Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT
MODEL-BASED DEEP SIAMESE AUTOENCODER FOR CLUSTERING SINGLE CELL RNA-SEQ DATA

by
Zixia Meng

In the biological field, the smallest unit of organisms in most biological systems is the single cell, and the classification of cells is an everlasting problem. A central task for analysis of single-cell RNA-seq data is to identify and characterize novel cell types. Currently, there are several classical methods, such as K-means algorithm, spectral clustering, and Gaussian Mixture Models (GMMs), which are widely used to cluster the cells. Furthermore, typical dimensional reduction methods such as PCA, t-SNE, and ZIDA have been introduced to overcome "the curse of dimensionality". A more recent method scDeepCluster has demonstrated improved and promising performances in clustering single-cell data. In this study, a clustering method is proposed to optimize scDeepCluster with Siamese networks, which will learn more reliable functions for mapping inputs to the latent space. Also, the spectral clustering based on the SpectralNet algorithm is employed to improve clustering performances. Extensive experiments are conducted to demonstrate its superior performance in comparison with the current state-of-art methods.

MODEL-BASED DEPP SIAMESE AUTOENCODER FOR CLUSTERING SINGLE CELL RNA-SEQ DATA

by
Zixia Meng

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

Department of Computer Science

May 2020

APPROVAL PAGE

MODEL-BASED DEPP SIAMESE AUTOENCODER FOR CLUSTERING SINGLE CELL RNA-SEQ DATA

Zixia Meng

Dr. Wei Zhi, Dissertation Advisor Professor of Computer Science, NJIT Associate Chair for Graduate Studies of Computer Science, NJIT	Date
Dr. Usman Roshan, Committee Member Associate Professor of Computer Science, NJIT	Date
Dr. Wenge Guo, Committee Member Associate Professor of Mathematical Sciences, NJIT	Date
Dr. Nan Gao, Committee Member Associate Professor of Biological Sciences, Rutgers	Date

BIOGRAPHICAL SKETCH

Author: Zixia Meng

Degree: Master of Science in Computer Science

Date: May 2020

Undergraduate and Graduate Education:

- Master of Science in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2020

- Bachelor of Science in Mechanical Engineering,
  Xi'an Jiaotong University, Xi'an, Shaanxi, 2018

Major: Computer Science

ACKNOWLEDGEMENTS

These years as a master's student at NJIT are a splendid and fast-paced experience of my life. Deep appreciations fill my mind. I would like to mention those people who offered help, support, and unforgettable experiences for me through these years.

First and foremost, I would like to express my special gratitude to Dr. Zhi Wei, my research advisor and mentor, for his invaluable guidance and support throughout my graduate journey. I appreciate Dr. Wei, who introduced me to a great research area that I feel fortunate to have learned about. He always encourages me to join different professional conferences and attend student presentations or poster competitions, apply to scholarships, and mentor other students. He has been an amazing role model for me during my graduate work and constantly motivates me to push myself more. His mentorship positively influences me and helps me build confidence for my current research and future career. Thank you for the tireless encouragement, unparalleled opportunities, and for pushing me forward throughout my time here which has helped me grow as a researcher.

Moreover, I would like to acknowledge my friends, Jiaheng Zhang, Zhonghao Li, Shuyue Yang, Peiran Jia, Qi Meng and Leqi Lin, who always encourage and accompany me when I feel stressed. They are my family in the United States, and will be my friends forever, even if we may separate in different places someday.

Last, but not least, I want to thank my mother and my father for their unconditional love, patience, and support. Without their understanding and encouragement, I would not

have this chance to experience this two-year wonderful journal. I am forever grateful for all they have done for me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

In order to realize the functional consequences of a DNA sequence, we have to study its product which is RNA and proteins. Normally, the site of RNA or proteins are in the cell which is the smallest unit of organisms in any biological system. There are numerous types of functional RNA, and some of them play an active role within cells by catalyzing biological reactions, controlling gene expression, or sensing and communicating responses to cellular signals.[1] Moreover, messenger RNA (mRNA) in which its function is to transport the DNA code (genetic information) into the cytoplasm where it can be translated into proteins. It was a quite hard work to quantify and analyze hundreds of RNA in the sample. However, RNA sequencing (RNA-seq) as a particular technology-based sequencing technique can reveal the identities of most RNA species inside a cell by using a variety of next-generation sequencing (NGS).[2] In 2009, Tang F. et al.[3] first proposed a single-cell genes expression profiling assay, also known as mRNA-seq, which then RNA-seq has become one of the most promising technology to study complex biological questions. Standard methods such as microarrays and bulk RNA-seq utilize large populations of cells to analyze the expression of the RNAs. However, there are several limitations of microarrays and bulk RNA-seq. In microarrays, designed arrays often have

multiple related DNA/RNA sequences that bind to the same probe.[4] That is, if we want to detect “gene A”, we may also detect “gene B” and “gene C”. Besides, the sequences can only be detected when the array is designed to detect, which means that genes that have not yet been annotated in a genome will not be represented. In bulk RNA-seq, the data formally represents an average of gene expression patterns, which may miss biological information differentiating between cells. Although individual cells are estimated to contain a huge number of molecules, the high variability of relative proportions of different transcript classes in a population should not be ignored.[5] Thus, single cell analysis is needed. Single cell analysis includes the study of genomics, transcriptomics, proteomics and metabolomics with several purposes such as tracking the changes that occurs in populations, determining gene expression in each cell and understanding the activity of certain cells. However, this technology is particularly prone to dropout events due to the relatively shallow sequencing depth per cell.[6] This makes clustering analysis on scRNA-seq a particularly challenging task.

Normally, classical clustering methods including K-means algorithm, spectral clustering and Gaussian Mixture Models (GMMs) are commonly used. Very recent state-of-the-art methods for scRNA-seq have been proposed. In 2015, Xu and Su described SNN-Cliq, a quasi-clique-base algorithm combined with concepts of shared nearest-neighbor similarity measurement, which worked well especially in clustering high-dimensional scRNA-seq datasets.[7] In 2018, Sinha D. et al.[8] presented dropClust as a new

clustering strategy, looking for nearest neighbors using Locality Sensitive Hashing (LSH). Although these methods have shown very decent performances, there are still some limitations. Some of them relied on the full graph Laplacian matrix, which usually had quadratic or super-quadratic complexities to compute and store.[9] Decomposition of the Laplacian matrix may require cubic complexities.[9] In 2019, Tian Tian et al.[10] proposed scDeepCluster – a model-based deep learning approach for clustering scRNA-seq data to solve mentioned issues. Although scDeepCluster used the model appropriate for characterizing data with excessive zeros, the latent space generated from the model was lack of biological interpretation. Based on the research study of Tian Tian et al., this study furthers to present a new combined model which could represent biological meanings and suggest a corresponding clustering algorithm that has better potential and performance than scDeepCluster.

1.2 Standard Sequencing Methods

1.2.1 Microarrays

Due to the rapid development and implementation of genomic microarray technologies, a large number of microarray data has been analyzed which has been proved and widely used in several fields such as cancer diagnosis[11, 12] , prediction[13] and prevention [14, 15] based on the assessment of mRNA transcript levels on a genome-wide scale.[11, 16] The mRNA is an intermediary molecule which carries the genetic information from the cell nucleus to the

cytoplasm for protein synthesis.[17] These mRNAs synthesize the corresponding protein by translation in which we can assess the genetic information or the gene expression indirectly by assessing the various mRNAs.[17] Microarrays are based on nucleic acid hybridization principle where arrays are comprised of a collection of DNA probes that are spotted on a solid support ideally in a glass or silicon platform. They are used to detect the presence of gene transcripts.[18] Standardized microarray dataset consists of thousands of gene expression and a few hundred of samples. In this technique, genomic DNA is fluorescently labeled and used to determine the presence of gene loss or amplification.[12, 14, 19] Typical experimental steps for microarray is shown in **Figure 1.1**. Each expression measures the level of activity of genes within a given tissue. The small variations in the DNA sequence that lead to different characteristics (such as skin color, facial features, or height) are known as polymorphisms, and also can contribute to the development of many syndromes and diseases.[20] By comparing the genes expressed in abnormal cancerous tissues with those in normal tissues, we may get a good insight into the disease pathology which allows better diagnosis and predictions for future samples.[13] These genetic variations can be easily identified by the microarray technique. However, this technique has several limitations due to its cost and access problems of the sample. Besides, it is still not easy to analyze the huge amount of data generated by this technology.[21] Moreover, each microarray can only provide information about the genes that are included in the array.[22]

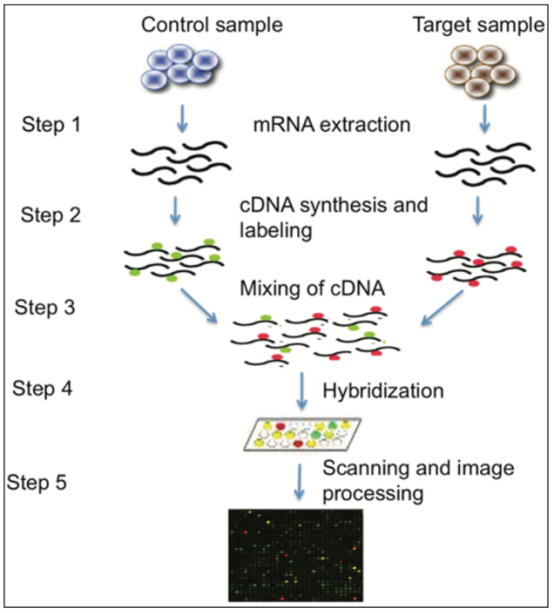


Figure 1.1 Typical steps for microarray experiment.

Source:[23]

1.2.2 Bulk Sequencing

Bulk RNA sequencing (RNA-seq) is a technique which has been widely used to analyze entire genomes by its gene expression patterns at population level in the past decade.[24] In bulk sequencing, the data points observed (observations) are not single cells, but rather represent bulk samples (many cells). This tends to reduce the sparsity of values within the expression matrix which makes the parameters richer and less susceptible to dropouts. Bulk RNA-seq mainly reflects the averaged gene expression from an assembly of cells[25], which is the sum of cell type-specific gene expression weighted by cell type proportions.[26] This bulk RNA-seq data provides reliable measurements of gene expression levels throughout the genome for bulk samples. With sufficient sequencing depth, even weakly expressed transcripts can be accurately captured by RNA-seq data. Fromer et al. (2016)[27] used bulk

cell datasets, obtained from the prefrontal cortex of post-mortem subjects, to gain insights into how genetic risk variation for schizophrenia affects gene expression and likely generates risk for this severe psychiatric disorder. A number of approaches have been developed for between-sample normalization of bulk RNA-seq data, such as DESeq2[28]and trimmed mean of M values (TMM).[29] However, in complex tissues with multiple heterogeneous cell types, bulk RNA-seq require a priori knowledge, either of gene expression profiles of purified cell types [30-32] or of cell-type compositions.[33] Recent advances in single-cell RNA-seq enable characterization of transcriptomic profiles with single-cell resolution and circumvent averaging artifacts associated with traditional bulk RNA-seq data.[30] In **Figure 1.2**, we perform a schematic of bulk RNA-seq and the differences between Single cell RNA-seq.

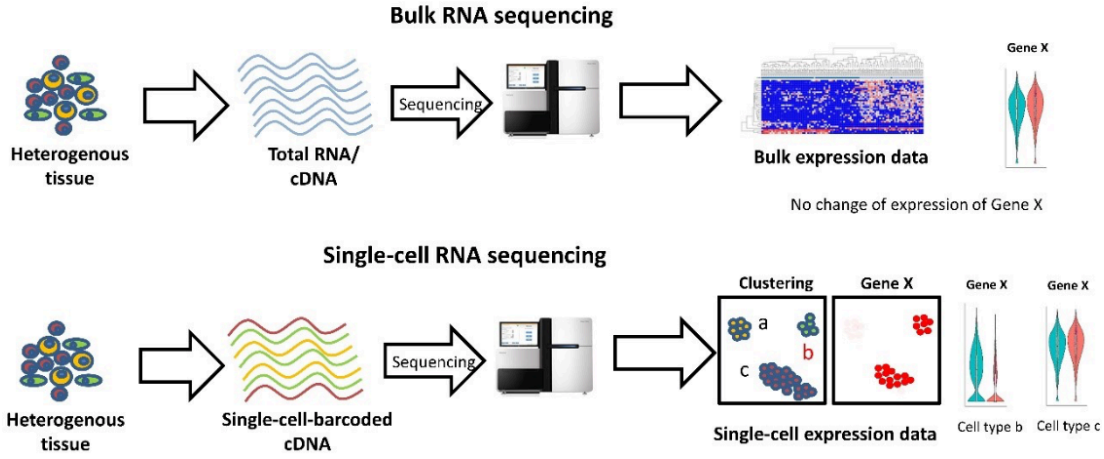


Figure 1.2 Schematic of bulk RNA-seq and single-cell RNA-seq.

Source:[34]

1.2.3 Single-cell Sequencing

The scRNA-seq is a method that examines the sequence information from individual cells equipped with optimized next-generation sequencing (NGS) technologies. It helps researchers comprehend different functions between individual cells or cell types under their microenvironments.[35] There are six main steps in the general procedure of single cell sequencing: isolation of single cells; cell lysis to obtain DNA or RNA; addition of barcodes in single cells; amplification of DNA and RNA for sequencing; library preparation and sequencing; and data analysis[36], as shown in **Figure1.3**. Typical single cell isolation methods, such as fluorescence-activate cell sorting (FACS), laser capture microdissection (LCM), allow the precise isolation of selected single cells from complex samples. Microfluidics is a highly integrated system used widely, and it allows sequential processing of small volumes of fluids in hundreds of channels to achieve single cell sequencing.[37] There are several available microfluidics platforms, such as 10X Genomics Chromium and Drop-seq.

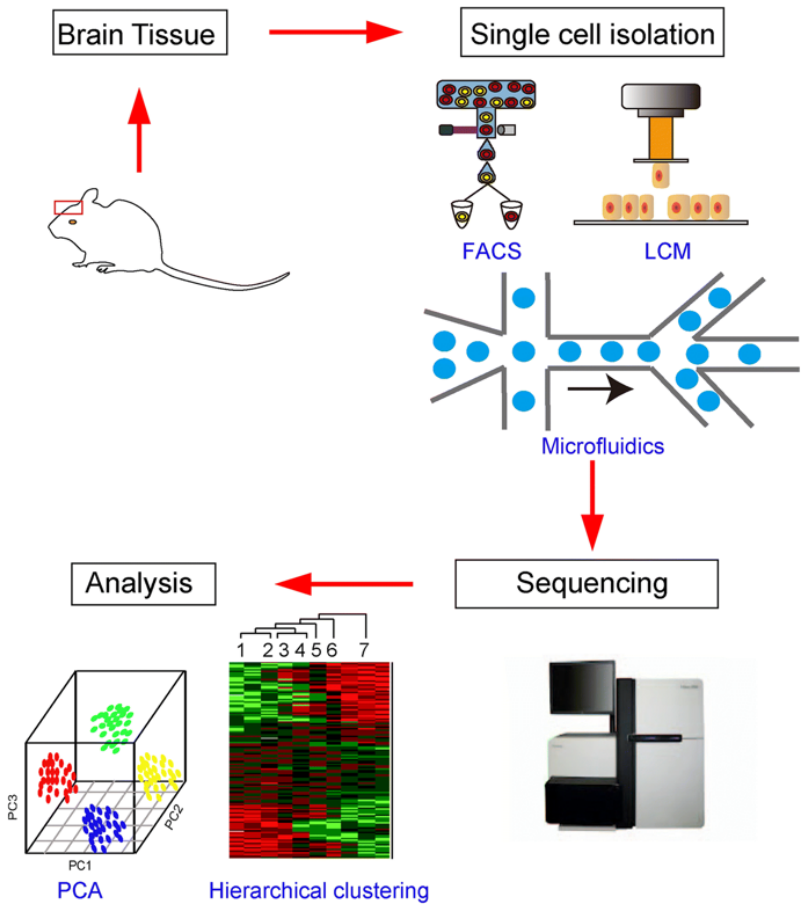


Figure 1.3 Single-cell sequencing flow chart

Source:[36]

Current scRNA-seq protocols involve isolating single cells and their RNA, reverse transcription (RT), amplification, library generation and sequencing. There are several challenges when using scRNS-seq, including the difficulty of identifying rare transcripts[38], and improving the efficiency of the RT reaction, which determines the amount of the cell's RNA population will be eventually analyzed by the sequencer. Due to the small available amount of material, scRNA-seq is infeasible to obtain complete information of expression files. Thus, gene clustering methods are proposed to identify patterns of gene expression. The main goal of clustering is to find a way to determine the identity of cells that do not

have known genetic markers.

