ABSTRACT

MODEL-BASED DEEP AUTOENCODERS FOR CLUSTERING SINGLE-CELL RNA SEQUENCING DATA WITH SIDE INFORMATION

by
Xiang Lin

Clustering analysis has been conducted extensively in single-cell RNA sequencing (scRNA-seq) studies. scRNA-seq can profile tens of thousands of genes' activities within a single cell. Thousands or tens of thousands of cells can be captured simultaneously in a typical scRNA-seq experiment. Biologists would like to cluster these cells for exploring and elucidating cell types or subtypes. Numerous methods have been designed for clustering scRNA-seq data. Yet, single-cell technologies develop so fast in the past few years that those existing methods do not catch up with these rapid changes and fail to fully fulfil their potential. For instance, besides profiling transcription expression levels of genes, recent single-cell technologies can capture other auxiliary information at the single-cell level, such as protein expression (multi-omics scRNA-seq) and cells' spatial location information (spatial-resolved scRNA-seq). Most existing clustering methods for scRNA-seq are performed in an unsupervised manner and fail to exploit available side information for optimizing clustering performance.

This dissertation focuses on developing novel computational methods for clustering scRNA-seq data. The basic models are built on a deep autoencoder (AE) framework, which is coupled with a ZINB (zero-inflated negative binomial) loss to characterize the zero-inflated and over-dispersed scRNA-seq count data. To integrate multi-omics scRNA-seq data, a multimodal autoencoder (MAE) is employed. It applies one encoder for the multimodal inputs and two decoders for reconstructing each omics of data. This model is named scMDC (Single-Cell Multi-omics Deep Clustering). Besides, it is expected that cells in spatial proximity tend to be of the same cell types. To exploit cellular spatial information available for spatial-resolved scRNA-seq (sp-scRNA-seq) data, a novel model, DSSC (Deep Spatial-constrained Single-cell Clustering), is developed. DSSC integrates the spatial information of cells into the clustering process by two steps: 1) the spatial information is encoded by using a graphical neural network model; 2) cell-to-cell constraints are built based on the spatially expression pattern of the marker genes and added in the model to guide the clustering process. DSSC is the first model which can utilize the information from both the spatial coordinates and the marker genes to guide the cell/spot clustering. For both scMDC and DSSC, a clustering loss is optimized on the bottleneck layer of autoencoder along with the learning of feature representation. Extensive experiments on both simulated and real datasets demonstrate that scMDC and DSSC boost clustering performance significantly while costing no extra time and space during the training process. These models hold great promise as valuable tools for harnessing the full potential of state-of-the-art single-cell data.

MODEL-BASED DEEP AUTOENCODERS FOR CLUSTERING SINGLE-CELL RNA SEQUENCING DATA WITH SIDE INFORMATION

by
Xiang Lin

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Department of Computer Science

December 2023

APPROVAL PAGE

MODEL-BASED DEEP AUTOENCODERS FOR CLUSTERING SINGLE-CELL RNA SEQUENCING DATA WITH SIDE INFORMATION

Xiang Lin

Zhi Wei, Dissertation Advisor — Date
Professor of Computer Science, NJIT

Ioannis Koutis, Committee Member — Date
Associate Professor of Computer Science, NJIT

Wenge Guo, Committee Member — Date
Associate Professor of Mathematical Sciences, NJIT

Junwen Wang, Committee Member — Date
Professor of Applied Oral Sciences, University of Hong Kong,
Hong Kong, China

Nan Gao, Committee Member — Date
Professor of Cell Biology, Rutgers University, Newark, New Jersey

Yao Ma, Committee Member — Date
Assistant Professor of Computer Science,
Rensselaer Polytechnic Institute, Troy, New York

BIOGRAPHICAL SKETCH

Author: Xiang Lin

Degree: Doctor of Philosophy

Date: December 2023

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2023

- Master of Science in Biology,
  New Jersey Institute of Technology, Newark, NJ, 2019

- Bachelor of Education in Kinesiology,
  Fuyang Normal University, Anhui, People's Republic of China, 2012

Major: Computer Science

Presentations and Publications:

Lin, X., Tian, T., Wei, Z. et al. (2022). Clustering of single-cell multi-omics data with a multimodal deep learning method. Nature Communications 13, 7705.

Lin, X., Gao, L., Whitener, N., Ahmed, A., & Wei, Z. (2022). A model-based constrained deep learning clustering approach for spatially resolved single-cell data. Genome Research, 32(10), 1906-1917.

Lin, X., Liu, H., Wei, Z., Roy, S. B., & Gao, N. (2022). An active learning approach for clustering single-cell RNA-seq data. Laboratory Investigation, 102(3), 227-235.

Lin, X., Zhang, J., Wei, Z., & Turki, T. (2021). An Omnibus Test for Differential Distribution Analysis of Continuous Microbiome Data. IEEE Access, 9, 100029-100039.

Lin, X., Ren, J., Gao, L., Wang, J., & Wei, Z. (2023). scDILT: a model-based and constrained deep learning framework for single-cell Data Integration, Label Transferring, and clustering. BioRxiv, 2023-10.

