory). The performance results on 3 other datasets are shown in Table 3.

number of cell types in the datasets are also various, ranging from 3 to 46. The annotation of cell types from the original publications is used as the ground truth. We select 14 other methods for comparison, among them 12 are the state-of-the-art methods, as listed below:

- **Deep Graph Embedded Methods**: scDCCA (Wang et al. 2023a), scBGEDA (Wang et al. 2023b), scDFC (Hu et al. 2023), scNAME (Wan, Chen, and Deng 2022), scGAC (Cheng and Ma 2022), scDSC (Gan et al. 2022), scGAE (Luo et al. 2021), GraphSCC (Zeng et al. 2020).
- **Deep Embedded Methods**: scziDesk (Chen et al. 2020), scDeepCluster (Tian et al. 2019), DCA (Eraslan et al. 2019)
- **Non-Deep Learning Methods**: CAPKM++2.0 (Li and Wang 2023), K-means, Spectral Clustering.

Implementation Details

All experiments are conducted on a Ubuntu 20.04 server equipped with 128GB memory and two RTX 4090 GPUs. The proposed scGCOT is constructed with PyTorch 2.0.0, PyG 2.3.0, and Python 3.10.11. In the proposed scGCOT method, the cell graph and gene graph are constructed using KNN algorithm with the nearest neighbor parameter $K = 15$ and the number of highly variable genes $n = 500$. In the graph autoencoders, f_E^1 and f_E^2 are both set as a two-layer Graph Transformer network, with the hidden dimensions of 128 and 15, respectively. The output dimension of the fully-connected networks to obtain $\hat{\mathbf{Z}}_c$ and $\hat{\mathbf{Z}}_g$ are both set to 32. Our algorithm consists of 300 epochs of embedding learning and 100 epochs of representation alignment and cluster assignment learning. The loss weights $\{\lambda_1, \lambda_2, \lambda_3\}$ and $\{\lambda_1^{'}, \frac{1}{2\sigma_1^2}, \frac{1}{2\sigma_2^2}\}$ are set to $\{1, 0.3, 1\}$ and $\{0.3, 1.5, 2\}$, respectively. Our model is optimized using equation (24) and the Adam algorithm with the learning rate 5e-4 in embedding learning stage. In the representation alignment and cluster assignment stage, we use equation (25) to train the model and let the learning rate increase from 1e-7 to 1e-4 linearly in the first half of total epochs and decrease linearly to 1e-7 in the second half. The parameters of other baseline methods remain as the default.

	Methods	Xin	Pollen	Guo	AVG
NMI↑	K-means****	0.2228	0.7613	0.4050	0.3405
	Spectral***	**0.7524**	0.8833	0.3846	0.5610
	DCA****	0.3831	*0.9229*	0.4002	0.6177
	scDeepCluster****	0.3006	0.9049	*0.4035*	0.6609
	GraphSCC****	0.3402	0.9053	0.2884	0.6976
	scziDesk****	0.5739	0.9047	0.3448	0.7670
	scGAE****	0.3506	0.8091	0.3310	0.3452
	scDSC****	0.5690	0.6985	0.3237	0.6824
	scGAC****	0.3960	0.8004	0.3383	0.6413
	scNAME**	0.5697	0.9132	0.2947	*0.7930*
	CAPKM++2.0***	0.6081	0.9219	0.3140	0.7583
	scDFC****	*0.6219*	0.8508	0.2381	0.5510
	scBGEDA****	0.5733	0.9083	0.3707	0.7728
	scDCCA****	0.5817	0.6639	0.2114	0.5173
	scGCOT (Ours)	0.6088	**0.9294**	**0.4896**	**0.8226**
	Methods	Xin	Pollen	Guo	AVG
ARI↑	K-means****	0.1774	0.5435	0.2550	0.2082
	Spectral****	0.6298	0.7864	0.2241	0.4370
	DCA****	0.1406	0.8491	0.2296	0.3370
	scDeepCluster****	0.2770	0.8412	0.2723	0.5354
	GraphSCC****	0.3886	0.8894	0.0856	0.6222
	scziDesk***	0.6454	0.8663	0.2290	0.6999
	scGAE****	0.0682	0.5654	0.1598	0.1060
	scDSC***	*0.7177*	0.4179	0.1856	0.6298
	scGAC****	0.4483	0.6267	0.1513	0.5747
	scNAME**	0.6079	0.8861	0.1212	0.7383
	CAPKM++2.0***	0.7107	*0.9065*	0.0988	0.6947
	scDFC****	**0.7214**	0.8770	0.1018	0.4948
	scBGEDA***	0.6987	0.8835	*0.2737*	*0.7496*
	scDCCA****	0.4510	0.5005	0.1066	0.4094
	scGCOT (Ours)	0.7074	**0.9184**	**0.3402**	**0.8084**

Table 3: The NMI and ARI comparison of scGCOT and 14 other baseline methods on other datasets.