The increasing size of high-dimensional scRNA-seq datasets also enhances statistic challenges due to the "curse of dimensionality". Typical dimensional reduction methods, such as PCA and t-SNE, are introduced to solve this kind of problem. However, there are huge differences between normal high-dimensional data and scRNA-seq data. Particularly, the most frequent expression level in scRNA-seq data is zero (typically > 50%). These 'false' zero count observations are caused by so-called dropout events, that could be either biological characteristics in which the genes may fail to express at the time of measurement, or technology limitation in which the sequencing tool does not detect a certain level of expression. In addition, high variation is another feature of scRNA-seq count data, even among cells from the same type.

1.3 Dimensional Reduction and Clustering

1.3.1 K-means

K-means clustering is an iterative algorithm of vector quantization that tries to partition n observations into k pre-defined non-overlapped clusters $S = \{s_1, s_2, \dots, s_k\}$. It assigns data points to the cluster with the nearest mean (formally called as cluster center or cluster centroid). The optimization process is to make the inter-cluster points as similar (close) as possible while keeping the clusters as different (far) as possible. Formally, the objection is:

$$\underset{y}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in s_i} \|x - \mu_i\|^2$$

where μ_i is the mean of cluster s_i.

In a recent work, Zheng and colleagues[39] used K-means as the method for clustering droplet-seq data. This method struggled to identify clusters of non-spherical shapes. The original K-means, which uses squared Euclidean distances, tends to cluster equal-group-size sphere-like data. In order to perform K-means on different types of data, several methods of adapting distance function were proposed. DropClust[8] used Locality Sensitive Hashing (LSH) to find nearest neighbors of individual transcriptomes. DendropSplit was an end-to-end framework for clustering scRNA-seq data with interpretable hyperparameters.[40] It performed Pearson correlation distance between cells and separation scores between clusters which would lead to biologically meaningful hierarchical clustering dendrograms. However, an exhaustive nearest neighbor search requires quadratic time computing pair-wise distances. For large sample sizes, this approach turns out to be significantly slow. Thus, methods of dimensional reduction are needed to be performed before the clustering process.

1.3.2 Modern Methods of Dimensional Reduction

1.3.2.1 Principle Component Analysis

scRNA-seq measurements are commonly affected by high levels of technical noise, posing challenges for data analysis and visualization.[41] Unsupervised learning techniques have been increasingly popular and useful for exploring and analyzing scRNA-seq data.

Particularly, principal component analysis (PCA) is one of the most frequently used method by reducing the dimensionality of the data while retaining most of the variation in the data set through the mathematical algorithm.[42] Furthermore, closely related to factor analysis and latent variable models, principal components (PCs) help us to identify hidden and unmeasured structures that arise from biological and technical sources of variation.[43, 44] However, PCA would become inefficient when an increasing size and sparsity of genomic data happens. Furthermore, the outcome of PCA may be easily biased by outline observations, which is not an expected behavior.[45] Thus, several attempts of PCA or PCA based algorithm were developed and adapted on the scRNA-seq to achieve high effectiveness and denoising accuracy. In 2017, Lin P. et al. proposed a PCA-like algorithm, CIDR[46], with the ability of ultrafast speed when handling rapid-growing datasets. In this algorithm, the inflation of the distance matrix caused by the dropout event was takem into consideration. Besides, Lin P. et al. also imputed the value of dropout candidates based on the given probability distribution to shrink the inflation. Y. h. Taguchi applied PCA-based unsupervised FE[47] to gene expression profiles retrieved by scRNA-seq analysis. The evaluation results showed that the proposed method had identified more genes associated with significant biological terms enrichment than the conventional approaches.

1.3.2.2 t-Stochastic Neighbor Embedding (t-SNE) Visualization:

Another extensively used tool of dimensional reduction is called t-Stochastic Neighbor

Embedding (t-SNE)[48]. Compared to PCA, it handles non-linear data efficiently, for it constructs probability distribution in high dimensions, which means that similar points have a high probability of being picked, and then it defines the similar distribution in low-dimensional space. This method is usually used as visualization, and has been applied to several fields, such as cell segmentation and tissue image processing[49], human genetic association studies[50] and so on. The limitation of t-SNE is the problem of computational complexity, for it computes pairwise conditional probabilities for each data point. Gisbrecht, A. et al.[51] tested the ability of their model, kernel t-SNE, in comparison to standard t-SNE for several datasets (shown as **Figure 1.4**). Though they showed that the model could be solved in linear time, the improvement of clustering was limited. Yu, M. et al.[52] extended t-SNE by a deep feed-forward network for target recognition, but their model was pre-trained using Restricted Boltzmann Machines (RBMs).

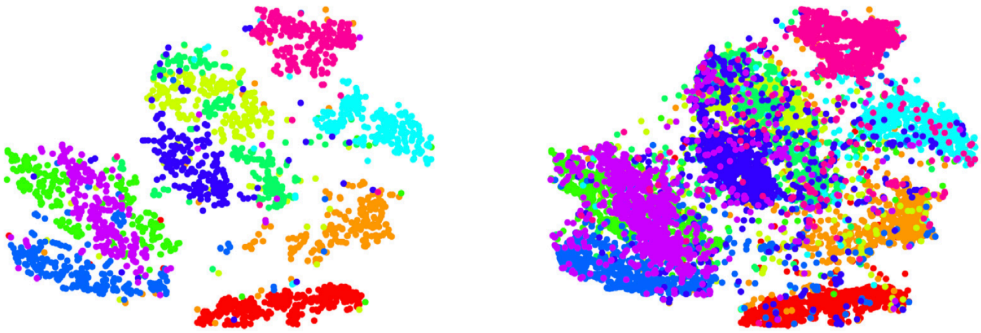


Figure 1.4 Comparison of t-SNE and kernel t-SNE applied to the dataset MNIST.

Source [51]

1.3.2.3 Unsupervised Deep Neural Network Model: Autoencoder

Autoencoder is an unsupervised deep neural network (DNN) model which allows reducing the dimensionality of data[53], as shown in **Figure 1.5**. Zhou C. and Paffenroth R. C.[54] developed "Robust Deep Autoencoder" (RDA) to deal with the outliers and noise. They split data into two parts, which one of them could be successfully performed by a regular deep autoencoder and the other contained the noise and outliers. Peng J. et al. [55] put prior biological knowledge and an autoencoder together to build a model named Gene Ontology AntoEncoder (GOAE). Instead of using the conventional mean square loss as the loss function in a regular autoencoder, a deep count autoencoder (DCA) chose to use a zero-inflated negative binomial (ZINB) as the loss function.[56] In their experiments, DCA showed it worked well in some kinds of downstream analyses. In the previous work, the ZINB model had been applied to microbiome sequencing data and was proved effective on characterizing discrete, over-dispersed and zero-inflated count data.[57]

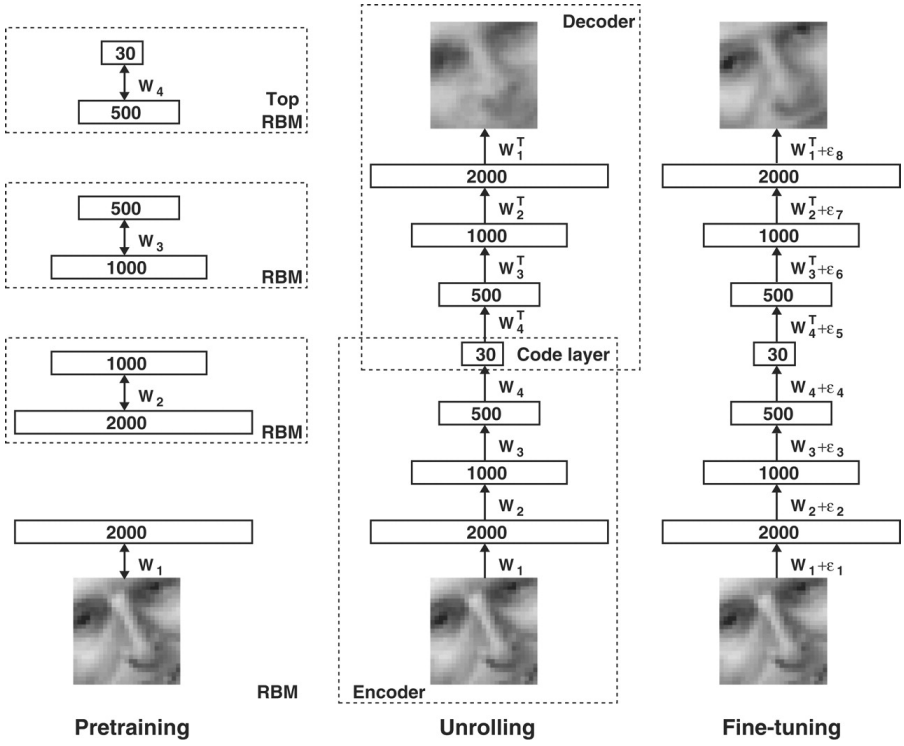


Figure 1.5 An autoencoder with pretraining consists of learning a stack of restricted Boltzmann machines (RBMs).

Source: [53]

1.3.2.4 Spectral Clustering

In recent decades, spectral clustering has become one of the most popular methods. Spectral clustering techniques use the spectrum (eigenvalues) of the similarity matrix of the data to perform dimensionality reduction before clustering in lower dimensions. Spectral clustering has many advantages; it is simple to implement and can be solved by standard linear algebra methods. Several methods have been developed to apply spectral clustering to large datasets. Kadim Taşdemir[58] proposed a method of vector quantization to speed up spectral clustering by reducing the computation of the decomposition. Cao, J. et

al.[59] suggested an improved spectral clustering method only based on local information. That is, only the affinity graph with local relations was needed in order to accelerate the algorithm. Shaham, U. et al.[60] proposed a network, called SpectralNet, which used a procedure that involved constrained stochastic optimization. The example results are shown in **Figure 1.6**. They replaced the standard affinities with affinities learned from a Siamese network. Their results also showed that applying SpectralNet to transformed data obtained by an autoencoder allowed further improvement.

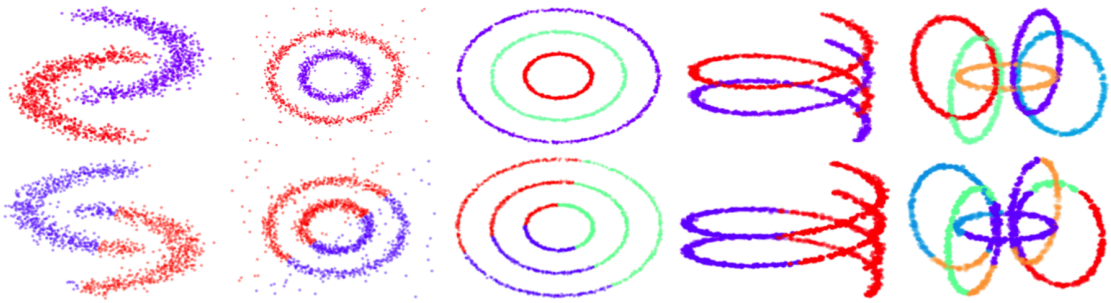


Figure 1.6 Illustrative 2D and 3D examples showing the results of SpectralNet clustering (top) compared to typical results (bottom).

Source: [60]

1.3.3 Specific Imputation Methods for Single-cell Sequencing

Various statistic methods are proposed to address the special characteristic of the scRNA-seq data, for common features, such as mean, median and standard deviation, fail to depict. Scher, J.U. et al.[61] developed a statistical test based on zero-inflated Gaussian model, with regard to addressing the feature of zero-inflated count data with variable library size. Normalizing the count data or rarefying the data into equal library sizes was commonly

used to deal with the problem. McMurdie, P.J., and Holmes, S.[62] suggested to the direct application of the negative-binomial (NB) based methods for RNA-seq data, with support of DESeq[63] and edgeR[64]. Meanwhile, zero-inflated models were proved to have well controlled Type I errors, and were more accurate and efficient when estimating parameters, compared with other models.[65] For instance, these models assumed that the observed zero are consisted of 'structural zeros' (due to physical absence) and 'sampling zeros' (due to under-sampling), which were biological interpretable. Jun Chen et al.[57] used a zero-inflated negative binomial (ZINB) model on analysis of microbiome sequencing data. Their omnibus test suggested that allowing covariate-dependent dispersion could improve robustness of discrete, over-dispersed and zero-inflated count data. An example of differential dispersion is shown in **Figure 1.7**.

However, these imputation methods are not designed for clustering. Besides, scRNA-seq data has more dimensions than microbiome sequencing data, a new method, that could perform imputation clustering and dimensional reduction at the same time, is needed.

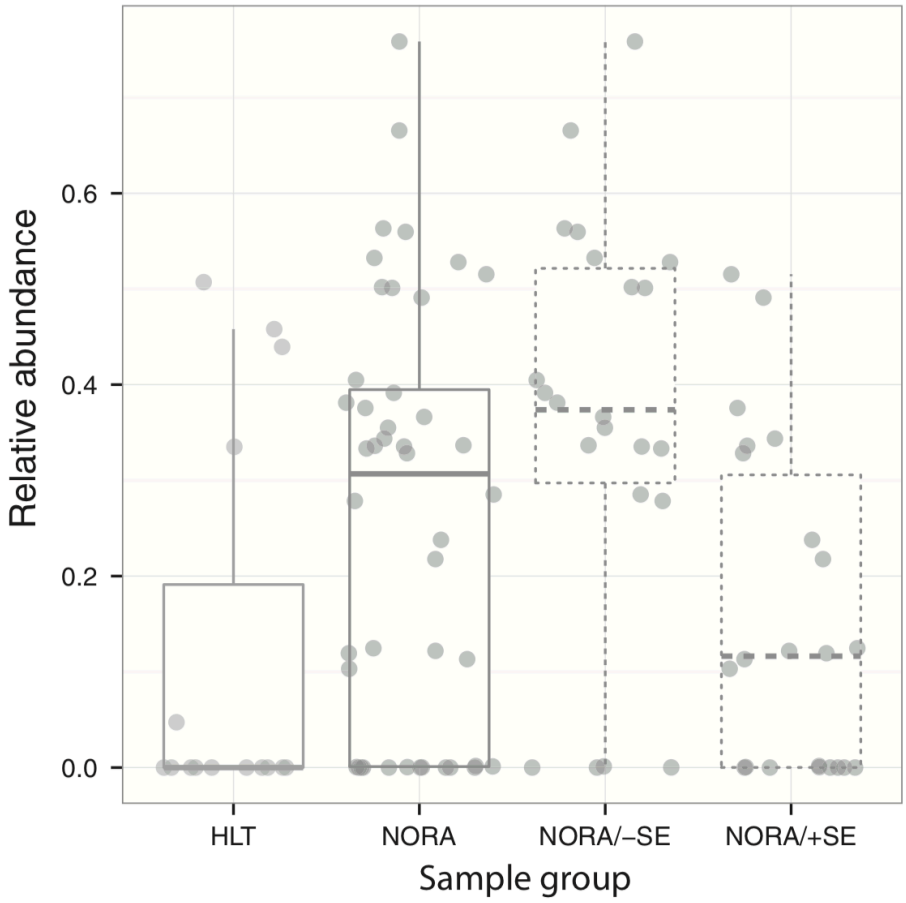


Figure 1.7 An example of differential dispersion.