Seckar, T., Lin, X., Bose, D., Wei, Z., Rohrbaugh, J., Collman, R. G., & Robertson, E. S. (2021). Detection of Microbial Agents in Oropharyngeal and Nasopharyngeal Samples of SARS-CoV-2 Patients. Frontiers in Microbiology, 12, 454

Tian, T., Zhong, C., Lin, X., Wei, Z., & Hakonarson, H. (2023). Complex hierarchical structures in single-cell genomics data unveiled by deep hyperbolic manifold learning. Genome Research, 33(2), 232-246.

Tian, T., Zhang, J., Lin, X., Wei, Z., & Hakonarson, H. (2021). Model-based deep embedding for constrained clustering analysis of single cell RNA-seq data. Nature Communications, 12(1), 1-12.

Bose, D., Lin, X., Gao, L., Wei, Z., Pei, Y., & Robertson, E. S. (2023). Attenuation of IFN signaling due to m6A modification of the host epitranscriptome promotes EBV lytic reactivation. Journal of Biomedical Science, 30(1), 1-17.

Tong, C.C., Lin, X., Seckar, T., Koptyra, M., Kohanski, M.A., Cohen, N.A., Kennedy, D.W., Adappa, N.D., Papagiannopoulos, P., Kuan, E.C. & Baranov, E. (2023). A Metagenomic Analysis of the Virome of Inverted Papilloma and Squamous Cell Carcinoma. In International Forum of Allergy & Rhinology.

Wen, L., Lin, X., Li, C., Zhao, Y., Yu, Z., & Han, X. (2022). Sagittal imbalance of the spine is associated with poor sitting posture among primary and secondary school students in China: a cross-sectional study. BMC Musculoskeletal Disorders, 23(1), 1-12

Joseph, I., Flores, J., Farrell, V., Davis, J., Bianchi ‐ Smak, J., Feng, Q., Goswami, S., Lin, X., Wei, Z., Tong, K. & Feng, Z. (2023). RAB11A and RAB11B control mitotic spindle function in intestinal epithelial progenitor cells. EMBO Reports, 24(9), p.e56240.

Grover, S., Seckar, T., Gao, L., Bhatia, R., Lin, X., Zetola, N., Ramogola-Masire, D. & Robertson, E. (2023). Characterization of HPV subtypes in invasive cervical cancer in Botswana patients using a pan-pathogen microarray technology. Tumour Virus Research, 15, p.200262.

Wei, S., Yin, D., Yu, S., Lin, X., Savani, M.R., Du, K., Ku, Y., Wu, D., Li, S., Liu, H. & Tian, M. (2022). Antitumor activity of a mitochondrial-targeted HSP90 inhibitor in gliomas. Clinical Cancer Research, 28(10), pp.2180-2195.

Tong, C.C., Koptyra, M., Raman, P., Rathi, K.S., Choudhari, N., Lin, X., Seckar, T., Wei, Z., Kohanski, M.A., O'Malley, B.W. & Cohen, N.A. (2022). Targeted gene expression profiling of inverted papilloma and squamous cell carcinoma. In International Forum of Allergy & Rhinology (Vol. 12, No. 2, pp. 200-209).

Das, S., Feng, Q., Balasubramanian, I., Lin, X., Liu, H., Pellón-Cardenas, O., Yu, S., Zhang, X., Liu, Y., Wei, Z. & Bonder, E.M. (2022). Colonic healing requires Wnt produced by epithelium as well as Tagln+ and Acta2+ stromal cells. Development, 149(1), p.dev199587.

Yu, S., Wei, S., Savani, M., Lin, X., Du, K., Mender, I., Siteni, S., Vasilopoulos, T., Reitman, Z.J., Ku, Y. & Wu, D. (2021). A Modified Nucleoside 6-Thio-2′-Deoxyguanosine Exhibits Antitumor Activity in Gliomas. Clinical Cancer Research, 27(24), pp.6800-6814.

Rajasekaran, K., Carey, R.M., Lin, X., Seckar, T.D., Wei, Z., Chorath, K., Newman, J.G., O'Malley, B.W., Weinstein, G.S., Feldman, M.D. & Robertson, E. (2021). The microbiome of HPV-positive tonsil squamous cell carcinoma and neck metastasis. Oral Oncology, 117, p.105305.

Gromeier, M., Brown, M.C., Zhang, G., Lin, X., Chen, Y., Wei, Z., Beaubier, N., Yan, H., He, Y., Desjardins, A. & Herndon, J.E. (2021). Very low mutation burden is a feature of inflamed recurrent glioblastomas responsive to cancer immunotherapy. Nature Communications, 12(1), p.352.

Brown, M., Zhang, G., Stevenson, K., Lin, X., Chen, Y., Wei, Z., Beaubier, N., Yan, H., He, Y., Desjardins, A. & Herndon, J. (2021). Tumor-intrinsic and peripheral features associate with survival after polio virotherapy in recurrent GBM. Neuro-Oncology, 23(Supplement_6), pp.vi14-vi15.