Clustering Performance

To evaluate the performance of scGCOT well, we adopt two commonly-used metrics in clustering, namely, the Normalised Mutual Information (NMI) and the Adjusted Rand Index (ARI). The higher NMI and ARI indicate the clustering performance is better. The clustering performance of scGCOT with the comparision to the baseline methods on 15 scRNA-seq datasets is presented in Table 2 and Table 3. We run all the methods for five times and take the average performance. As shown in Table 2 and Table 3, compared to the latest state-of-the-art methods, scGCOT obtains the best NMI on 11 out of 15 datasets, and also obtains the best ARI on 10 out of 15 datasets. Furthermore, considering the second best NMI and ARI, scGCOT works very well on 13 out of 15 datasets for NMI, and on 11 out of 15 datasets for ARI. The overall average NMI and ARI on the all datasets of scGCOT are 0.8226 and 0.8084, respectively. Both the NMI and ARI across the all datasets are the highest among all the baselines, as shown in the “AVG” column in Table 3. Meanwhile, we notice that deep embedded methods, such as DCA, scDeepCluster, and scziDesk cannot obtain a stable performance across different datasets. A possible reason may be that they are solely based on gene expression count matrix and cannot fully capture the characteristics of the highly sparse scRNA-seq data. Compared to other state-of-the-art deep graph embedded methods, such as scDFC and scDCCA, scGCOT can also generally get higher NMI and ARI, demonstrating its competitive edges.

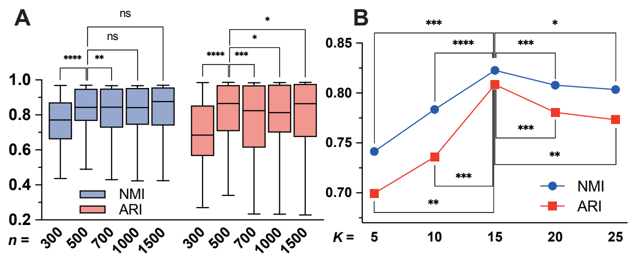

Figure 2: Parameter analysis results. The p-values are obtained via one-sided Wilcoxon signed-rank test. (A) The boxplot depicting the clustering performance of scGCOT given different numbers of n. (B) The line chart depicting the clustering performance of scGCOT given different numbers of K.

Parameter Analysis

In this section, we conduct an analysis on the number of highly variable genes n and the K in the KNN algorithm, due to the significance of these two parameters in single cell data analysis. First, to explore the impact of the number of the selected highly variable genes, we run scGCOT on all the datasets described in Table 1 for five times with highly variable genes ranging from 300 to 1500. Fig. 2A shows the box plot of the average NMI and ARI on 15 datasets with highly variable genes of 300, 500, 700, 1000, 1500. In terms of NMI, scGCOT performs well when selecting 500, 1000, and 1500 highly variable genes. However, when selecting $n = 500$, the ARI given by scGCOT has statistically significant differences compared to other conditions. Therefore, we choose $n = 500$. Subsequently, we explore the effect of the K neighbor parameter in the KNN algorithm to construct the cell graph and the gene graph, given the K in $\{5, 10, 15, 20, 25\}$. Fig. 2B shows that the two metrics first increase fast from the condition $K = 5$ to $K = 10$, and reach the peak performance with $K = 15$, then have a slight drop with K larger than 15. Hence, we set $K = 15$ in scGCOT.