Source:[66]

1.3.4 State-of-Art Clustering Approaches for scRNA-seq Data

Wang et al. proposed to apply multi-kernel learning (SIMLR) for single-cell interpretation[67, 68]. By combining multiple kernels, SIMLR allowed to learn distance metric best fit the structure of scRNA-seq data. It also addressed that even under an appropriate distance metric, the results could be poor because of the frequent dropout events. MPSSC was another novel spectral clustering, which imposed a specific sparse structure on target matrix via L1 penalty[69]. Although decent performances were achieved using these methods

(as shown in **Figure 1.8**), Tian and his colleagues' works[10] showed that these spectral clustering-based methods relied significantly on the full graph Laplacian matrix, which required expensive computation and storing space. For example, in MPSSC, a machine with 800Gb memory was necessary for clustering thousands of cells. In addition, spectral clustering fails to characterize the features of scRNA-seq data such as over-dispersion and zero inflation.

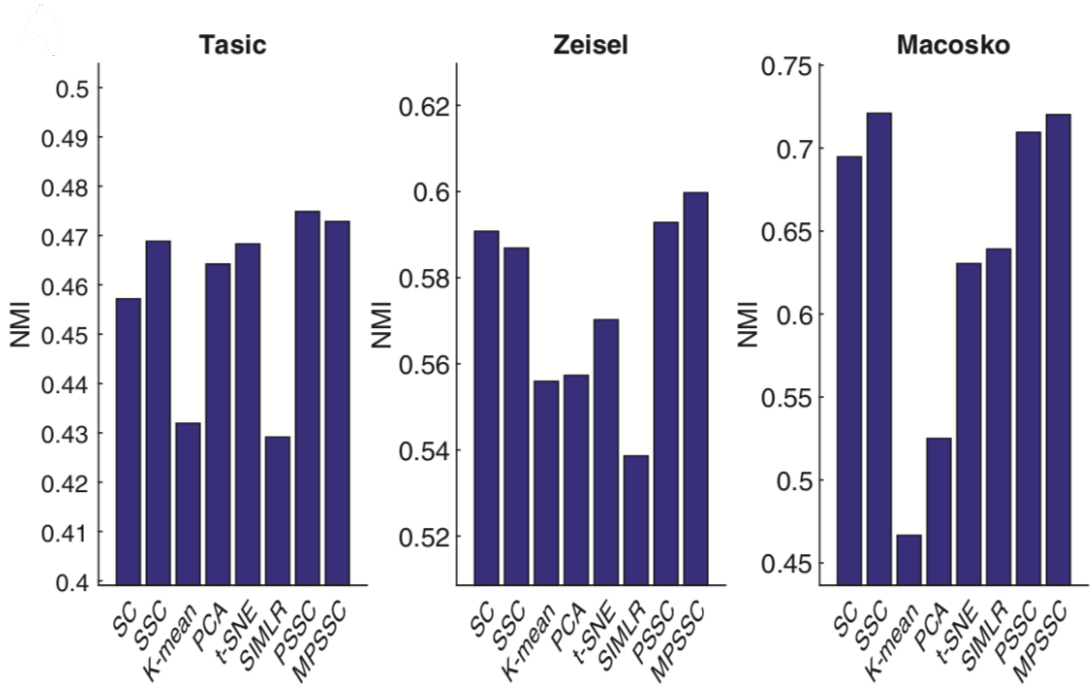


Figure 1.8 Evaluation of the eight clustering methods by NMI, implemented on the computing cluster (6 CPUs, 800 GB of memory).

Source:[69]

1.4 Research Objective

Tian Tian et al. proposed a model, named scDeepCluster[10] (single-cell model-based deeply embedded clustering), which introduced DNNs into the ZINB-based denoising[70, 71] autoencoders as well as the KL-divergence described in DEC algorithms.[72] The scDeepCluster model solved the main challenge in scRNA-seq data that ZINB model was not designed and optimized for clustering by integrating ZINB model with clustering loss in a principled way. However, this method could not maintain the distances between cells when mapping cells into latent features space. Siamese networks were first used to learn meaningful mappings when it was applied to face recognition problems.[73] Before training the network, positive pairs, both of which have the same label, and negative pairs, both of which do not have the same label, are generated. The distances are maintained by maximizing the distance between the ones in negative pairs and minimizing the distance between the ones in positive pairs. Therefore, we propose a method that connects the former model with Siamese networks to learn reliable mapping functions from inputs to the latent space. We then perform the spectral clustering based on SpectralNet algorithm to improve clustering performances.

CHAPTER 2

METHODS

2.1 Preparatory Work

2.1.1 Software and Tools

In this study, we implement the ideas in Python 3.0. The NumPy Python library is frequently used for scientific computing operations. However, in order to use the power of GPUs, we use PyTorch instead of NumPy for flexibility and speed. From Sickit-learn, we import KMeans and metrics for evaluation of the clustering process. Some tools used for, such as, generating pairs, normalization, and loss calculations are also implemented in Python. Most parts of our programs are running on the NVIDIA P100 GPUs, which are provided under the Extreme Science and Engineering Discovery Environment (XSEDE) digital service by the San Diego Supercomputer Center (SDSC).

2.1.2 Raw Count Data Pre-Processing

We use SCANPY[74] to deal with the biological data. SCANPY is a scalable toolkit for analyzing single-cell gene expression data. Its Python-based implementation efficiently deals with data sets of more than one million cells. SCANPY can perform essential pre-process such as normalization, which will reduce data redundancy and improve data integrity. It introduces a general class which could handle annotated data matrices, called

ANNDATA. **Figure 2.1** shows several functions that SCANPY is capable of, including t-SNE visualization.

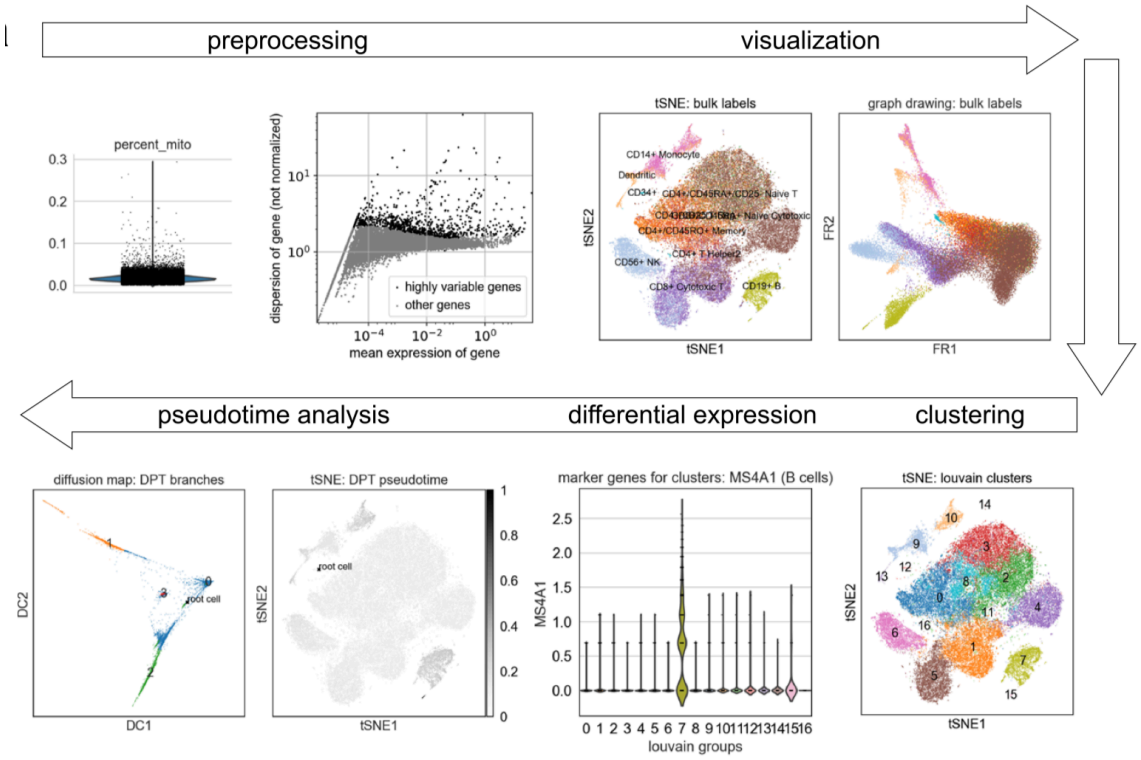


Figure 2.1 Processing that SCANPY is capable of, including regressing out confounding variables, normalization, and identification of highly variable genes, TSNE and graph-drawing.

Source: [60]

2.1.3 Real Data

In this study, we apply our model to four datasets that are described as followings. We got the 10X PBMC dataset (4K PBMCs from a healthy donor) from 10X scRNA-seq platform[75], which profiled the transcriptome of the peripheral blood mononuclear cells (PBMCs) from a healthy donor. PBMC 4k data were downloaded from the website of 10X

genomics (https://support.10xgenomics.com/single-cell-gene-expression/datasets/2.1.0/pbmc4k). Filtered gene/cell matrix and cell labels are identified by graph-based clustering (for the method description see https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/output/analysis).

The mouse ES cells dataset[76] was downloaded from GSE65525. Droplet-microfluidic was used for profiling transcriptomes. We downloaded the read count matrices of mouse ES cells sample 1, mouse ES cells LIF - 2 days, mouse ES cells LIF - 4 days and mouse ES cells LIF - 7 days, from Allon and colleagues' work[76], and put them together.

We got the mouse bladder cells dataset of the Mouse Cell Atlas project[77] from the authors (https://figshare.com/s/865e694ad06d5857db4b). The cells were sorted by tissues and the table of cell assignments and we downloaded the digital expression matrix, with the batch gene background removed, of all 400,000 single cells. From the raw count matrix, cells from bladder tissue were selected.

The worm neuron cells dataset was profiled by sci-RNA-seq (single-cell combinatorial indexing RNA sequencing)[78]. 50000 cells from nematode Caenorhabditis elegans at the L2 larval stage and were profiled and labeled (http://atlas .gs.washington.edu/worm-rna/docs/). Among them, a subset of neural cells was selected and labeled with "unclassified neurons" are removed. Thus, we had 4186 neural cells.

2.2 Networks of Dimensionality Reduction, Feature Selection and Spectral Clustering

2.2.1 ZINB Model-Based Autoencoder

The autoencoder is a kind of neural network model used to learn efficient coding in an unsupervised manner. As we presented before, the ZINB-based autoencoder used in this study is showed in **Figure 2.2**. Each autoencoder has two parts: the encoder and the decoder, as denoted in **Figure 2.2**. The aim of an autoencoder is to find a set of candidates which could stand for the dataset, and the encoder part will transform the input into hidden features.

Suppose that there is an autoencoder which has only one hidden layer. Let the x_i is one of the input vectors, $x_i \in \mathbb{R}^m$, and the hidden layer h has p units. Then the output of the hidden layer in the encoding process can be represented as following:

$$h_i = \sigma(Wx_i + b),$$

where $W = (w_{11}, \dots, w_{pm}) \in \mathbb{R}^{p \times m}$ is the weight matrix from the input data with m dimensions to each p units in the hidden layer; $b = (b_1, \dots, b_p)$ is a bias vector.

Then the output of the hidden layer in the decoding process can be represented as following:

$$x_i' = \sigma(W'h_i + b'),$$

where $W' = \left(w'_{11}, \dots, w'_{mp}\right) = (w_{11}^T, \dots, w_{pm}^T) \in \mathbb{R}^{m \times p}$.

Formally, we define the encoder function as $h = f(X)$ and the decoder function as

$X' = g(h)$, where $X = (x_1, \dots, x_m)^T$. Thus, the loss function of a regular autoencoder is defined as:

$$l(x_i, x_i') = \|x_i - x_i'\|^2$$

Unlike the regular autoencoder, the denoising ZINB model-based autoencoder used in this study is enhanced to implement functions of both imputation and denoising for sparse count data. The denoising autoencoder is an autoencoder with high robustness to partially destroyed inputs and is expected to predict original uncorrupted data as its output.[70] The empirical results show that an explicit denoising criterion does help the autoencoder to learn the structure of the input that are corrupted by small irrelevant noise in input. Therefore, the denoising autoencoder model is employed to map the input data from its original space to a low-dimensional embedded (latent) space as the clustering is processing. In the following experiments, random Gaussian noise is added into the input data and then the entire model is constructed with a normal fully connected layer. The corrupted input is noted by:

$$X^{corrupt} = X + e,$$

where e is the random Gaussian noise. Thus, the encoder and the decoder functions are defined as $h = f(X^{corrupt})$ and $X' = g(h)$, respectively. Both the encoder and the decoder are fully connected neural networks with rectifier activation function which is known as ReLUs.[79] The weights of the functions are learned by the training process of

which minimizing the loss function:

$$L(X, g(f(X^{corrupt}))).$$

Compared with the regular autoencoder, the major improvement of the ZINB model-based autoencoder is that the loss function of the later one is the likelihood of a ZINB distribution. The dropout events in scRNA-seq can be depicted by ZINB. Formally, the mean (μ), the dispersion (θ) of the negative binomial distribution and an additional coefficient (π) that represents the weight of the point mass of probability at zero (the probability of dropout events) are used to describe ZINB:

$$NB(X^{count}|\mu,\theta) = \frac{\Gamma(X^{count}+\theta)}{X^{count}!\Gamma(\theta)}\left(\frac{\theta}{\theta+\mu}\right)^{\theta}\left(\frac{\mu}{\theta+\mu}\right)^{X^{count}},$$

$$ZINB(X^{count}|\pi,\mu,\theta) = \pi\delta_0(X^{count}) + (1-\pi)NB(X^{count}|\mu,\theta),$$

where X^{count} represents the raw counts data. The parameters π, μ, θ will be estimated in the ZINB model-based autoencoder by appending three independent fully connected layer at the end of the decoder. Let $X' = g(f(X^{corrupt}))$ represents the last hidden layer of the decoder, then the functions of these three layers are denoted as:

$$M = diag(s_i) \times \exp(W_\mu X'),$$

$$\Theta = \exp(\mathrm{W}_\theta \mathrm{X}'),$$

$$\Pi = sigmoid(W_\pi X'),$$

where M, Θ, Π are the estimations of mean, dispersion and drop probability in matrix form, respectively. The size factors s_i in the first equation are considered as independent input and are calculated before the training process. Note that the activation function for the mean and dispersion layer is exponential because all parameters are non-negative values, and the activation function for the additional coefficient is sigmoid because the dropout probability lies between 0 and 1. Thus, the loss function is described as following:

$$L_{ZINB} = -\log\left(ZINB(X^{count}|\pi, \mu, \theta)\right)$$

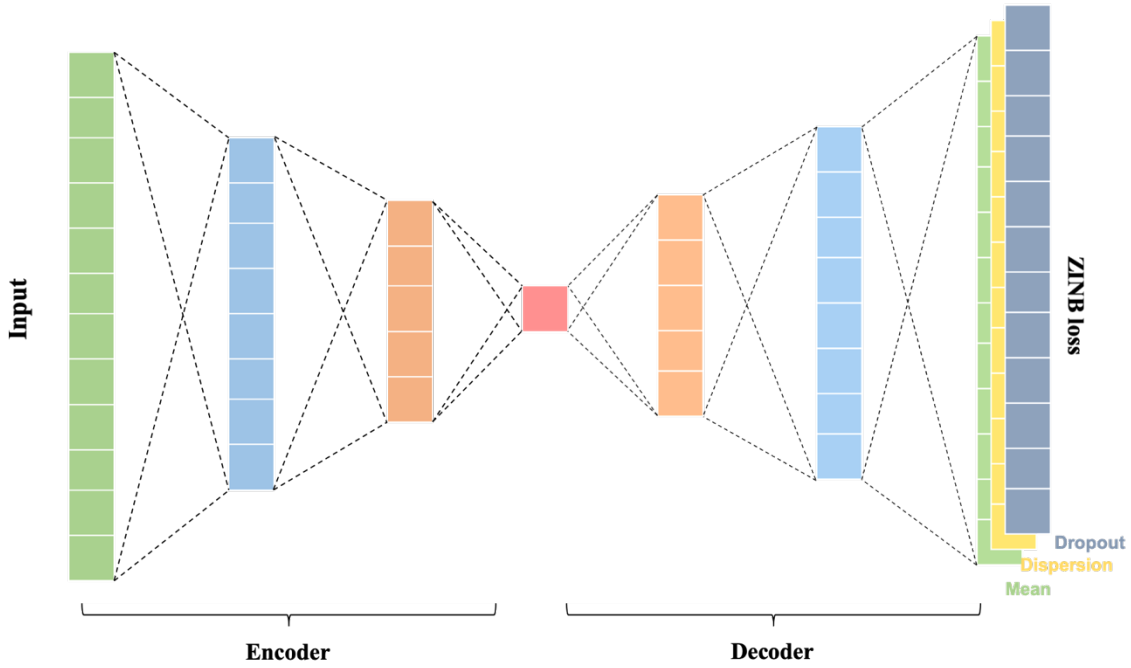


Figure 2.2 Network architecture of ZINB-based autoencoder.