Yu, S., Balasubramanian, I., Laubitz, D., Tong, K., Bandyopadhyay, S., Lin, X., Flores, J., Singh, R., Liu, Y., Macazana, C. & Zhao, Y. (2020). Paneth cell-derived lysozyme defines the composition of mucolytic microbiota and the inflammatory tone of the intestine. Immunity, 53(2), pp.398-416.

Carey, R.M., Rajasekaran, K., Seckar, T., Lin, X., Wei, Z., Tong, C.C., Ranasinghe, V.J., Newman, J.G., O'Malley Jr, B.W., Weinstein, G.S. & Feldman, M.D. (2020). The virome of HPV-positive tonsil squamous cell carcinoma and neck metastasis. Oncotarget, 11(3), p.282.

Cardinale, C.J., March, M.E., Lin, X., Liu, Y., Spruce, L.A., Bradfield, J.P., Wei, Z., Seeholzer, S.H., Grant, S.F. & Hakonarson, H. (2020). Regulation of Janus kinase 2 by an inflammatory bowel disease causal non-coding single nucleotide polymorphism. Journal of Crohn's and Colitis, 14(5), pp.646-653.

To My Beloved Parents.

ACKNOWLEDGMENTS

Foremost, I would like to express my deepest appreciation to my dissertation advisor, Dr. Zhi Wei, for his extreme kindness, patience and encouragement throughout my entire PhD studying period. He is a knowledgeable and supportive professor, who can always give invaluable insight into the nature of problems and provide practical advice. It is a great honor to work under his guidance. I could not have been able to complete this study without him.

Secondly, I would like to express my appreciation to Dr. Ioannis Koutis, Dr. Wenge Guo, Dr. Nan Gao, and Dr. Yao Ma for their kind advice and instruction in my study and research career.

My studies were supported by grant R15HG012087 from the National Institutes of Health, and the National Center for Advancing Translational Sciences (NCATS), a component of NIH under award number UL1TR003017. The computing resource for my study was partially provided by Extreme Science and Engineering Discovery Environment (XSEDE) through allocation CIE160021 and CIE170034 (supported by National Science Foundation Grant No. ACI-1548562).

Thirdly, I would like to acknowledge the assistance of my friend, Dr. Tian Tian. He had deep insights in my research fields, and he provided me with various ingenious suggestions. In addition, I would like to recognize all members of our group, who also give me lots of help on my research.

Lastly, my special thanks to my beloved parents: Zhaohuai Lin and Ping Jiang. I will always appreciate their unconditional support and encouragement during my PhD study.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

1.1 Multi-omics scRNA-seq

Single-cell RNA sequence (scRNA-seq) profiles a high-resolution picture inside an individual cell. Based on the scRNA-seq technology, recently, many multimodal sequencing technologies have been developed to jointly profile multiple modalities of data in a single cell. For example, cellular Indexing of Transcriptomes and Epitopes by Sequencing (CITE-seq) (**Figure 1.1**) and RNA expression and protein sequencing assay (REAP-seq) have been developed to profile mRNA expression and quantify surface protein simultaneously at cellular level (Mimitou et al., 2019; Peterson et al., 2017). Specifically, CITE-Seq employs existing single-cell sequencing technologies, such as the 10X Genomics Chromium platform (Zheng et al., 2017), and allows the counting of Antibody-Derived Tags (ADT) to quantify the cell surface protein abundance. Each cell with ADT labels and DNA-barcoded microbeads will be encapsulated in a droplet for the single-cell sequencing (Stoeckius et al., 2017). REAP-seq also combines DNA-barcoded antibodies with existing scRNA-seq approaches to measure the expression levels of genes and cell-surface proteins (Peterson et al., 2017). In addition to studying single-cell transcriptomes and surface proteins, recently, the development of single-cell approaches for the assay of the transposase accessible chromatin sequencing (scATAC-seq) provides us a chance to measure chromatin accessibility in a single cell (Buenrostro et al., 2015).

Specifically, these technologies are designed to identify open chromatin regions in the genome by using the hyperactive Tn5 transposase, which simultaneously tags and fragments DNA sequences in open chromatin regions (Cusanovich et al., 2015). The scATAC-seq enables us to explore cell type-specific biological activities by investigating the chromatin-accessibility signatures, such as the transcription factors that control the gene expression of cells. More recently, some multi-omics single-cell technologies have been developed to jointly profile chromatin accessibility and gene expression within a single cell (Ma, McDermaid, Xu, Chang, & Ma, 2020), such as SNARE-seq and 10X Single-Cell Multiome ATAC + Gene Expression (we denote it as SMAGE-seq) (S. Chen, Lake, & Zhang, 2019; S. Ma et al., 2020). Overall, these multimodal sequencing technologies provide us with a more comprehensive and complicated profile of a single cell. Therefore, computational tools for jointly integrating different data views for downstream analyses, such as clustering, are desired for exploiting these new powerful experimental technologies.