Ablation Study

To fully investigate the effect of the proposed components in scGCOT, we conduct ablation experiments on it. In particular, we consider the following ablation conditions: 1) remove $\mathcal{L}_g$; 2) remove the reconstruction loss $\mathcal{L}_r$; 3) remove $\mathcal{L}_z$; 4) replace the graph transformer layers stated in eq. (2) and (3) with graph convolutional layers; 5) remove $\mathcal{L}_s$; the NMI and ARI results are shown in Table 4. It is evident that each component in scGCOT helps improve the clustering performance. From Table 3 we observe that removing $\mathcal{L}_z$ in the training process has the largest effect, where the ZINB properties cannot be well-reserved. The deletion of $\mathcal{L}_g$ and

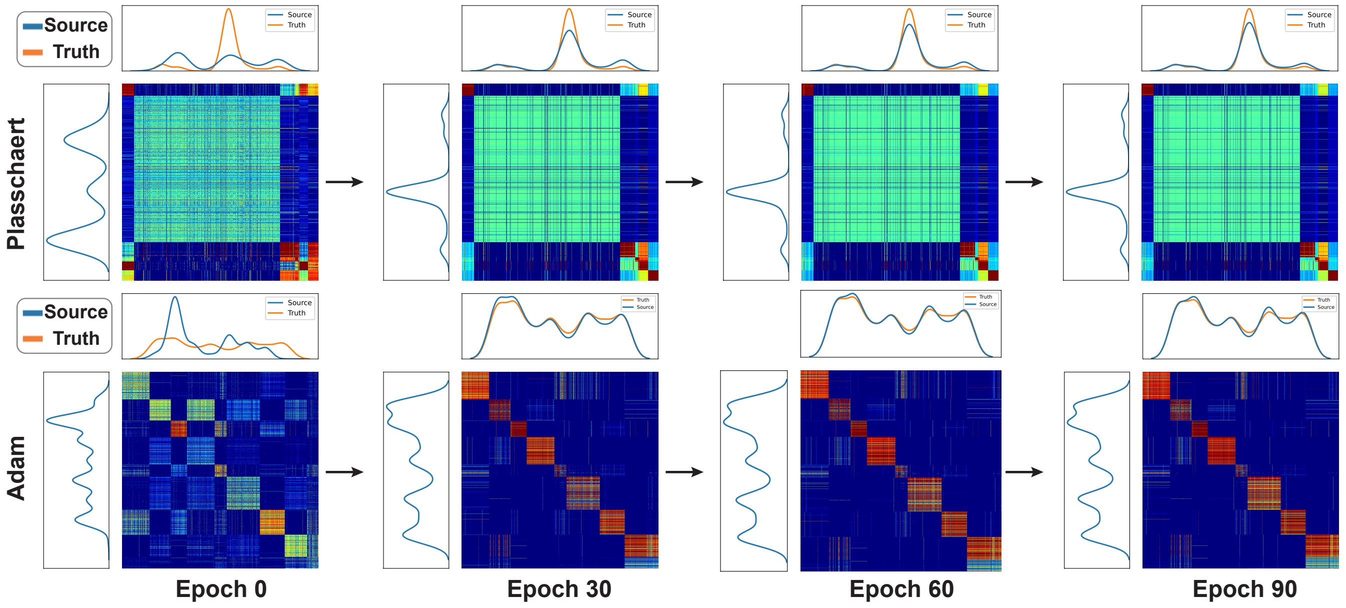

Figure 3: An intuitive illustration depicting the changing process of the OT plan with clustering distribution density of $\mathbb{Q}$ (shown on the top with the ground truth distribution) and $\mathbb{P}$ (shown on the left). To show the distribution density alignment process clearly on *Plasschaert*, the density and OT plan of two large and easy to learn categories are not drawn.

Method	NMI	ARI
scGCOT w/o $\mathcal{L}_g$	0.8104	0.7864
scGCOT w/o $\mathcal{L}_r$	0.7990	0.7658
scGCOT w/o $\mathcal{L}_z$	0.7733	0.7139
scGCOT w/o $\mathcal{L}_s$	0.7976	0.7648
scGCOT w/o attention	0.7906	0.7544
scGCOT (Ours)	**0.8226**	**0.8084**

Table 4: Ablation study measured by NMI and ARI values

$\mathcal{L}_r$ also let the performance of scGCOT drop, which may reflect that $\mathcal{L}_g$ and $\mathcal{L}_r$ can help the ZINB estimation. The deletion of attention mechanism and $\mathcal{L}_s$ also affects the overall performance, verifying their importance.