2.2.2 Siamese Network

The Siamese network is one kind of similarity learning approach, which is initially used for image recognition. The network has two sub-networks and the outputs of both are aggregated together for the loss calculation. **Figure 2.2** illustrates a basic model of the Siamese network. Let x_i and x_j be a pair of cells from a training set. Let Y be a binary label of the pair; if the cells x_i and x_j are from the same category, $Y = 1$, and $Y = 0$ otherwise. For the ones with labels $Y = 1$ and those with labels $Y = 0$,we usually call them positive pairs and negative pairs, respectively. As mentioned above, each pair will be sent into the Siamese network which consists of two shared-weights encoders. Suppose that the encoder function mentioned before maps one pair into low-dimensional latent space as $h_i = f(x_i)$ and $h_j = f(x_j)$.

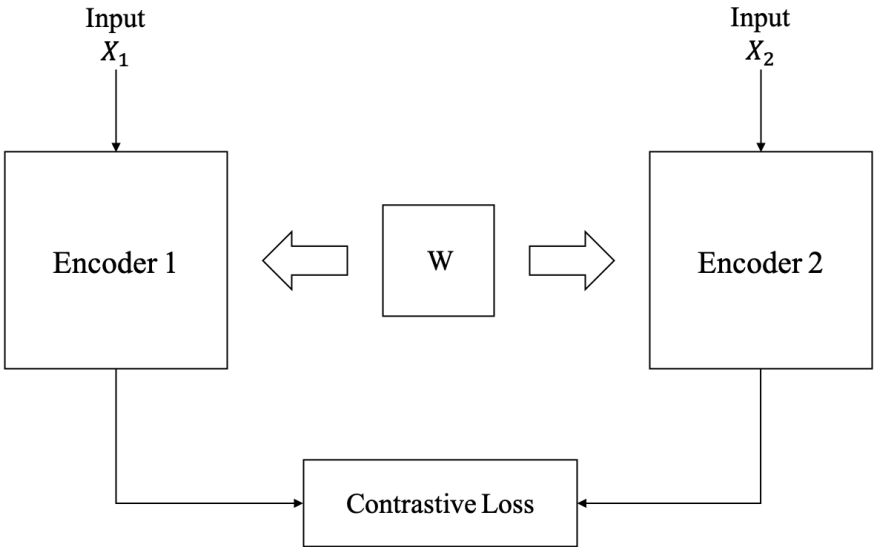


Figure 2.3 Network architecture of Siamese network architecture.

The similarity for $h_i$and h_j, corresponding to x_i and x_j in each pair, is measured by means of the Euclidean distance, which is defined as following:

$$d(h_i,\ h_j) = \left\|h_i - h_j\right\|_2^2$$

Hence, the total contrastive loss for minimizing is defined as:

$$l(x_i, x_j, Y) = \begin{cases} \left\|h_i - h_j\right\|_2^2, & Y = 1, \\ max(0, \omega - \left\|h_i - h_j\right\|_2^2), & Y = 0, \end{cases}$$

where ω is a predefined threshold.

2.2.3 Spectral Clustering

Clustering is an essential process for exploratory data analysis. In this study, spectral clustering algorithms are introduced into the clustering process. Based on the study of some practical issues, the performance of spectral clustering often exceeds that of traditional approaches, such as K-means or single linkage. Moreover, it is easy to implement in any programming language and solved by linear algebra approaches.

SpectralNet overcomes some weak points of the spectral clustering, like scalability and generalization. The major step in SpectralNet is to learn the function F_θ that maps each data points to spectral embedding space while enforcing orthogonality. Let w: $\mathbb{R}^m \times \mathbb{R}^m \to [0, \infty)$ be a symmetric affinity function, such that $w(x_i, x_j)$ represents the similarity between x_i and x_j. In this step, two points, x_i and x_j, that has large value of $w(x_i, x_j)$, should be embedded as close as possible to each other. Therefore, the loss is defined as:

$$L_{spect}(\theta) = \mathrm{E}\left(w(x_i, x_j)\|y_i - y_j\|^2\right),$$

where $y_i,\ y_j \in \mathbb{R}^p$ represent the map function $y = F_\theta(x)$, θ represents the parameters in the map function, and E represents the expectation taken with respect to i.i.d. pair elements drawn from the distribution of the input. To minimize the loss $L_{spect}(\theta)$, we could map all the input points to the same output vector; but this mapping process does no help to the performances. We would enforce the output to be orthonormal in expectation to prevent the dead end:

$$\mathrm{E}(yy^T) = I_{p\times p},$$

As the distribution of the input data remains unknown in most cases, the empirical analogues will replace the expectation in the equations above. In the experiments of this study, at each training epochs, a batch of k samples are randomly selected to calculate the loss:

$$L_{spect}(\theta) = \frac{1}{k^2}\sum_{i,j}^{k} w(x_i, x_j)\|y_i - y_j\|^2,$$

where $y_i = F_\theta(x_i)$.

$$\frac{1}{k}Y^TY = I_{p\times p},$$

where Y is a $k \times p$ matrix of the outputs whose ith row is y_i^T. The orthogonality constraint is implemented by adding one linear-like layer as the last layer of the whole

neural network. The orthogonalization of the matrix Y can be computed through its QR decomposition. In empirical, Cholesky decomposition is frequently performed to obtain the QR decomposition of one matrix A, only if A^TA is full rank. The Cholesky decomposition of the matrix A is denoted as $A^TA = LL^T$, where L is a lower triangular matrix, and setting $Q = A(L^{-1})^T$. Thus, in this study, in order to orthogonalize Y, the last layer multiplies Y from the right side by $\sqrt{k}(L^{-1})^T$.

In SpectralNet, a good affinity matrix is essential for the success of spectral clustering. Due to that our clustering process is using the data transformed by the ZINB model-based autoencoder, we would simply use common Gaussian kernel to generate the affinity matrix in our experiments.

$$W_{i,j} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \text{ if } x_j \text{ is among the nearest neighbors of } x_i$$

2.2.4 Model-Based Deep Siamese Autoencoder

A novel approach which brings together the ZINB model with clustering loss and Siamese networks with contrastive loss, ZINB model-based deep Siamese autoencoder (ZMDSAE), is proposed to deal with unlabeled datasets. Its architecture is shown in **Figure. 2.3**. The network has two subnetworks and their encoders and decoders have shared weights. ZINB loss, denoted as L_{recon}, is taken into consideration in a principled way to improve the clustering performance while doing dimension reduction. At the end of the encoder, one or more layers are added to reduce the data into reasonable dimensions to calculate the

contrastive loss. We will call them as Siamese layer and outputs of this layer as deep embedded representations. Contrastive loss, denoted as L_{siam}, is trying to keep distance in embedding space as that in the original space. The goal of training this network is to optimize the following loss:

$$L = \alpha(L_{ZINB1} + L_{ZINB2}) + \beta L_{siam},$$

where α, β here are weights of different loss terms.

Before training this network, we need to generate positive pairs and negative pairs since we are using unlabeled datasets. A PCA dimensional reduction is performed here to turn the data into a new space (normally into a low dimensional space) such that the greater variance appears on the top coordinates[80]. Afterwards, two cells in which an edge appeared between them will be considered as a positive pair based on the k-nearest neighbors algorithm (k-NN). The rest duo combinations of the whole dataset will be negative pairs. Then we randomly select parts of the pairs for training.

In the first step, we use the data to train one autoencoder with merely the ZINB loss for the backward propagation. After hundreds of epochs, the contrastive loss will be added into the loss calculation that comes from the Siamese network. Although the Siamese network suggests two subnets, in practical, only one is stored due to that their weights are shared. The states of the weights of the encoder will be saved for dimensional reduction, when the training process of the autoencoder is completed. Finally, the spectral clustering

is performed for further improvement of the clustering accuracy. The output of the Siamese layer will be used to calculate the distance matrix in the spectral clustering.

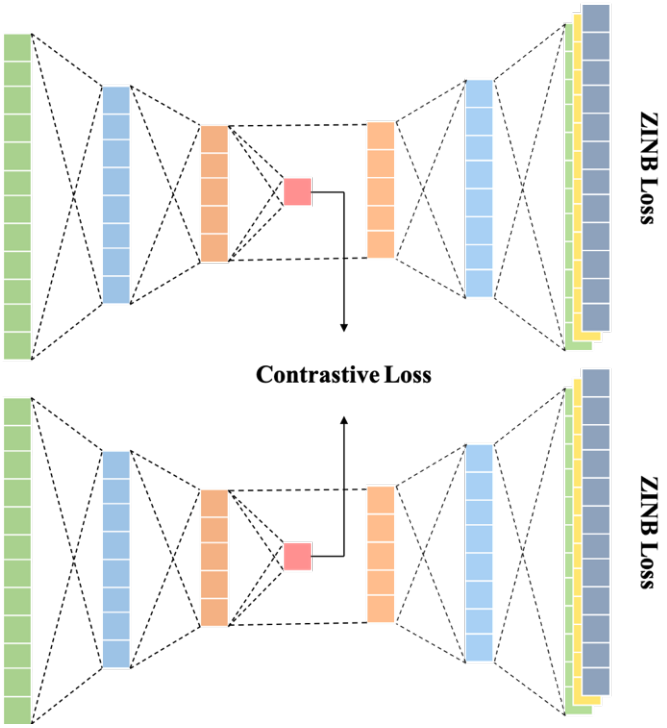


Figure 2.4 Network architecture of model-based deep Siamese autoencoder.

2.3 Competing Method

In this study, three common measures are used for numerical evaluations: the unsupervised clustering accuracy (CA), the normalized mutual information (NMI) ,and adjusted rand index (ARI).

Assuming there are n clusters and let T represents the contingency table with size of $n \times n$, such that T_{ij} is the number of cells that belongs to cluster i but with predicted label j. CA is calculated by compared the compared the number of predicted labels and that of true labels, while NMI and ARI follows the following equations:

$$NMI(Y,\hat{Y}) = \frac{2I(Y;\hat{Y})}{H(Y)+H(\hat{Y})},$$

where Y represents the true label, and $\hat{Y}$ represents the predicted label. $H(Y)$ is entropy function and $I(Y;\hat{Y})$ is the mutual information.

$$ARI = \frac{\sum_i\binom{T_{ii}}{2} - [\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i\binom{a_i}{2}+\sum_j\binom{b_j}{2}] - [\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}]/\binom{n}{2}},$$

where a_i is the sum of the ith row and b_j is the sum of the jth column. The values of CA and NMI are between 0 and 1 while ARI may have negative values. These three metrics are capable to depict the concordance of two clustering label, which means the higher value represent the higher concordance.

DCA, SIMLR, MPSSC, CIDR, PCA + k-means, scvis and DEC are used as competing methods. DCA is conducted directly by using the authors' API functions (https://github.com/theislab/dca). DCA is not designed for clustering. So, we first apply the DCA (with the default parameters given by the authors) to denoise the raw read count data (impute the dropouted counts), then reduce the high-dimensional denoised read count matrix to the 2D space by principal component analysis (PCA). k-means clustering was conducted on the projected 2D space. This method is called 'DCA + k-means'. We pre-process the read count matrix then use the pre-processed data as the input for the SIMLR, PCA + k-means and MPSSC. First, the read count matrix is normalized by library size, so total counts are the same across cells. Next, normalized read counts are log-transformed.

SIMLR is a spectral clustering method, where similarities between cells are learned by multi-kernel. SIMLR is set to use default settings. MPSSC is a multi-kernel spectral clustering framework with the imposition of sparse structures on a target matrix. The parameters for MPSSC are rho = 0.2, lam = 0.0001, lam2 = 0.0001, eta = 1, c = 0.1. PCA + k-means is a method that applies PCA to project the processed raw read count matrix to 2D space directly, followed by k-means clustering. We follow the steps described by the authors for CIDR (https://github.com/VCCRI/CIDR). The input for CIDR is a scData R object constructed by the raw count matrix. The clustering steps for CIDR include determining the dropout events and imputation weighting thresholds, computing the CIDR dissimilarity matrix, reducing the dimensionality and clustering. We use the first two principal components computed by CIDR to show the latent representations. The scvis is a variational autoencoder50 based model used to capture the low-dimensional representation of scRNA-seq data. We use scvis to reduce scRNA-seq data to 2D space then apply k-means clustering. For scvis, we follow the pre-process steps described by the authors: the expression of each gene is quantified as log2(CPM/10 + 1), where 'CPM' stands for 'counts per million'. Next, the data are projected to a 100-dimensional space by PCA and used as input for scvis. DEC (https://github.com/XifengGuo/DEC-keras) uses the same inputs as scDeepCluster: the raw count matrix is library-size normalized, log transformed, scaled and centred. The hyperparameters in DEC remain the same as the authors' originals (for example, the sizes of the hidden layers are 500, 500, 2,000, 10).

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Performance of ZINB-Model Based Autoencoder

To evaluate the performance of this model-based Deep Siamese Autoencoder proposed in this study, we apply it to the real scRNA-seq datasets. The four datasets are generated from four sequencing platforms: PBMC 4k cells from the 10X genomics platform (10X PBMC)[75], worm neuron cells from the sci-RNA-seq platform (worm neuron cells)[78], mouse bladder cells from Microwell-seq platform (mouse bladder cells)[77] and, mouse embryonic stem cells from a droplet barcoding platform (mouse ES cells)[76]. Three common measures, the unsupervised clustering accuracy (CA), the normalized mutual information (NMI) and adjusted rand index (ARI) are used for numerical evaluations.

The four datasets, respectively, have 4271, 4186, 2746 and 2717 cells per sample, with 16653, 13488, 20670, 24175 genes after pre-processing, and form 8, 10, 16 and 4 groups as shown in **Table 3.1**. After a simple PCA dimensional reduction to 2, the distribution of the data with the original labels are showed below in **Figure 3.1** which, meanwhile, illustrates the difficulty of the clustering.

Table 3.1 Summary of Four Real ScRNA-seq Datasets

Dataset	Sequencing platform	Sample size/cell numbers	No. of genes	No. of groups
10X PBMC	10X	4271	16653	8
Worm neuron cells	Sci-RNA-seq	4186	13488	10
Mouse bladder cells	Microwell-seq	2746	20670	16
Mouse ES cells	Droplet barcoding	2717	24175	4

We randomly sampled 2100 cells from each dataset (available at https://github.com /ttgump /scDeepCluster/ tree/master/scRNA-seq%20data)[10].