It is noted that in the multimodal data, the biological information provided by different modalities is complementary (Peterson et al., 2017; Stoeckius et al., 2017), and each modality generally has its own strengths and weaknesses. Using CITE-seq as an example, its ADT modal focuses on surface proteins. ADT data have demonstrated a low dropout rate [4] and thus can reliably quantify gene activities. For the five CITE-seq datasets analyzed in this study, we observed dropout rates of up to 12% in ADT data. In contrast, there were more than 80% or even 90% zero entries in its corresponding mRNA data. For most genes, protein is the final product to fulfill their functions and messenger RNA is an immediate product. Thus, ADT data seems ideal for characterizing cell functions and types. However, due to current technique limits, ADT can profile only up to a couple of hundreds of genes. Because of this limit, investigators generally include marker genes for well-known cell types in ADT modal first. Therefore, ADT data is good at identifying common cell types (Stoeckius et al., 2017; X. Wang et al., 2020), such as CD4+ and CD8+ T cells, when their marker genes are profiled.

However, because of its limited dimensions, ADT data may not detect rare or minor cell types well. In contrast, the full transcriptome of mRNA data can capture comprehensive cell types. Nevertheless, clustering cells based on scRNA-seq may be challenged by its large dropout rate and sparse signal with high dimensionality. Furthermore, the quantity of ADT and mRNA sources produced by the same gene may not be the same when considering the post-transcriptional and post-translational regulations (Haider & Pal, 2013; Stoeckius et al., 2017). In this case, ADT and mRNA data provide complementary information in cell type identification (X. Wang et al., 2020). For SNARE-seq and SMAGE-seq, scATAC-seq data provide chromatin accessibility information which is also complementary to mRNA data (S. Chen et al., 2019). Thus, by integrating the information from multimodalities, we should be able to arrive at a higher resolution of cell typing.

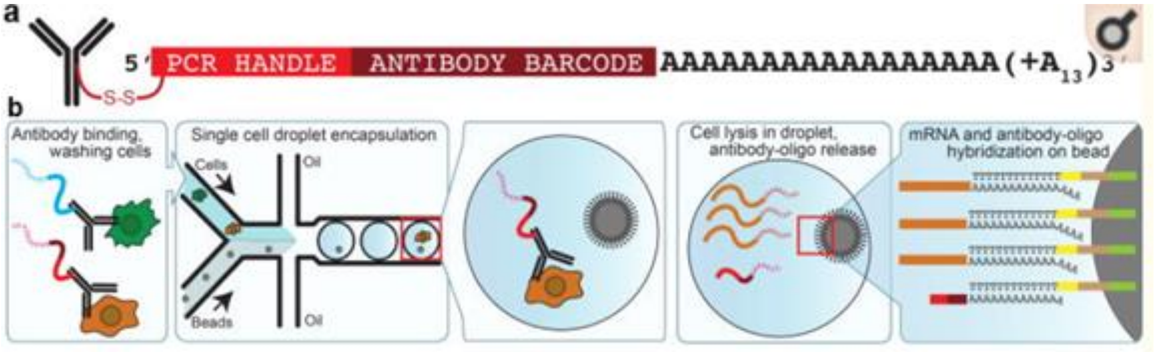

a
5′
PCR HANDLE
ANTIBODY BARCODE
AAAAAAAAAAAAAAAAA(+A13)3′
S-S
b
Antibody binding, washing cells
Single cell droplet encapsulation
Cells
Oil
Beads
Oil
Cell lysis in droplet, antibody-oligo release
mRNA and antibody-oligo hybridization on bead

Figure 1.1 The rationale of CITE-seq technology. (a) the DNA-barcoded with antibodies used in CITE-seq. (b) Schematic view of CITE-seq with Drop-seq technology. Briefly, cells are incubated with antibodies, washed, and passed through a microfluidic chip where a single cell and one bead are occasionally encapsulated in the same droplet. After cell lysis, mRNAs and antibody-oligos anneal to oligos on Drop-seq beads, linking cell barcodes with cellular transcripts and antibody-derived oligos.

Source: Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P. K., Swerdlow, H., ... & Smibert, P. (2017). Large-scale simultaneous measurement of epitopes and transcriptomes in single cells. Nature methods, 14(9), 865. 10.1038/nmeth.4380

1.2 Spatial-resolved scRNA-seq

The conventional scRNA-seq alone leaves the tissue landscape undefined as cells are dissociated from their respective tissues and suspended in solution (Longo, Guo, Ji, & Khavari, 2021), neglecting and underappreciating the spatial complexity of cells and their relations to functions (Liao, Lu, Shao, Zhu, & Fan, 2021). Furthermore, cellular organization and intercellular communication networks for novel types identified by scRNA-seq remain uncharacterized unless ligand-receptor relationships are established (Efremova, Vento-Tormo, Teichmann, & Vento-Tormo, 2020; Skelly et al., 2018; S. Wang, Karikomi, MacLean, & Nie, 2019). As cellular spatial distributions are deeply intertwined with gene expression and cell functions (Zhuang, 2021), retaining this information is pivotal to further understand the collective dynamics of biological activities.