OT Analysis and Visualization

To further illustrate the effect of the OT-based representation alignment, we choose two representative datasets *Plasschaert* and *Adam*, and draw a chart depicting the cluster density and the OT plan on them in the alignment process, as shown in Fig. 3. At the begining, according to the OT plan, the units to be transported is dense and most parts in the OT plan are inter-cluster transport. This phenomenon is consistent with the fact that the cluster distribution densities of $\mathbb{P}$ and $\mathbb{Q}$ are not very close, especially in the Adam case. Meanwhile, the source distribution density $\mathbb{Q}$ is far away from the ground truth. At epoch 30, the units to be transported become relatively sparse, and there is not much cross-cluster transport, because most parts of the distributions $\mathbb{P}$ and $\mathbb{Q}$ have basically aligned to each other in a short period. During the training epoch 0 to epoch 30, most parts of the distribution $\mathbb{Q}$ are also aligned to the ground truth distribution. Subsequently, from epoch 30 to epoch 60 and epoch 90, the inter-cluster transport further reduces compared to previous OT plans. Additionally, the parts that the distribution $\mathbb{Q}$ not overlapped with the ground truth slightly move to real distribution in this period. Eventually, the OT plan becomes inclusive of a significant amount of intra-cluster transport and a small amount of inter-cluster transport, where intra-cluster transport is treated as an confidence enhancement. Above observations support that the OT-based joint representation alignment is necessary, for it is able to optimize the cluster distribution to be approximately aligned with the real distribution. Overall, the OT-based joint representation alignment is reasonable and effective.

Conclusion

In this paper, we propose an unsupervised gene-cell collective representation learning and OT approach, named scGCOT. scGCOT incorporates attention-based graph autoencoders to learn the cell correlations and gene correlations, then performs the count matrix reconstruction and ZINB model parameter estimation. Moreover, the OT theory is devised for the joint representation alignment in clustering. scGCOT can effectively capture the properties of scRNA-seq data for accurate cell type identification, so as to facilitate other bioinformatics downstream tasks, e.g., predicting the differentiation trajectories between cell types, gene regulatory network inference, and pathology analysis. On the other hand, the introduction of OT with earth-mover distance not only addresses the drawbacks of KL-divergence (e.g., zero truncation) but also improves efficiency. Experimental results on 15 real scRNA-seq datasets indicate the competitive edges of scGCOT.

Acknowledgements

This research was substantially sponsored by the research projects (Grant No. 32170654 and Grant No. 32000464) supported by the National Natural Science Foundation of China and was substantially supported by the Shenzhen Research Institute, City University of Hong Kong. The work described in this paper was substantially supported by the grant from the Research Grants Council of the Hong Kong Special Administrative Region [CityU 11203723]. This project was substantially funded by the Strategic Interdisciplinary Research Grant of City University of Hong Kong (Project No. 2021SIRG036). The work described in this paper was partially supported by the grant from City University of Hong Kong (CityU 9667265). The author Ming Gao also thanks the funding of Dalian Scientific and Technological Talents Innovation Support Plan (2022RG17) and Basic Scientific Research Project of Liaoning Provincial Department of Education (JYTZD2023050).

References

Adam, M.; Potter, A. S.; and Potter, S. S. 2017. Psychrophilic proteases dramatically reduce single-cell RNA-seq artifacts: a molecular atlas of kidney development. *Development*, 144(19): 3625–3632.

Butler, A.; Hoffman, P.; Smibert, P.; Papalexi, E.; and Satija, R. 2018. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5): 411–420.

Chen, L.; Wang, W.; Zhai, Y.; and Deng, M. 2020. Deep soft K-means clustering with self-training for single-cell RNA sequence data. *NAR genomics and bioinformatics*, 2(2): lqaa039.

Chen, R.; Wu, X.; Jiang, L.; and Zhang, Y. 2017. Single-cell RNA-seq reveals hypothalamic cell diversity. *Cell reports*, 18(13): 3227–3241.

Cheng, Y.; and Ma, X. 2022. scGAC: a graph attentional architecture for clustering single-cell RNA-seq data. *Bioinformatics*, 38(8): 2187–2193.

Dong, M.; Thennavan, A.; Urrutia, E.; Li, Y.; Perou, C. M.; Zou, F.; and Jiang, Y. 2021. SCDC: bulk gene expression deconvolution by multiple single-cell RNA sequencing references. *Briefings in bioinformatics*, 22(1): 416–427.