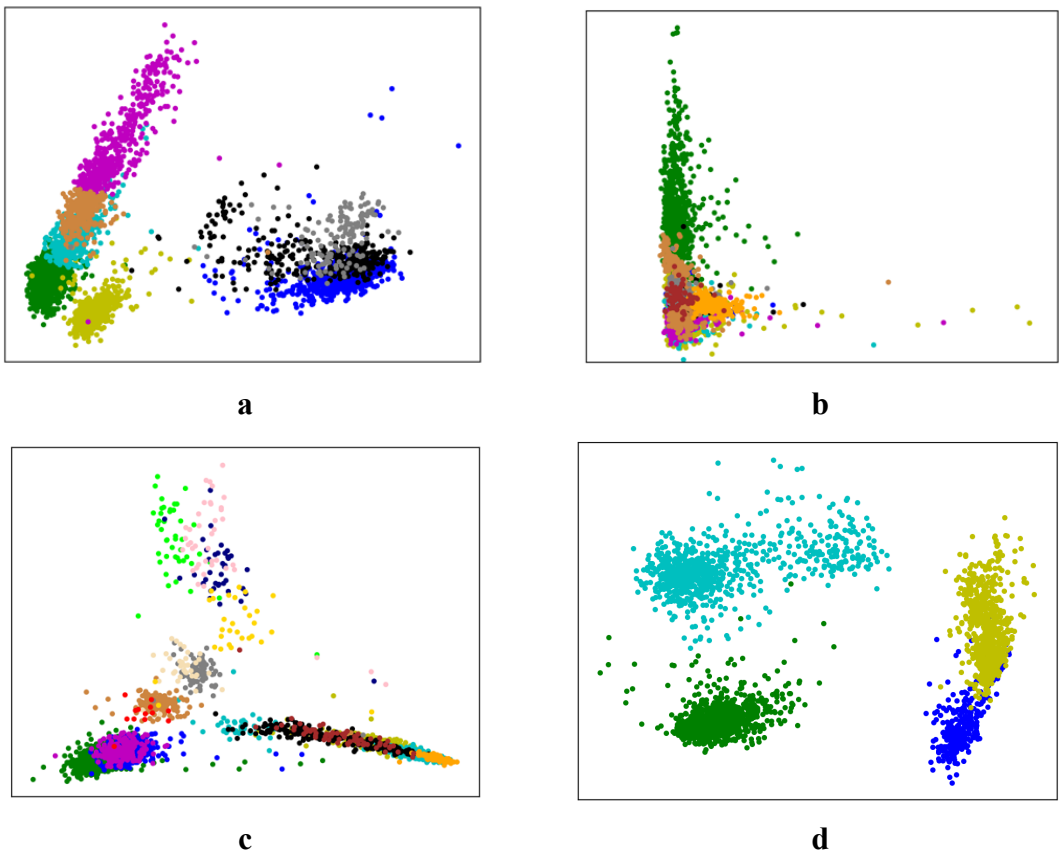


Figure 3.1 Distribution of four datasets directly using PCA. **a**, 10X PBMC. **b**, Worm neuron cells. **c**, Mouse bladder cells. **d**, Mouse ES cells.

In Tian T. and his colleagues' work, scDeepCluster, they randomly selected 2100 cells from each cluster and compared the NMI, CA, ARI with other methods, including

DCA+ k-means, MPSSC, SIMLR, CIDR, PCA + k-means, Scvis + k-means, and DEC. The three metrics are shown in **Figure 3.2**[10], such that the performance of the deep learning clustering scDeepCluster is better than all of other methods in all four datasets. The latent space provided by scDeepCluster also shows great representation effectiveness. **Figure 3.3 – Figure 3.6**[10] illustrate the distribution of the embedded points obtained by applying t-SNE two-dimensional (2D) visualization for four datasets. Noted that for scDeepCluster, only a few points are mixed up with wrong clusters, while other methods fail to provide such clustering performance. Thus, in this study, we would compare the results between our model and scDeepCluster, including metrics of NMI, AC and ARI, and 2D visualization generated by t-SNE.

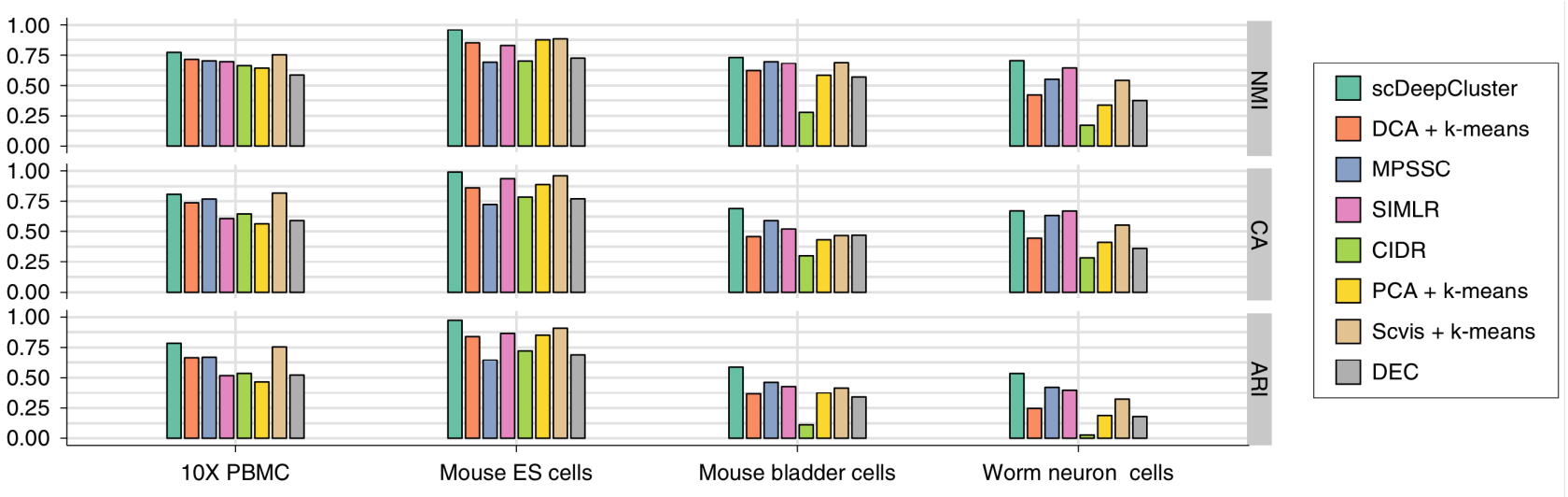


Figure 3.2 Comparison of clustering performances of scDeepCluster, DCA + k-means, MPSSC, SIMLR, CIDR, PCA + k-means, scvis + k-means and DEC, by NMI, CA and ARI.

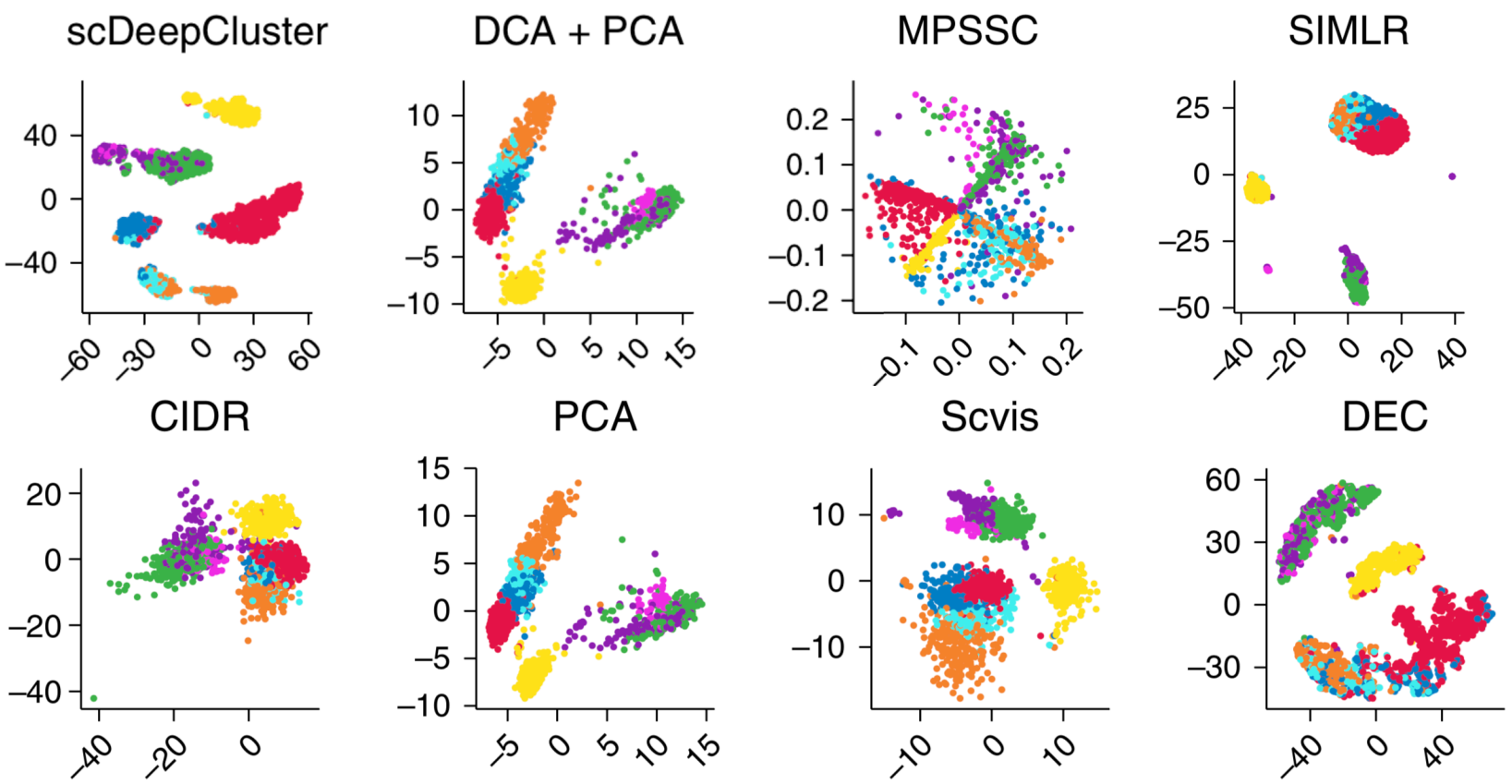


Figure 3.3 Comparison of 2D visualization of embedded representations of 10X PBMC.

Source: [10]

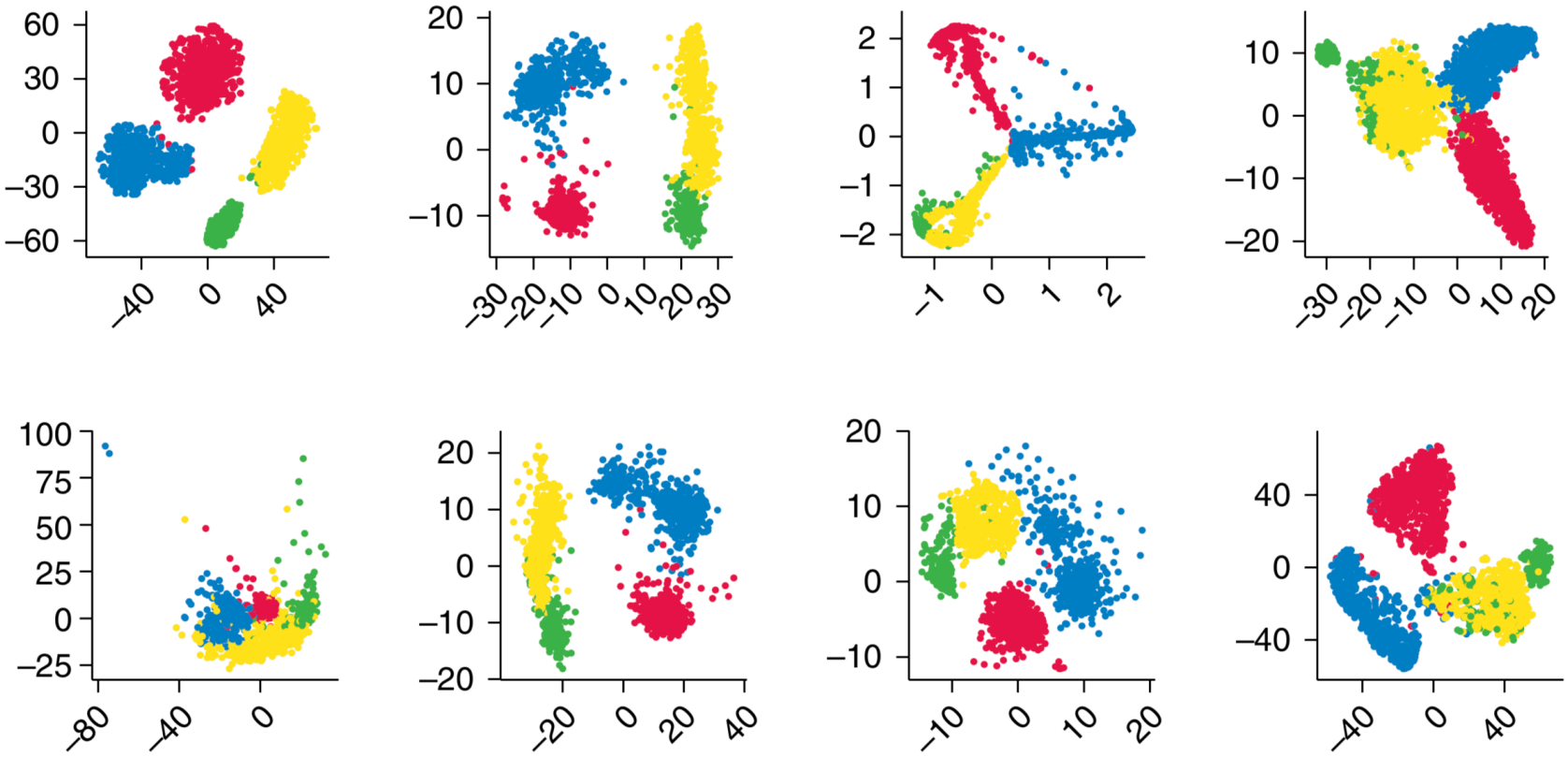


Figure 3.4 Comparison of 2D visualization of embedded representations of mouse ES cells.

Source: [10]

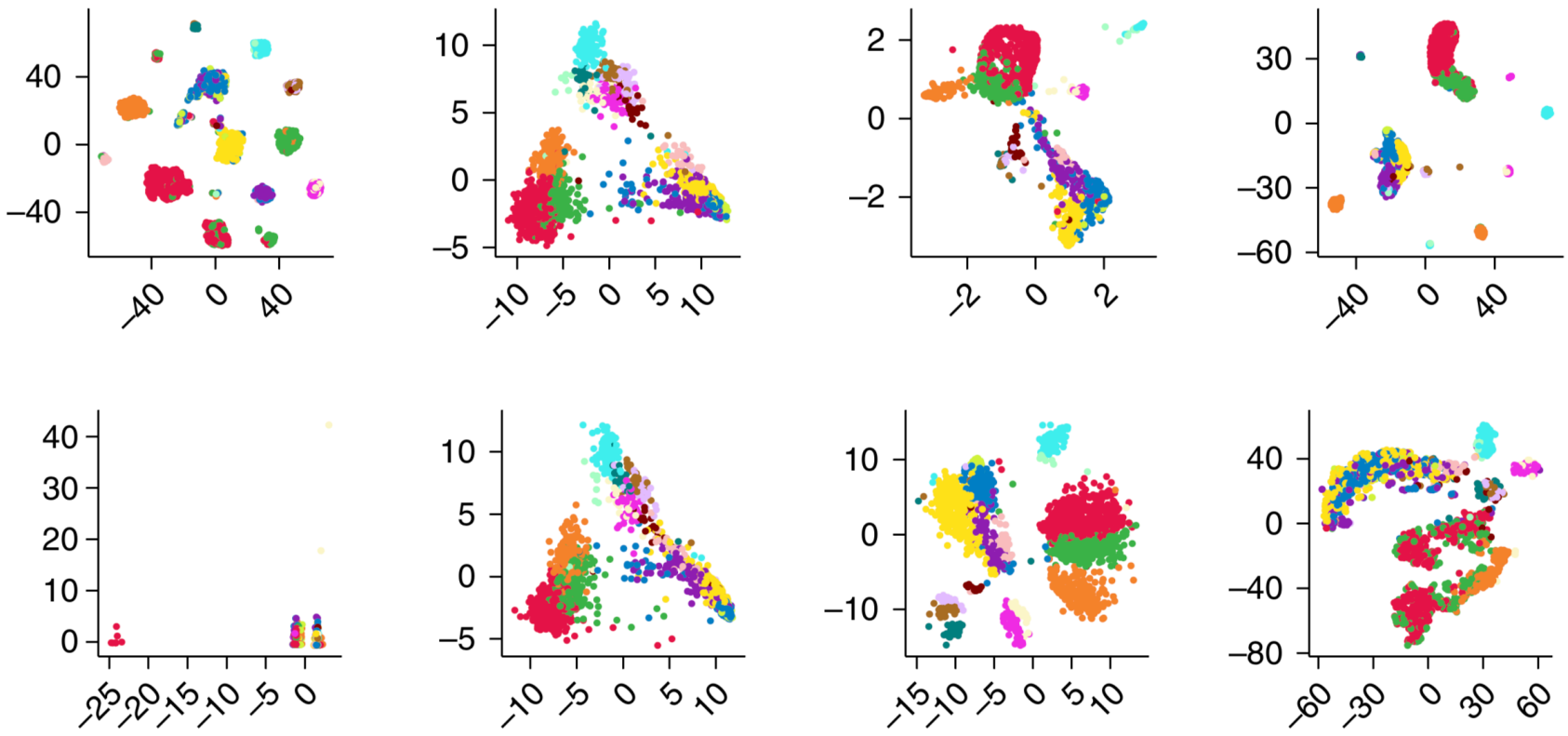


Figure 3.5 Comparison of 2D visualization of embedded representations of mouse bladder cells.