Spatially resolved single-cell transcriptomics (sp-scRNA-seq) provides an exciting opportunity to map RNA molecules in their tissue locations, allowing for comprehensive profiling of cell heterogeneity (Liao et al., 2021).

Basically, the technologies to profile the spatial-resolved single-cell transcriptomics (or targeted genes) can be divided into two types: 1) hybridization-based (or called image-based) approaches, such as MERFISH, smFISH, and osmFISH. These technologies profile the physical location attributes of cells by single-molecule fluorescence in situ hybridization (Codeluppi et al., 2018; Miller, Bambah-Mukku, Dulac, Zhuang, & Fan, 2021). Pioneering studies in spatial genomics sought to explore fluorescence in situ hybridization (FISH) and digital imaging microscopy to allow for the detection of single RNA molecules in single cells (Femino, Fay, Fogarty, & Singer, 1998). Thereafter, various FISH probes were developed for single-cell transcript profiling, allowing for higher accuracy and sensitivity when quantifying RNA molecules at the single-molecule level such as single-molecule in situ hybridization (smFISH) (Femino et al., 1998; Kwon, 2013; Lubeck & Cai, 2012; Shah, Lubeck, Zhou, & Cai, 2016). As some smFISH methods are multiplexed by barcoding (Femino et al., 1998; Lubeck & Cai, 2012), limitations such as optical crowding and transcript length hinder marker gene targeting and cell-type mapping (Femino et al., 1998; Shah et al., 2016). Codeluppi et al. developed a non-barcoded and unamplified cyclic-ouroboros smFISH (osmFISH) method, optimized for brain tissue, to overcome the limitations of other smFISH methods (Codeluppi et al., 2018). This method demonstrates the ability to process and map large tissue areas and allows for the construction of data-driven reference atlases of human tissue. 2) Sequencing-based approaches, such as 10x Visium (see **Figure 1.2**), and Slide-seq. A joint robust dissection of scRNA-seq data with spatially resolved single-cell transcriptomics captures a detailed illustration of the concerted cell-cell interactions within the tissue architecture. These technologies provide spatially resolved, untargeted transcriptomic profiling at the pixel level, with a pixel size of 10-100μm (Larsson, Frisén, & Lundeberg, 2021). Using Visium as an example, it employs spatially barcoded mRNA-binding oligonucleotides grouped in spots (larger than one cell) on the tissue slides. The mRNA from the specialized tissue will bind to the oligos. Then, based on the collected mRNA, a cDNA library with spatial barcodes will be built, preserving the spatial information of spots. In this way, both the gene expression level and the cells/spots spatial organization in the tissue can be measured. The two types of technologies have their own advantages and disadvantages. Briefly, Imaging-based technologies can reach the single-cell resolution, but they can only profile a limited number of targeted genes/proteins; on the other hand, some sequencing-based technologies can profile the whole transcriptomes, but they cannot reach the single-cell resolution. **Figure 1.3** shows the current spatially resolved transcriptomics method summarized by Liao et al.(Liao et al., 2021). It reveals the fast development of spatial-resolved single-cell technologies in the past few years.

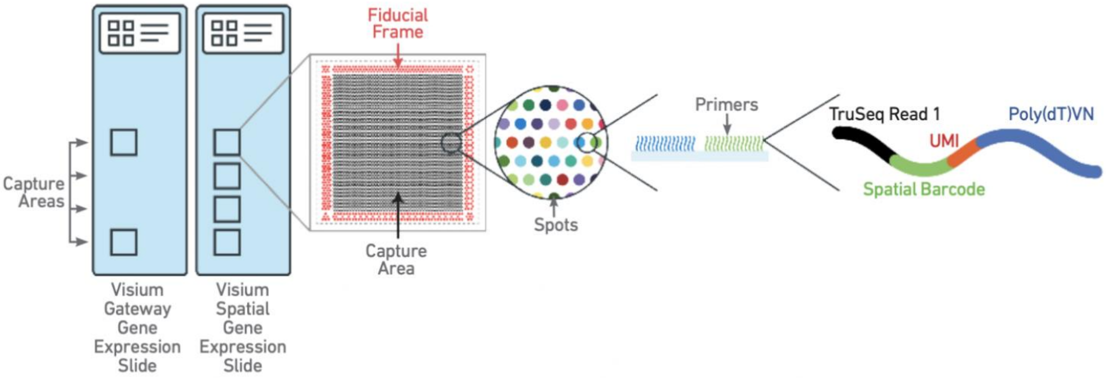


Figure 1.2 The rational of 10X Visium spatial transcriptome sequencing technology. Fresh-frozen tissue sections are placed on the slide, which is H&E stained and imaged in bright field. Depending on the tissue, an average of 1-10 cells will cover a spot. The spatial barcode assigned to the spot is incorporated during cDNA synthesis and enables gene expression data to be mapped back to its location within the tissue. Data is processed with the 10X SpaceRanger analysis software and can be visualized with the Loupe Browser software.

Source: Image provided by 10x Genomics (https://www.10xgenomics.com/).