Du, J.; Zhang, S.; Wu, G.; Moura, J. M.; and Kar, S. 2017. Topology adaptive graph convolutional networks. *arXiv preprint arXiv:1710.10370*.

Eraslan, G.; Simon, L. M.; Mircea, M.; Mueller, N. S.; and Theis, F. J. 2019. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications*, 10(1): 390.

Gan, Y.; Huang, X.; Zou, G.; Zhou, S.; and Guan, J. 2022. Deep structural clustering for single-cell RNA-seq data jointly through autoencoder and graph neural network. *Briefings in Bioinformatics*, 23(2): bbac018.

Genevay, A.; Peyré, G.; and Cuturi, M. 2018. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, 1608–1617. PMLR.

Grün, D.; Kester, L.; and Van Oudenaarden, A. 2014. Validation of noise models for single-cell transcriptomics. *Nature methods*, 11(6): 637–640.

Guo, F.; Yan, L.; Guo, H.; Li, L.; Hu, B.; Zhao, Y.; Yong, J.; Hu, Y.; Wang, X.; Wei, Y.; et al. 2015. The transcriptome and DNA methylome landscapes of human primordial germ cells. *Cell*, 161(6): 1437–1452.

Hu, D.; Liang, K.; Zhou, S.; Tu, W.; Liu, M.; and Liu, X. 2023. scDFC: A deep fusion clustering method for single-cell RNA-seq data. *Briefings in Bioinformatics*, bbad216.

Kendall, A.; Gal, Y.; and Cipolla, R. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7482–7491.

Kiselev, V. Y.; Andrews, T. S.; and Hemberg, M. 2019. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics*, 20(5): 273–282.

Kulkarni, A.; Anderson, A. G.; Merullo, D. P.; and Konopka, G. 2019. Beyond bulk: a review of single cell transcriptomics methodologies and applications. *Current opinion in biotechnology*, 58: 129–136.

Li, H.; and Wang, J. 2023. CAPKM++ 2.0: An upgraded version of the collaborative annealing power k-means++ clustering algorithm. *Knowledge-Based Systems*, 262: 110241.

Li, X.; Wang, K.; Lyu, Y.; Pan, H.; Zhang, J.; Stambolian, D.; Susztak, K.; Reilly, M. P.; Hu, G.; and Li, M. 2020. Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. *Nature communications*, 11(1): 2338.

Lopez, R.; Regier, J.; Cole, M. B.; Jordan, M. I.; and Yosef, N. 2018. Deep generative modeling for single-cell transcriptomics. *Nature methods*, 15(12): 1053–1058.

Luo, Z.; Xu, C.; Zhang, Z.; and Jin, W. 2021. A topology-preserving dimensionality reduction method for single-cell RNA-seq data using graph autoencoder. *Scientific reports*, 11(1): 20028.

Muraro, M. J.; Dharmadhikari, G.; Grün, D.; Groen, N.; Dielen, T.; Jansen, E.; Van Gurp, L.; Engelse, M. A.; Carlotti, F.; De Koning, E. J.; et al. 2016. A single-cell transcriptome atlas of the human pancreas. *Cell systems*, 3(4): 385–394.

Petropoulos, S.; Edsgärd, D.; Reinius, B.; Deng, Q.; Panula, S. P.; Codeluppi, S.; Reyes, A. P.; Linnarsson, S.; Sandberg, R.; and Lanner, F. 2016. Single-cell RNA-seq reveals lineage and X chromosome dynamics in human preimplantation embryos. *Cell*, 165(4): 1012–1026.

Plasschaert, L. W.; Žilionis, R.; Choo-Wing, R.; Savova, V.; Knehr, J.; Roma, G.; Klein, A. M.; and Jaffe, A. B. 2018. A single-cell atlas of the airway epithelium reveals the CFTR-rich pulmonary ionocyte. *Nature*, 560(7718): 377–381.

Pollen, A. A.; Nowakowski, T. J.; Shuga, J.; Wang, X.; Leyrat, A. A.; Lui, J. H.; Li, N.; Szpankowski, L.; Fowler, B.; Chen, P.; et al. 2014. Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature biotechnology*, 32(10): 1053–1058.