Source: [10]

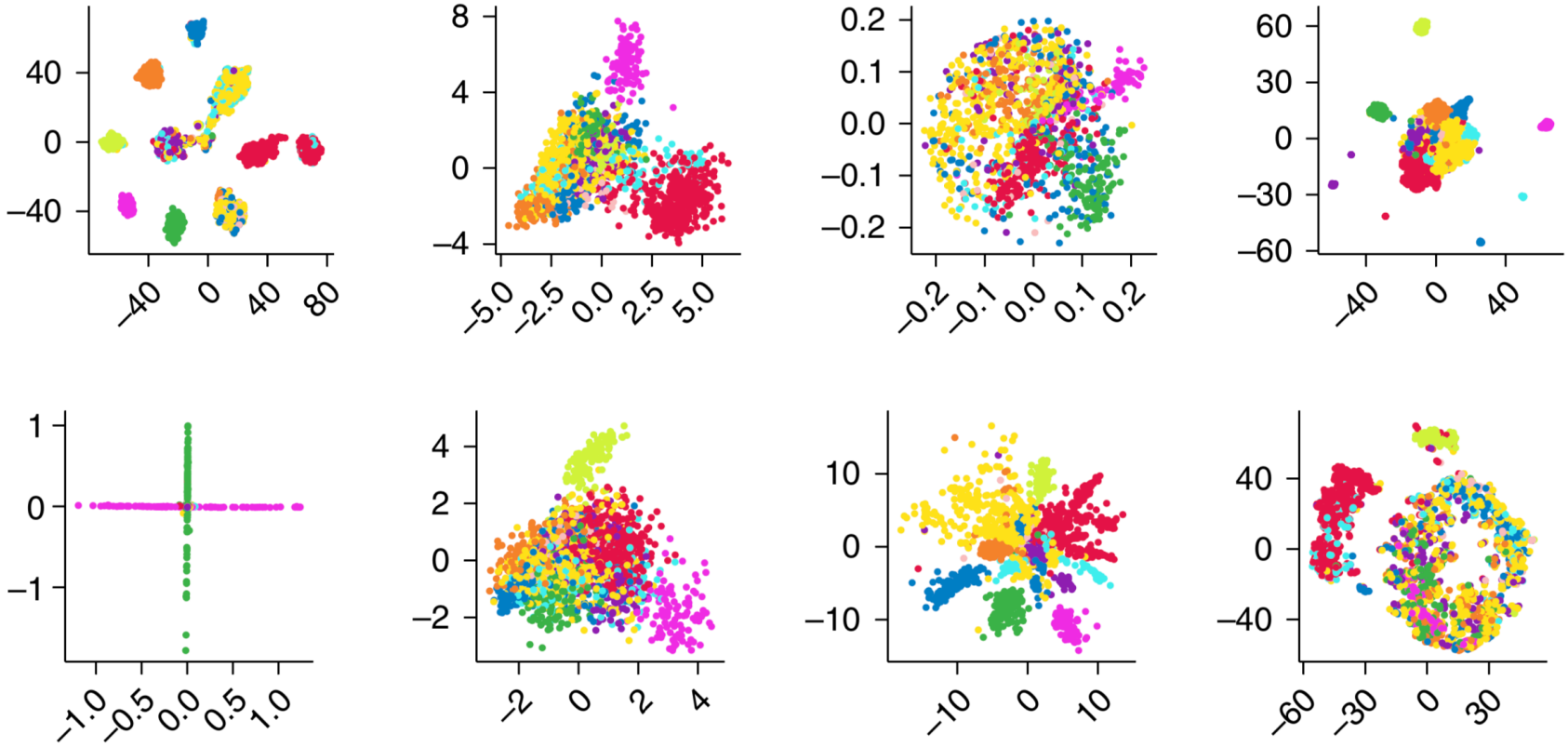


Figure 3.6 Comparison of 2D visualization of embedded representations of worm neuron cells.

Source: [10]

3.2 Performance of ZINB-Model Based Deep Siamese Autoencoder

In our model, a PCA algorithm is first performed in order to get every dataset that has 2100 cells each with 50 representations. Then the k-NN algorithm is used with $k = 10$ to generate positive pairs of cells, while the rest pairs automatically become negative pairs. We randomly sample the equal number of negative pairs from all of them. It is noted that random sampling is taken with the original proportion of the positive pairs and negative ones**. Table 3.2** summaries the detailed information of the selected pairs for every dataset.

Table 3.2 Summary of Selected Pairs for Four Real ScRNA-seq Datasets

Dataset	Positive pairs	Negative pairs	Total pairs
10X PBMC	4902	4902	9804
Worm neuron cells	7387	7387	14774
Mouse bladder cells	5782	5782	11564
Mouse ES cells	5924	5924	11848

Our experiments can be separated into 3 steps. The first step is to pre-train ZINB-based denoising autoencoder, just like scDeepCluster. The second step is to train the same autoencoder with extra contrastive loss between pairs. This step would let the autoencoder start to learn weights in Siamese network. The third step is to perform the spectral clustering on deep embedded outputs of the Siamese layer. In order to evaluate how much improvement is achieved by each training step, we perform K-means clustering process at the end of each step.

We use a n-256-64-32 autoencoder to generate latent space. Here, n means the original dimensions of input data, such as $n = 16653$ for 10X PMBC dataset. The Siamese layer is set as 32-8. Numbers of epochs of the pre-training and training part are

both set as 500. When the autoencoder is pre-training, optimizer Adam (adaptive moment estimation) is used with learning rate starting as 1e-3 and other parameters as default. After pre-training, the network will continue with extra contrastive loss. The learning rate is set as 1e-3 for Siamese layer and 1e-5 for the rest layers. For every 50 epochs, learning rate decay is set to 0.1. Early stopping is also added with patience as 100 epochs. That is, if the loss does not improve in 100 epochs, we will consider the model to be well trained and stop the process. In the loss function, we set $\alpha = 1$ and $\beta = 2$.

In the spectral clustering procedure, we set architecture of SpectralNet as 256-128-64-8. Learning rate is set as 1e-4 at the beginning and decay as 0.1. Early stopping is essential with the patience of 30. In this study, the input space of SpectralNet is the deep embedded space of Siamese layer, so we will not train another Siamese network to estimate affinities. We set k=12 to generate the pairwise affinity metrics using k-NN algorithm, such that one cell would consider the nearest 12 points as neighbors; the distances from other points are set as zero.

The three metrics (NMI, AC and ARI) of clustering performance are calculated by performing k-means algorithm on corresponding space generated by previous process of each step and are visualized in **Figure 3.7,** which illustrates progressively improvement as training steps are carrying on. Detailed information about each dataset are summarized in **Table 3.3** to **Table 3.6**. We observe that, the values of AC, NMI, and ARI do not drop significantly after the dimensional reduction operations between the latent layer and the

Siamese layer. Moreover, the deep embedded space has better accuracy in worm neuron cells and mouse ES cells. It suggests that although the deep embedded space has fewer dimensions than the latent space, the features captured by the Siamese layer are capable of depicting the differences (or similarities) between cells.

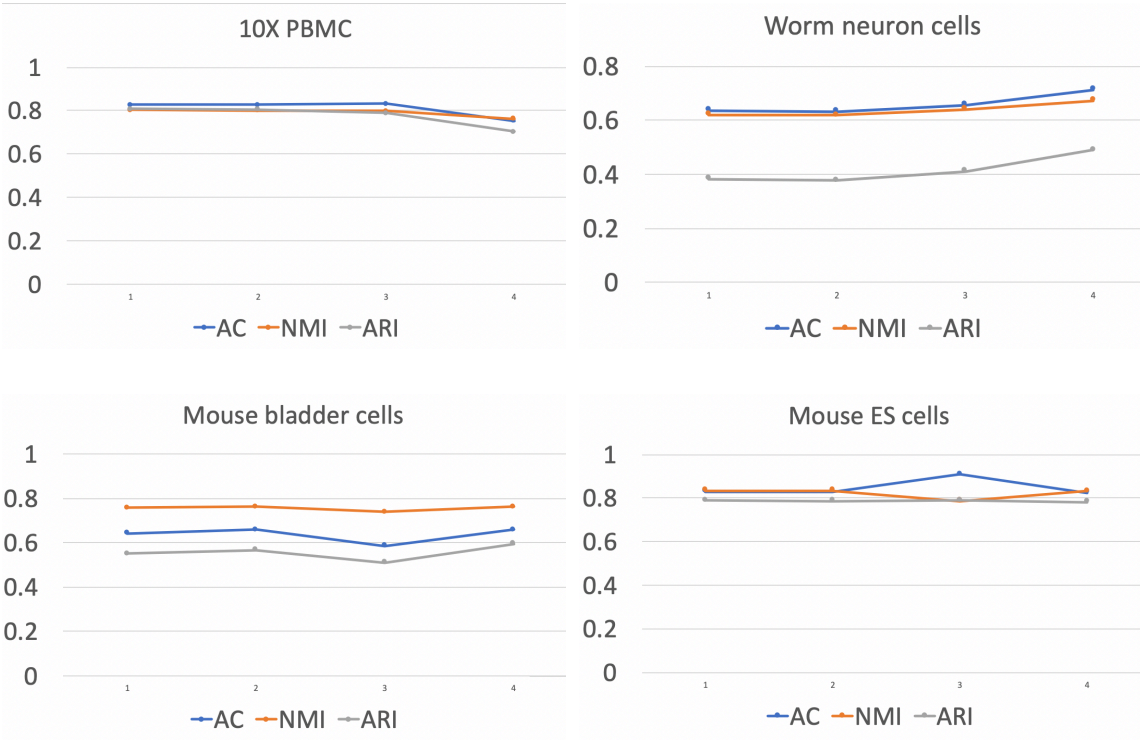


Figure 3.7 Visualizations of AC, NMI, and ARI on four datasets.

Table 3.3 Performance of ScDeepCluster and ZMDSAE on 10X PBMC

Algorithm	Dataset 10X PBMC		
	AC	NMI	ARI
scDeepCluster	.8276	.8024	.8088
ZMDSAE (latent space)	.8271	.8010	.8047
ZMDSAE (deep embedded space)	.8329	.7981	.7879
ZMDSAE (deep embedded space, spectral clustering)	.7533	.7626	.7041

Table 3.4 Performance of scDeepCluster and ZMDSAE on Worm Neuron Cells

Algorithm	Dataset Worm Neuron Cells		
	AC	NMI	ARI
scDeepCluster	.6371	.6196	.3828
ZMDSAE (latent space)	.6333	.6175	.3778
ZMDSAE (deep embedded space)	.6567	.6412	.4111
ZMDSAE (deep embedded space, spectral clustering)	.7138	.6733	.4889

Table 3.5 Performance of scDeepCluster and ZMDSAE on Mouse Bladder Cells

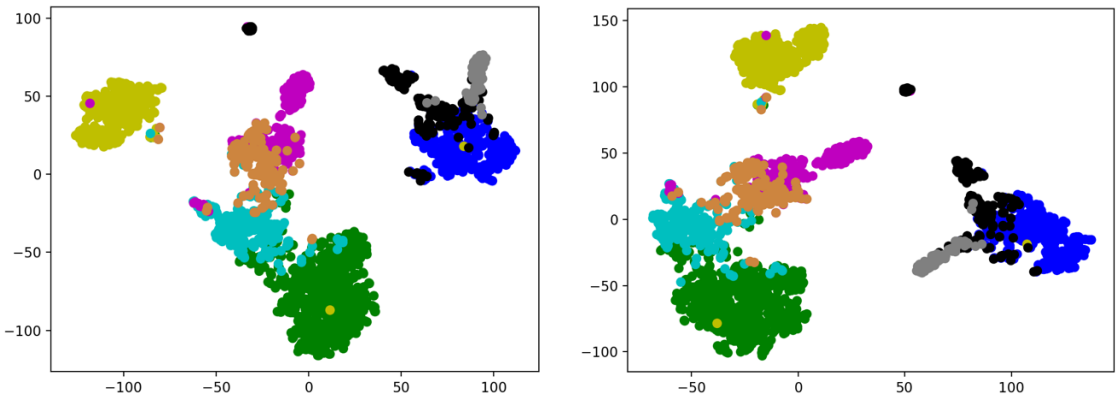
Algorithm	Dataset Mouse Bladder Cells		
	AC	NMI	ARI
scDeepCluster	.6433	.7577	.5496
ZMDSAE (latent space)	.6591	.7626	.5681
ZMDSAE (deep embedded space)	.5867	.7375	.5100
ZMDSAE (deep embedded space, spectral clustering)	.7052	.7795	.6179

Table 3.6 Performance of scDeepCluster and ZMDSAE on Mouse ES Cells

Algorithm	Dataset Mouse ES Cells		
	AC	NMI	ARI
scDeepCluster	.8300	.8354	.7884
ZMDSAE (latent space)	.8295	.8340	.7871
ZMDSAE (deep embedded space)	.9081	.7860	.7887
ZMDSAE (deep embedded space, spectral clustering)	.8248	.8313	.7823

The 2D visualizations of the latent space before and after adding contrastive loss are shown in **Figure 3.8 – Figure 3.9**.