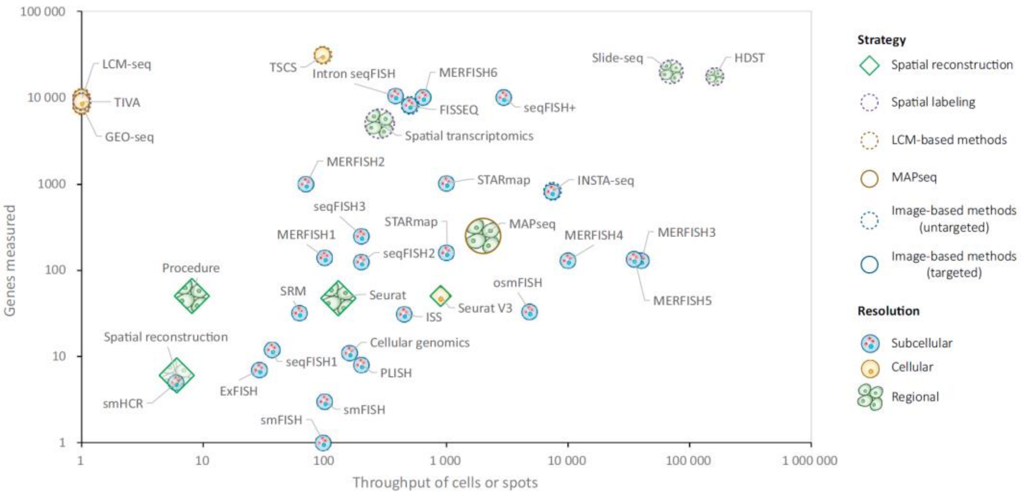


Figure 1.3 Throughput of genes and cells for each spatially resolved transcriptomics method.

Source: Liao, J., Lu, X., Shao, X., Zhu, L., & Fan, X. (2021). Uncovering an organ's molecular architecture at single-cell resolution by spatially resolved transcriptomics. Trends in Biotechnology, 39(1), 43-58. https://doi.org/10.1016/j.tibtech.2020.05.006

Such high throughput data generation, both multi-omics and spatial-resolved scRNA-seq, revealed the great demands of scalable computational methods that can take advantages of the multi-dimensional measurements to efficiently improve the downstream analyses, such as the clustering and differential expression analysis. However, to our current knowledge, there are only a few methods that are specifically developed for the multi-omics scRNA-seq data clustering, and even less methods for the spatial-resolved scRNA-seq data clustering. So, the existing computational methods do not catch up with the rapid changes in technologies and fail to fully fulfil their potential.

Table 1.1 Summary of the Real CITE-seq Datasets

Datasets	**Platform**	**Tissue**	**# of cells**	**# of total genes**	**# of ADTs**	**# of groups**
PBMC	10X	PBMC	3,762	33,538	49	16
GSE100866	10X	CBMN	1,372	33,514	10	6
BMNC	10X	BMNC	30,672	17,009	25	27
SLN111D1	10X	SLN	9,264	13,553	111	35
SLN111D2	10X	SLN	7,564	13,553	111	35
SLN208D1	10X	SLN	8,715	13,553	208	35
SLN208D2	10X	SLN	7,105	13,553	208	35

Table 1.2 Summary of the Real Single-cell Multiome ATAC Gene Expression Datasets

Datasets	**Platform**	**Tissue**	**# of cells**	**# of total genes**	**# of genes from ATAC**	**# of groups**
PBMC3k	10X	PBMC	2,585	36,601	20,010	14
PBMC10K	10X	PBMC	11,020	36,601	20,010	12
MBE18	10X	Brain	4,780	32,285	21,807	18

CHAPTER 2

CLUSTERING ANALYSIS OF SINGLE-CELL DATA

2.1 Clustering Analysis of Traditional scRNA-seq Data

Clustering analysis is an essential step in most single-cell studies and has been studied extensively. Based on the clustering results, researchers can explore the biological activities in cell type or subtype level, which could not be reached by studying bulk data (Kiselev, Andrews, & Hemberg, 2019; Kolodziejczyk, Kim, Svensson, Marioni, & Teichmann, 2015; Shapiro, Biezuner, & Linnarsson, 2013). Numerous clustering methods have been designed for the analysis of scRNA-seq data. For example, Tscan applies principal component analysis (PCA) on the scRNA-seq data and then performs the Gaussian mixture model (GMM) clustering on the low-dimensional representation (Ji & Ji, 2016). Seurat V3 employs PCA dimension reduction and performs shared nearest neighbor (SNN) clustering on the selected PCs (Butler, Hoffman, Smibert, Papalexi, & Satija, 2018) for scRNA-seq data. The SNN graph can also be used for estimating the number of clusters (K) by using the k-nearest neighbors (kNN) or Louvain (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008) algorithm. SC3 performs a consensus spectral clustering based on different types of distances between cells (Kiselev et al., 2017). The pervasive dropout events make single-cell mRNA count data to be zero-inflated and over-dispersed. A zero-inflated negative binomial (ZINB) model has been widely used to account for the large dispersion and the dropout events (Risso, Perraudeau, Gribkova, Dudoit, & Vert, 2018; Tian, Wan, Song, & Wei, 2019). Many ZINB model-based methods, including deep learning approaches, have been developed to analyze scRNA-seq count data, including ZINB-WaVE (Risso et al., 2018), DCA(Eraslan, Simon, Mircea, Mueller, & Theis, 2019), scVI (Lopez, Regier, Cole, Jordan, & Yosef, 2018) and scDeepCluster (Tian et al., 2019), to name a few. These studies show that the ZINB model can effectively characterize scRNA-seq data and improve the representation learning and clustering results.