Schaum, N.; Karkanias, J.; Neff, N. F.; May, A. P.; Quake, S. R.; Wyss-Coray, T.; Darmanis, S.; Batson, J.; Botvinnik, O.; Chen, M. B.; et al. 2018. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris: The Tabula Muris Consortium. *Nature*, 562(7727): 367.

Shi, Y.; Huang, Z.; Feng, S.; Zhong, H.; Wang, W.; and Sun, Y. 2020. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*.

Simmons, S. K.; Lithwick-Yanai, G.; Adiconis, X.; Oberstrass, F.; Iremadze, N.; Geiger-Schuller, K.; Thakore, P. I.; Frangieh, C. J.; Barad, O.; Almogy, G.; et al. 2023. Mostly natural sequencing-by-synthesis for scRNA-seq using Ultima sequencing. *Nature Biotechnology*, 41(2): 204–211.

Tian, T.; Wan, J.; Song, Q.; and Wei, Z. 2019. Clustering single-cell RNA-seq data with a model-based deep learning approach. *Nature Machine Intelligence*, 1(4): 191–198.

Tran, D.; Nguyen, H.; Tran, B.; La Vecchia, C.; Luu, H. N.; and Nguyen, T. 2021. Fast and precise single-cell data analysis using a hierarchical autoencoder. *Nature communications*, 12(1): 1029.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wan, H.; Chen, L.; and Deng, M. 2022. scNAME: neighborhood contrastive clustering with ancillary mask estimation for scRNA-seq data. *Bioinformatics*, 38(6): 1575–1583.

Wang, B.; Ramazzotti, D.; De Sano, L.; Zhu, J.; Pierson, E.; and Batzoglou, S. 2018. SIMLR: A tool for large-scale genomic analyses by multi-kernel learning. *Proteomics*, 18(2): 1700232.

Wang, D.; and Gu, J. 2018. VASC: dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder. *Genomics, proteomics & bioinformatics*, 16(5): 320–331.

Wang, J.; Ma, A.; Chang, Y.; Gong, J.; Jiang, Y.; Qi, R.; Wang, C.; Fu, H.; Ma, Q.; and Xu, D. 2021. scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nature communications*, 12(1): 1882.

Wang, J.; Xia, J.; Wang, H.; Su, Y.; and Zheng, C.-H. 2023a. scDCCA: deep contrastive clustering for single-cell RNA-seq data based on auto-encoder network. *Briefings in Bioinformatics*, 24(1): bbac625.

Wang, Y.; Yu, Z.; Li, S.; Bian, C.; Liang, Y.; Wong, K.-C.; and Li, X. 2023b. scBGEDA: deep single-cell clustering analysis via a dual denoising autoencoder with bipartite graph ensemble clustering. *Bioinformatics*, 39(2): btad075.

Wolf, F. A.; Angerer, P.; and Theis, F. J. 2018. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology*, 19: 1–5.

Xie, J.; Girshick, R.; and Farhadi, A. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, 478–487. PMLR.

Xin, Y.; Kim, J.; Okamoto, H.; Ni, M.; Wei, Y.; Adler, C.; Murphy, A. J.; Yancopoulos, G. D.; Lin, C.; and Gromada, J. 2016. RNA sequencing of single human islet cells reveals type 2 diabetes genes. *Cell metabolism*, 24(4): 608–615.

Young, A. L.; Marinescu, R. V.; Oxtoby, N. P.; Bocchetta, M.; Yong, K.; Firth, N. C.; Cash, D. M.; Thomas, D. L.; Dick, K. M.; Cardoso, J.; et al. 2018. Uncovering the heterogeneity and temporal complexity of neurodegenerative diseases with Subtype and Stage Inference. *Nature communications*, 9(1): 4273.

Yu, Z.; Lu, Y.; Wang, Y.; Tang, F.; Wong, K.-C.; and Li, X. 2022. Zinb-based graph embedding autoencoder for single-cell rna-seq interpretations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 4671–4679.

Zeng, Y.; Zhou, X.; Rao, J.; Lu, Y.; and Yang, Y. 2020. Accurately clustering single-cell RNA-seq data by capturing structural relations between cells through graph convolutional network. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 519–522. IEEE.

Žurauskienė, J.; and Yau, C. 2016. pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC bioinformatics*, 17: 1–11.