10X PBMC



Worm neuron cells

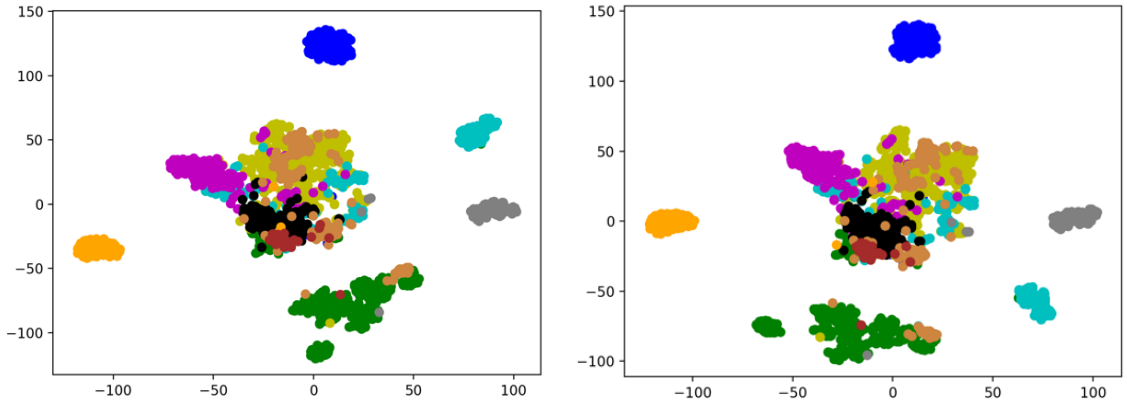
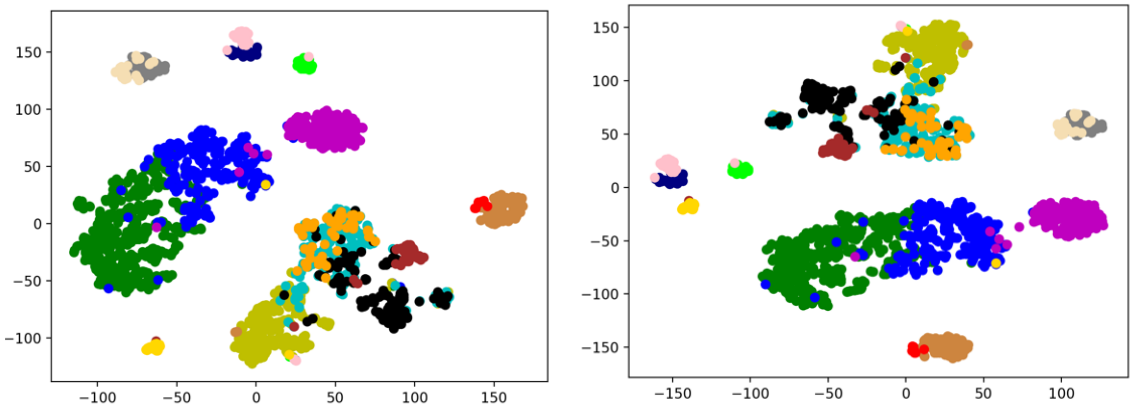


Figure 3.8 2D visualization of latent space generated before (left column) and after (right column) adding the Siamese layer on 10X PBMC and worm neuron cells.

Mouse bladder cells



Mouse ES cells

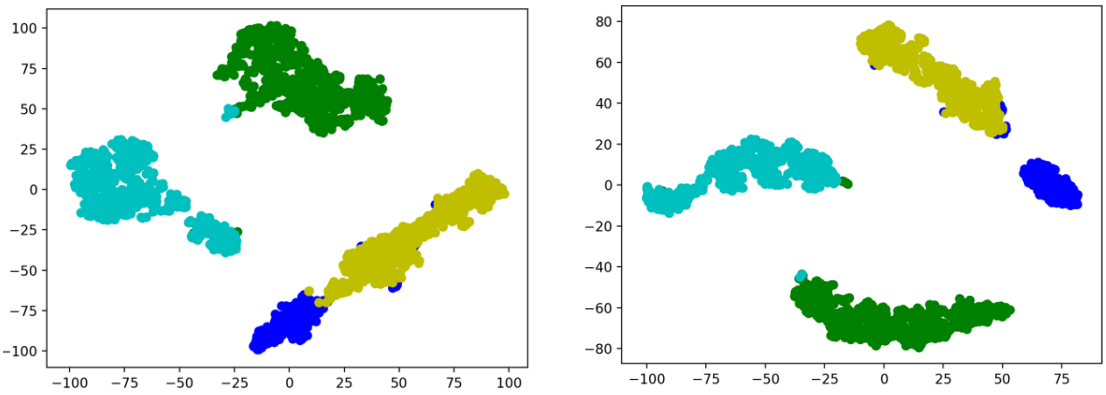


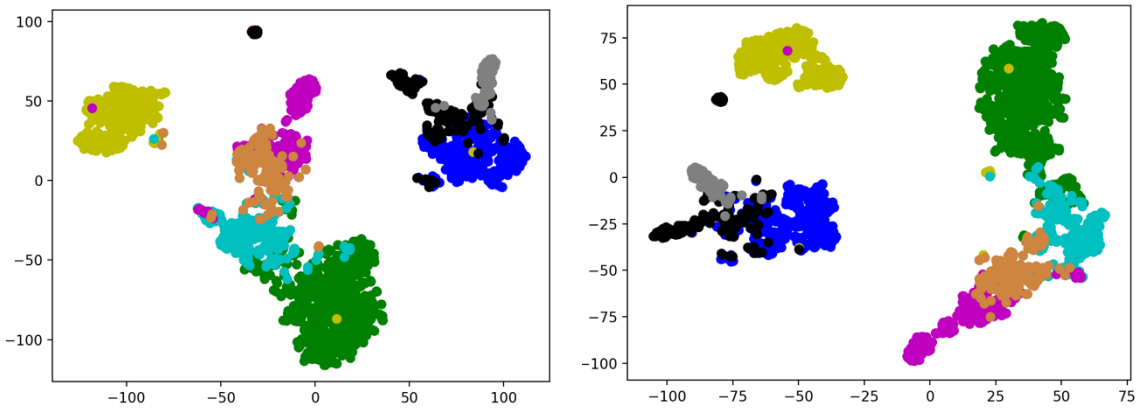
Figure 3.9 2D visualization of latent space generated before (left column) and after (right column) adding the Siamese layer on mouse bladder cells and mouse ES cells.

In **Figure 3.8**, yellow group and green group in 10X PBMC remain separable from other groups, as well as blue group, orange group, lime group, green group and grey group in worm neuron cells. In **Figure 3.9**, for mouse bladder cells data, green group and blue group mix a few points with each other, while purple group, brown group, lime group, red group yellow group are separable at some level. Comparing the left and the right columns, adding the Siamese network to our model will not weaken the clustering performances.

We also show the 2D visualization of deep embedded space to compare with the

latent space before adding the Siamese layer on the four datasets **in Figure 3.10** – **Figure 3.11**. The visualizations suggest that although the deep embedded space has fewer dimensions, the information differentiating cells learned by the autoencoder can be successfully carried on to deep embedded space. Moreover, introducing the contrastive loss into our model improves the clustering performances in 2D visualizations. In **Figure 3.10**, the green group and the lime group have more distance with each other in deep embedded space of 10X PBMC. In **Figure 3.11**, the group, with dark yellow in mouse bladder cells, is completed separated from others in deep embedded space. Two groups in mouse ES cells, gluing to each other in the latent space, also show great clusters in the deep embedded space. Some of the outliers are also divided into appropriate groups, such as points around the yellow group in 10X PMBC and points around the green group in worm neuron cells.

10X PBMC



Worm neuron cells

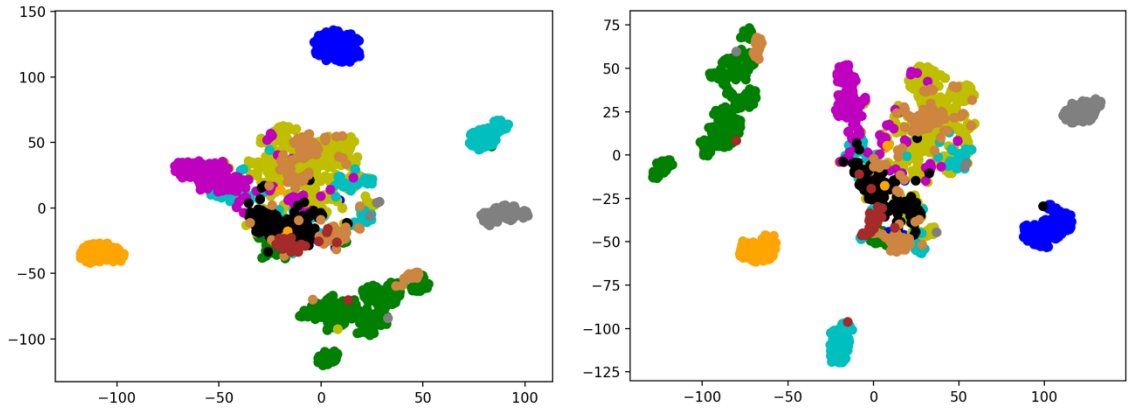


Figure 3.10 2D visualization of latent space generated before adding the Siamese layer (left column) and deep embedded space on mouse bladder cells and mouse ES cells.

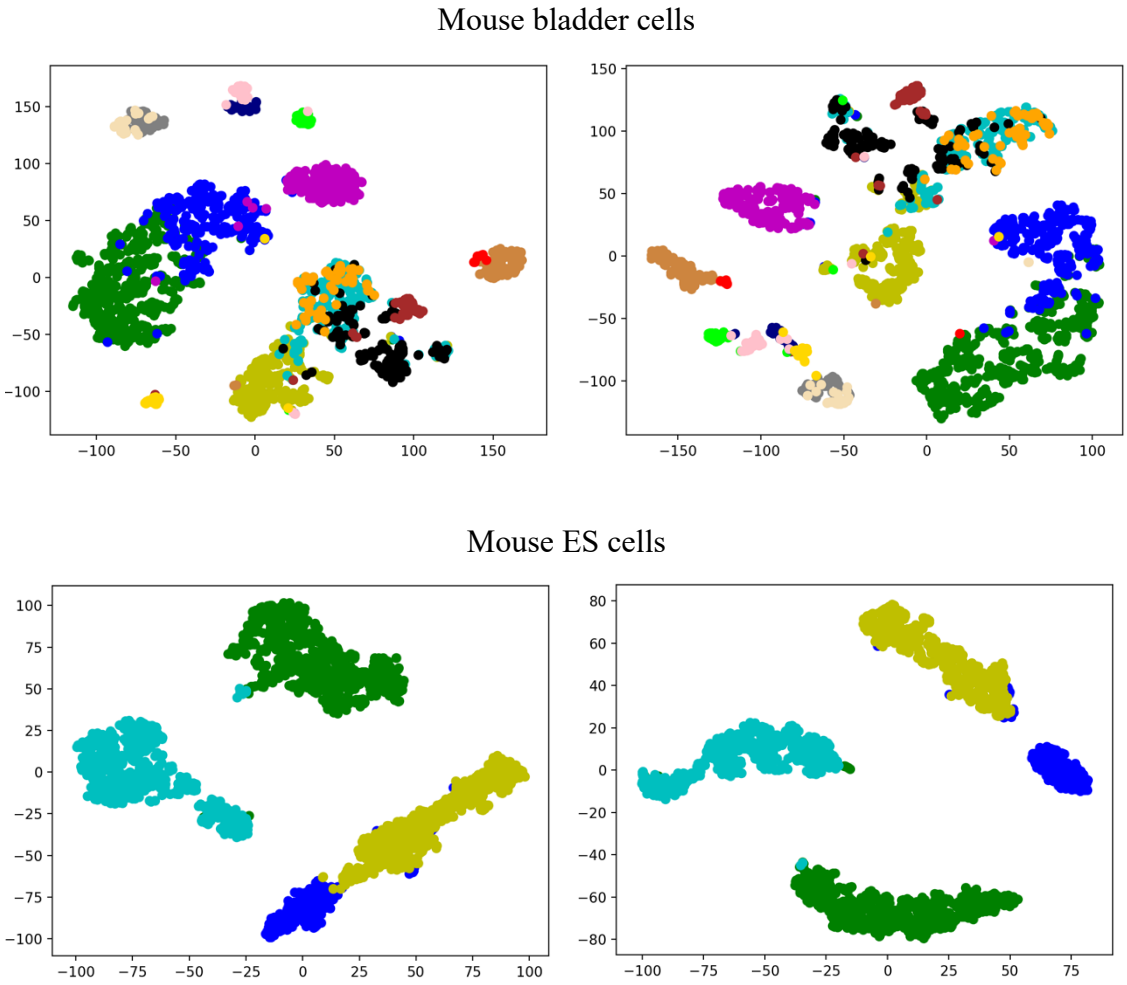


Figure 3.11 2D visualization of latent space generated before adding the Siamese layer (left column) and deep embedded space (right column) on mouse bladder cells and mouse ES cells.

CONCLUSION

In this study, we have proposed a ZINB model-based deep Siamese autoencoder (ZMDSAE) for clustering analysis of scRNA-seq data. The approach can learn a deep embedded representation that is optimized for clustering high-dimensional input in a non-linear manner. In particular, we explicitly model scRNA-seq data generation using a parametric model appropriate for characterizing count data with excessive zeros. Moreover, we introduce a clustering method based on SpectralNet which could efficiently utilizes the learned deep embedded representation. Comparing with our former model, scDeepCluster, real data applications have shown that our model could further improve the clustering performances. Moreover, the main challenge confronted in scRNA-seq data analysis, the pervasive dropout events, can be solved by our model efficiently. In contrast, other previous state-of-the-art spectral clustering methods (MPSSC and SIMLR) rely on multiple Gaussian kernels, which are proved to be less effective in characterizing sparse count data. As an ever-growing number of large-scale scRNA-seq datasets become available, we expect more applications of our method.

REFERENCES

1.Korah, L. V.; Anilkumar, G.; Thomas, S., 5 - Hydrogels, DNA, and RNA polypeptides for the preparation of biomaterials. In *Fundamental Biomaterials: Polymers*, Thomas, S.; Balakrishnan, P.; Sreekala, M. S., Eds. Woodhead Publishing: 2018; pp 85-104.

2.Chu, Y.; Corey, D. R., RNA Sequencing: Platform Selection, Experimental Design, and Data Interpretation. *Nucleic Acid Therapeutics* **2012,** *22* (4), 271-274.

3.Tang, F.; Barbacioru, C.; Wang, Y.; Nordman, E.; Lee, C.; Xu, N.; Wang, X.; Bodeau, J.; Tuch, B. B.; Siddiqui, A.; Lao, K.; Surani, M. A., mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods* **2009,** *6* (5), 377-382.

4.Bumgarner, R., Overview of DNA microarrays: types, applications, and their future. *Curr Protoc Mol Biol* **2013,** *Chapter 22*, Unit-22.1.

5.Saliba, A.-E.; Westermann, A. J.; Gorski, S. A.; Vogel, J., Single-cell RNA-seq: advances and future challenges. *Nucleic acids research* **2014,** *42* (14), 8845-8860.

6.Angerer, P.; Simon, L.; Tritschler, S.; Wolf, F.; Fischer, D.; Theis, F., Single cells make big data: New challenges and opportunities in transcriptomics. *Current Opinion in Systems Biology* **2017,** *4*.

7.Xu, C.; Su, Z., Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics* **2015,** *31* (12), 1974-1980.

8.Sinha, D.; Kumar, A.; Kumar, H.; Bandyopadhyay, S.; Sengupta, D., Dropclust: Efficient clustering of ultra-large scRNA-seq data. *Nucleic Acids Research* **2018,** *46* (6).

9.Jianbo, S.; Malik, J., Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2000,** *22* (8), 888-905.

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

11.Bi, W.; Borgan, C.; Pursley, A. N.; Hixson, P.; Shaw, C. A.; Bacino, C. A.; Lalani, S. R.; Patel, A.; Stankiewicz, P.; Lupski, J. R.; Beaudet, A. L.; Cheung, S. W., Comparison of chromosome analysis and chromosomal microarray analysis: what is the value of chromosome analysis in today's genomic array era? *Genetics in Medicine* **2013,** *15* (6), 450-457.

12.Pollack, J. R.; Perou, C. M.; Alizadeh, A. A.; Eisen, M. B.; Pergamenschikov, A.; Williams, C. F.; Jeffrey, S. S.; Botstein, D.; Brown, P. O., Genome-wide analysis of DNA copy-number changes using cDNA microarrays. *Nature Genetics* **1999,** *23* (1), 41-46.

13.Daoud, M.; Mayo, M., A survey of neural network-based cancer prediction models from microarray data. *Artificial Intelligence in Medicine* **2019,** *97*, 204-214.

14.Pollack, J. R.; Sørlie, T.; Perou, C. M.; Rees, C. A.; Jeffrey, S. S.; Lonning, P.