2.2 Clustering Analysis of Multi-omics scRNA-seq Data

Clustering analysis is an essential step in most single-cell studies and has been studied extensively. Based on the clustering results, researchers can explore the biological activities in cell type or subtype level, which could not be reached by studying bulk data (Kiselev et al., 2019; Kolodziejczyk et al., 2015; Shapiro et al., 2013). Numerous clustering methods have been designed for the analysis of scRNA-seq data. For example, Tscan applies principal component analysis (PCA) on the scRNA-seq data and then performs the Gaussian mixture model (GMM) clustering on the low-dimensional representation (Ji & Ji, 2016). Seurat constructs a k-nearest neighbors (KNN) graph based on the Euclidean distance in PCA space. With the graph, it then employs the Louvain (Blondel et al., 2008)/Leiden algorithm to iteratively group cells together by optimizing modularity (Butler et al., 2018). The Louvain/Leiden algorithm has already become one of the most popular methods for scRNA-seq clustering. SC3 employs spectral clustering to obtain individual clustering results based on the distance matrices derived from the Euclidean, Pearson and Spearman metrics, respectively. It then computes a consensus matrix by summarizing the three individual clustering results. Finally, the consensus matrix is clustered using hierarchical clustering to produce final clustering results (Kiselev et al., 2017). However, these traditional single-cell clustering methods are not ready to take advantage of multi-omics data to improve clustering performance and are thus not applicable to multimodal data.

A couple of methods have emerged for the clustering analysis of CITE-seq data in the past years. Recently, we proposed a single cell deep constrained clustering framework – scDCC that can integrate ADT information into clustering analysis of scRNA-seq data by manually defined constraints (Tian, Zhang, Lin, Wei, & Hakonarson, 2021b). BREM-SC (X. Wang et al., 2020), a hierarchical Bayesian mixture model, applies two multinomial models to jointly characterize scRNA-seq and ADT data. It assumes that the proportions (relative expression levels of genes or proteins) in the multinomial models follow Dirichlet distributions, and cell-specific random effects are introduced to model the correlation between the two data sources. Although BREM-SC is one of the first proposed models for clustering analysis of CITE-seq data, it has several limitations. Firstly, it assumes that the data follow a certain specific distribution. Such parametric assumptions may not hold in all real applications. Secondly, BREM-SC does not characterize the dropout events, which is the major problem in the clustering of scRNA-seq data. Finally, BREM-SC has a scalability issue. The running time of BREM-SC becomes costly slow when analyzing thousands of cells.

Meanwhile, CiteFuse, Seurat V4, and Specter can cluster CITE-seq data by using distance-based graphs. CiteFuse (Kim, Lin, Geddes, Yang, & Yang, 2020) calculates the cell-to-cell similarity matrices of ADT and mRNA separately and then merges them by a similarity network fusion algorithm (B. Wang et al., 2014). Clustering is performed on the merged similarity matrix by using graph-based clustering algorithms such as spectral (Ng, Jordan, & Weiss, 2001) and Louvain algorithm (Blondel et al., 2008). However, similarity matrix-based clustering cannot explicitly consider the dropout events in scRNA-seq data. Hao et al. developed a weighted nearest neighbor (WNN) procedure in Seurat V4 for multi-omics data clustering (Hao et al., 2021). Briefly, the WNN procedure learns the weights of multimodal data and generates a similarity graph of cells by a weighted combination of mRNA and protein views. Van et al.(Ringeling & Canzar, 2021) proposed a landmark-based spectral clustering (LSC) method, Spector, for clustering single-cell data with linear-time scalability. LSC picks a small set of cells as the landmarks and calculates a Gaussian kernel-based similarity matrix between the rest of the cells and the landmarks, then the whole Laplacian matrix is built. Different omics require a different choice of the number of landmarks and the kernel bandwidth, and consensus clustering is used for ensembles across modalities. Compared to BREM-SC and CiteFuse, the WNN algorithm and Specter run much faster and requires less memory. However, these two methods fail to take into consideration the dropout events in the count data too.