E.; Tibshirani, R.; Botstein, D.; Børresen-Dale, A.-L.; Brown, P. O., Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors. *Proceedings of the National Academy of Sciences* **2002,** *99* (20), 12963.

15. Tarca, A. L.; Lauria, M.; Unger, M.; Bilal, E.; Boue, S.; Kumar Dey, K.; Hoeng, J.; Koeppl, H.; Martin, F.; Meyer, P.; Nandy, P.; Norel, R.; Peitsch, M.; Rice, J. J.; Romero, R.; Stolovitzky, G.; Talikka, M.; Xiang, Y.; Zechner, C.; Collaborators, I. D., Strengths and limitations of microarray-based phenotype prediction: lessons learned from the IMPROVER Diagnostic Signature Challenge. *Bioinformatics* **2013,** *29* (22), 2892-2899.

16. Zheng, X.; Zhu, W.; Tang, C.; Wang, M., Gene selection for microarray data classification via adaptive hypergraph embedded dictionary learning. *Gene* **2019,** *706*, 188-200.

17. Govindarajan, R.; Duraiyan, J.; Kaliyappan, K.; Palanisamy, M., Microarray and its applications. *J Pharm Bioallied Sci* **2012,** *4* (Suppl 2), S310-S312.

18. Gresham, D.; Dunham, M.; Botstein, D., Comparing whole genomes using DNA microarrays. *Immunity* **2010,** *90*, 201-213.

19. Kononen, J.; Bubendorf, L.; Kallionimeni, A.; Bärlund, M.; Schraml, P.; Leighton, S.; Torhorst, J.; Mihatsch, M. J.; Sauter, G.; Kallionimeni, O.-P., Tissue microarrays for high-throughput molecular profiling of tumor specimens. *Nature Medicine* **1998,** *4* (7), 844-847.

20. Wang, D. G.; Fan, J.-B.; Siao, C.-J.; Berno, A.; Young, P.; Sapolsky, R.; Ghandour, G.; Perkins, N.; Winchester, E.; Spencer, J.; Kruglyak, L.; Stein, L.; Hsie, L.; Topaloglou, T.; Hubbell, E.; Robinson, E.; Mittmann, M.; Morris, M. S.; Shen, N.; Kilburn, D.; Rioux, J.; Nusbaum, C.; Rozen, S.; Hudson, T. J.; Lipshutz, R.; Chee, M.; Lander, E. S., Large-Scale Identification, Mapping, and Genotyping of Single-Nucleotide Polymorphisms in the Human Genome. *Science* **1998,** *280* (5366), 1077.

21. Pelizzola, M.; Pavelka, N.; Foti, M.; Ricciardi-Castagnoli, P., AMDA: an R package for the automated microarray data analysis. *BMC Bioinformatics* **2006,** *7* (1), 335.

22. Russo, G.; Zegar, C.; Giordano, A., Advantages and limitations of microarray technology in human cancer. *Oncogene* **2003,** *22* (42), 6497-6507.

23. Albin Ahmed, M.; Mukhopadhya, A.; Bahar, B., Application of High Throughput Molecular Techniques for Breeding of Farm Animals against Major Diseases. 2016; pp 309-345.

24. Chen, G.; Ning, B.; Shi, T., Single-Cell RNA-Seq Technologies and Related Computational Data Analysis. *Frontiers in Genetics* **2019,** *10* (317).

25. Hwang, B.; Lee, J. H.; Bang, D., Single-cell RNA sequencing technologies and bioinformatics pipelines. *Experimental & Molecular Medicine* **2018,** *50* (8), 96.

26.Wang, X.; Park, J.; Susztak, K.; Zhang, N. R.; Li, M., Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nature Communications* **2019,** *10* (1), 380.
27.Squair, J. A translational approach to understanding and treating autonomic dysfunction after spinal cord injury. Text, 2018.
28.Love, M. I.; Huber, W.; Anders, S., Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* **2014,** *15* (12), 550.
29.Robinson, M. D.; Oshlack, A., A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology* **2010,** *11* (3), R25.
30.Dong, M.; Thennavan, A.; Urrutia, E.; Li, Y.; Perou, C. M.; Zou, F.; Jiang, Y., SCDC: bulk gene expression deconvolution by multiple single-cell RNA sequencing references. *Briefings in Bioinformatics* **2020**.
31.Gong, T.; Szustakowski, J. D., DeconRNASeq: a statistical framework for deconvolution of heterogeneous tissue samples based on mRNA-Seq data. *Bioinformatics* **2013,** *29* (8), 1083-1085.
32.Newman, A. M.; Liu, C. L.; Green, M. R.; Gentles, A. J.; Feng, W.; Xu, Y.; Hoang, C. D.; Diehn, M.; Alizadeh, A. A., Robust enumeration of cell subsets from tissue expression profiles. *Nature Methods* **2015,** *12* (5), 453-457.
33.Shen-Orr, S. S.; Tibshirani, R.; Khatri, P.; Bodian, D. L.; Staedtler, F.; Perry, N. M.; Hastie, T.; Sarwal, M. M.; Davis, M. M.; Butte, A. J., Cell type–specific gene expression differences in complex tissues. *Nature Methods* **2010,** *7* (4), 287-289.
34.Velmeshev, D.; Schirmer, L.; Jung, D.; Haeussler, M.; Perez, Y.; Mayer, S.; Bhaduri, A.; Goyal, N.; Rowitch, D. H.; Kriegstein, A. R., Single-cell genomics identifies cell type–specific molecular changes in autism. *Science* **2019,** *364* (6441), 685.
35.Eberwine, J.; Sul, J.-Y.; Bartfai, T.; Kim, J., The promise of single-cell sequencing. *Nature Methods* **2014,** *11* (1), 25-27.
36.Zeng, Z.; Miao, N.; Sun, T., Revealing cellular and molecular complexity of the central nervous system using single cell sequencing. *Stem Cell Res Ther* **2018,** *9* (1), 234-234.
37.Whitesides, G. M., The origins and the future of microfluidics. *Nature* **2006,** *442* (7101), 368-373.
38.Hebenstreit, D., Methods, Challenges and Potentials of Single Cell RNA-seq. *Biology (Basel)* **2012,** *1* (3), 658-667.
39.Zheng, G. X. Y.; Terry, J. M.; Belgrader, P.; Ryvkin, P.; Bent, Z. W.; Wilson, R.; Ziraldo, S. B.; Wheeler, T. D.; McDermott, G. P.; Zhu, J.; Gregory, M. T.; Shuga, J.; Montesclaros, L.; Underwood, J. G.; Masquelier, D. A.; Nishimura, S. Y.; Schnall-Levin, M.; Wyatt, P. W.; Hindson, C. M.;

Bharadwaj, R.; Wong, A.; Ness, K. D.; Beppu, L. W.; Deeg, H. J.; McFarland, C.; Loeb, K. R.; Valente, W. J.; Ericson, N. G.; Stevens, E. A.; Radich, J. P.; Mikkelsen, T. S.; Hindson, B. J.; Bielas, J. H., Massively parallel digital transcriptional profiling of single cells. *Nature Communications* **2017,** *8* (1), 14049.

40. Zhang, J. M.; Fan, J.; Fan, H. C.; Rosenfeld, D.; Tse, D. N., An interpretable framework for clustering single-cell RNA-Seq datasets. *BMC Bioinformatics* **2018,** *19* (1).

41. Wagner, F.; Barkley, D.; Yanai, I., Accurate denoising of single-cell RNA-Seq data using unbiased principal component analysis. *bioRxiv* **2019**, 655365.

42. Ringnér, M., What is principal component analysis? *Nature Biotechnology* **2008,** *26* (3), 303-304.

43. Leek, J. T., Asymptotic Conditional Singular Value Decomposition for High-Dimensional Genomic Data. *Biometrics* **2011,** *67* (2), 344-352.

44. Chung, N. C.; Storey, J. D., Statistical significance of variables driving systematic variation in high-dimensional data. *Bioinformatics* **2014,** *31* (4), 545-554.

45. Gogolewski, K.; Sykulski, M.; Chung, N. C.; Gambin, A., Truncated Robust Principal Component Analysis and Noise Reduction for Single Cell RNA Sequencing Data. *Journal of Computational Biology* **2019,** *26* (8), 782-793.

46. Lin, P.; Troup, M.; Ho, J. W. K., CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome Biology* **2017,** *18* (1).

47. Taguchi, Y. h., Principal component analysis-based unsupervised feature extraction applied to single-cell gene expression analysis<sup>1</sup&gt. *bioRxiv* **2018**, 312892.

48. Van Der Maaten, L.; Hinton, G., Visualizing data using t-SNE. *Journal of Machine Learning Research* **2008,** *9*, 2579-2625.

49. McKinley, E. T.; Roland, J. T.; Franklin, J. L.; Macedonia, M. C.; Vega, P. N.; Shin, S.; Coffey, R. J.; Lau, K. S., Machine and deep learning single-cell segmentation and quantification of multi-dimensional tissue images. *bioRxiv* **2019**, 790162.

50. Li, W.; Cerise, J. E.; Yang, Y.; Han, H., Application of t-SNE to human genetic data. *Journal of Bioinformatics and Computational Biology* **2017,** *15* (4).

51. Gisbrecht, A.; Schulz, A.; Hammer, B., Parametric nonlinear dimensionality reduction using kernel t-SNE. *Neurocomputing* **2015,** *147* (1), 71-82.

52. Yu, M.; Zhang, S.; Zhao, L.; Kuang, G. In *Deep supervised t-SNE for SAR target recognition*, Proceedings of 2017 2nd International Conference on Frontiers of Sensors Technologies, ICFST 2017, 2017; pp 265-269.

53. Hinton, G. E.; Salakhutdinov, R. R., Reducing the dimensionality of data with neural networks. *Science* **2006,** *313* (5786), 504-507.

54.Zhou, C.; Paffenroth, R. C. In *Anomaly detection with robust deep autoencoders*, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017; pp 665-674.

55.Peng, J.; Wang, X.; Shang, X., Combining gene ontology with deep neural networks to enhance the clustering of single cell RNA-Seq data. *BMC Bioinformatics* **2019,** *20*.

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

57.Chen, J.; King, E.; Deek, R.; Wei, Z.; Yu, Y.; Grill, D.; Ballman, K., An omnibus test for differential distribution analysis of microbiome sequencing data. *Bioinformatics* **2018,** *34* (4), 643-651.

58.Taşdemir, K., Vector quantization based approximate spectral clustering of large datasets. *Pattern Recognition* **2012,** *45* (8), 3034-3044.

59.Cao, J.; Chen, P.; Dai, Q.; Ling, W.-K., Local information-based fast approximate spectral clustering. *Pattern Recognition Letters* **2014,** *38*, 63-69.

60.Shaham, U.; Stanton, K.; Li, H.; Nadler, B.; Basri, R.; Kluger, Y. In *SpectralNet: Spectral clustering using deep neural networks*, 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, 2018.

61.Scher, J. U.; Sczesnak, A.; Longman, R. S.; Segata, N.; Ubeda, C.; Bielski, C.; Rostron, T.; Cerundolo, V.; Pamer, E. G.; Abramson, S. B.; Huttenhower, C.; Littman, D. R., Expansion of intestinal Prevotella copri correlates with enhanced susceptibility to arthritis. *eLife* **2013,** *2013* (2).

62.McMurdie, P. J.; Holmes, S., Waste Not, Want Not: Why Rarefying Microbiome Data Is Inadmissible. *PLoS Computational Biology* **2014,** *10* (4).

63.Anders, S.; Huber, W., Differential expression analysis for sequence count data. *Genome Biology* **2010,** *11* (10).

64.Robinson, M. D.; McCarthy, D. J.; Smyth, G. K., edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **2009,** *26* (1), 139-140.

65.Xu, L.; Paterson, A. D.; Turpin, W.; Xu, W., Assessment and selection of competing models for zero-inflated microbiome data. *PLoS ONE* **2015,** *10* (7).

66.Chen, J.; King, E.; Deek, R.; Wei, Z.; Yu, Y.; Grill, D.; Ballman, K., An omnibus test for differential distribution analysis of microbiome sequencing data. *Bioinformatics* **2017,** *34* (4), 643-651.

67.Wang, B.; Zhu, J.; Pierson, E.; Ramazzotti, D.; Batzoglou, S., Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. *Nature Methods* **2017,** *14* (4), 414-416.

68.Wang, B.; Ramazzotti, D.; De Sano, L.; Zhu, J.; Pierson, E.; Batzoglou, S.,

SIMLR: A Tool for Large-Scale Genomic Analyses by Multi-Kernel Learning. *Proteomics* **2018,** *18* (2).

69. Park, S.; Zhao, H., Spectral clustering based on learning similarity matrix. *Bioinformatics* **2018,** *34* (12), 2069-2076.
70. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P. A. In *Extracting and composing robust features with denoising autoencoders*, Proceedings of the 25th International Conference on Machine Learning, 2008; pp 1096-1103.
71. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P. A., Stacked denoising autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research* **2010,** *11*, 3371-3408.
72. Xie, J.; Girshick, R.; Farhadi, A. In *Unsupervised deep embedding for clustering analysis*, 33rd International Conference on Machine Learning, ICML 2016, 2016; pp 740-749.
73. Chopra, S.; Hadsell, R.; LeCun, Y. In *Learning a similarity metric discriminatively, with application to face verification*, Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, 2005; pp 539-546.
74. Wolf, F. A.; Angerer, P.; Theis, F. J., SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology* **2018,** *19* (1), 15.
75. Zheng, G. X. Y.; Terry, J. M.; Belgrader, P.; Ryvkin, P.; Bent, Z. W.; Wilson, R.; Ziraldo, S. B.; Wheeler, T. D.; McDermott, G. P.; Zhu, J.; Gregory, M. T.; Shuga, J.; Montesclaros, L.; Underwood, J. G.; Masquelier, D. A.; Nishimura, S. Y.; Schnall-Levin, M.; Wyatt, P. W.; Hindson, C. M.; Bharadwaj, R.; Wong, A.; Ness, K. D.; Beppu, L. W.; Deeg, H. J.; McFarland, C.; Loeb, K. R.; Valente, W. J.; Ericson, N. G.; Stevens, E. A.; Radich, J. P.; Mikkelsen, T. S.; Hindson, B. J.; Bielas, J. H., Massively parallel digital transcriptional profiling of single cells. *Nature Communications* **2017,** *8*.
76. Klein, A. M.; Mazutis, L.; Akartuna, I.; Tallapragada, N.; Veres, A.; Li, V.; Peshkin, L.; Weitz, D. A.; Kirschner, M. W., Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* **2015,** *161* (5), 1187-1201.
77. Han, X.; Wang, R.; Zhou, Y.; Fei, L.; Sun, H.; Lai, S.; Saadatpour, A.; Zhou, Z.; Chen, H.; Ye, F.; Huang, D.; Xu, Y.; Huang, W.; Jiang, M.; Jiang, X.; Mao, J.; Chen, Y.; Lu, C.; Xie, J.; Fang, Q.; Wang, Y.; Yue, R.; Li, T.; Huang, H.; Orkin, S. H.; Yuan, G. C.; Chen, M.; Guo, G., Mapping the Mouse Cell Atlas by Microwell-Seq. *Cell* **2018,** *172* (5), 1091-1107.e17.
78. Cao, J.; Packer, J. S.; Ramani, V.; Cusanovich, D. A.; Huynh, C.; Daza, R.; Qiu, X.; Lee, C.; Furlan, S. N.; Steemers, F. J.; Adey, A.; Waterston, R. H.;

Trapnell, C.; Shendure, J., Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science* **2017,** *357* (6352), 661-667.

79.Nair, V.; Hinton, G. E. In *Rectified linear units improve Restricted Boltzmann machines*, ICML 2010 - Proceedings, 27th International Conference on Machine Learning, 2010; pp 807-814.

80.T., J. I., Principal Component Analysis. 2nd ed. ed.; Springer: 2002; p. 487.