Another line of research, which is relevant, focuses on learning a joint embedding of different modalities. Such joint embedding is expected to improve various downstream analyses, including clustering. TotalVI is a deep variational autoencoder which can capture the same latent space of different data types (Gayoso et al., 2021). With this design, totalVI can learn a joint probabilistic representation of the paired ADT and mRNA measurements from CITE-seq data that accounts for the distinct information of each modality. Similarly, for SNARE-seq or SMAGE-seq data, Cobolt (Gong, Zhou, & Purdom, 2021) and scMM (Minoura, Abe, Nam, Nishikawa, & Shimamura, 2021) employ a Multimodal Variational Autoencoder to jointly model the multiple modalities and learn a joint embedding of mRNA-seq and ATAC-seq data. However, these methods focusing on joint embedding are not designed and optimized for clustering, although we can, as a naïve solution, learn joint embeddings first, which is then followed by simple clustering using, for example, k-means. Such a divided strategy is suboptimal for clustering, as shown in our experiments later.

As we mentioned in the last three paragraphs, many existing methods fail to consider the dropout events in the single-cell data during the learning of embedding and/or clustering. However, the pervasive dropout events make single-cell count data to be zero-inflated and over-dispersed. To better characterize single-cell mRNA count data, a zero-inflated negative binomial (ZINB) model has been widely used to account for the large dispersion and the dropout events (Risso et al., 2018; Tian et al., 2019). Many ZINB model-based methods, including deep learning approaches, have been developed to analyze scRNA-seq count data, including ZINB-WaVE (Risso et al., 2018), DCA (Eraslan et al., 2019), scVI (Lopez et al., 2018), and scDeepCluster (Tian et al., 2019), to name a few. These studies show that the ZINB model can effectively characterize scRNA-seq data and improve the representation learning and clustering results.

2.3 Clustering Analysis of Spatial-resolved scRNA-seq Data

Clustering analysis is an essential step in most single-cell studies and has been studied extensively. Based on the clustering results, researchers can explore the biological activities in cell type or subtype level, which could not be reached by studying bulk data (Kiselev et al., 2019; Kolodziejczyk et al., 2015; Shapiro et al., 2013). It has been demonstrated that some cell types, such as the neurons, have high spatial dependency and heterogeneity (Codeluppi et al., 2018). Specifically, tissues are an ensemble of cell types that interactively give rise to a specific function. It has been shown that endothelial cells in the brain are located under certain spatial patterns (Stoltzfus et al., 2020; Xia, Fan, Emanuel, Hao, & Zhuang, 2019). Furthermore, within cells of the same type, high spatial self-affinity was measured in ependymal cells and spatial self-evasion was observed in inhibitory neurons such as microglia and astrocytes (Codeluppi et al., 2018). Cell neighbors identified by spatio-temporal organization within tissues in complex organs (e.g., the brain) provides important context to make inferences regarding cell interactions and behaviors. As such, highly accurate and sensitive mapping of tissue sections is important to reveal spatially dependent cells and can be used to understand the convolutions of cell heterogeneity. The set of neighboring cells from the spatial transcriptomics studies may provide valuable information for cell-type annotation. In other cases, such knowledge can lead to the identification of new cell types based on their neighborhood profiles. However, this entails that computational resources to analyze transcriptomic data are appropriately equipped with mechanisms to integrate the spatial features. Nevertheless, traditional methods, such as Seurat (Butler et al., 2018) and SC3 (Kiselev et al., 2017), cannot utilize valuable spatial information in the clustering analysis.

Some tools have been developed for spatially transcriptomic data. Giotto is a computational method specifically designed for spatial transcriptomic data analysis that performs cell-type enrichment analysis, spatially coherent gene detection, cell neighborhood, and interaction analyses, and spatial pattern recognition (Dries et al., 2021). Unlike other computational methods that are geared towards scRNA-seq analysis but utilize spatial information to identify cell types (Stuart et al., 2019), marker genes (Svensson, Teichmann, & Stegle, 2018), or domain patterns (Zhu, Shah, Dries, Cai, & Yuan, 2018), Giotto is purely centered towards spatial data analysis but is capable of integrating scRNA-seq data to enhance spatial-cell type enrichment analysis. In the clustering analysis, Giotto employs graphic clustering algorithms, such as Louvain (Blondel et al., 2008), to find cell communities. BayesSpace is a Bayesian statistical method that enhances spatial transcriptomic resolution and performs clustering analysis by modeling dimensionally reduced representation of the single-cell count matrix and grouping neighboring spots to the same cluster via spatial prior (Zhao et al., 2021). BayesSpace draws a distinction in use of a t-distributed error model to identify spatial clusters and employs a Markov chain Monte Carlo to estimate model parameters. However, BayesSpace has a limited scope of application as it is majorly designed for decomposing the data with low resolution from the sequencing-based technologies, such as the 10x Visium. Besides, some other methods, such as SpaGCN (Hu et al., 2021) and stLearn (Pham et al., 2020), employ deep neural networks, such as CNN and GCN, to analyze the sp-scRNA-seq data. These tools can also integrate the information from the H&E images to enhance the cell clustering.

It is widely demonstrated that in many tissues, especially in the brain, many marker genes have exhibited strong spatial expression dependencies (Guillozet-Bongaarts et al., 2014; Maynard et al., 2021; Zeisel et al., 2015). Therefore, the information from the markers can be used as the prior knowledge to guide the sp-scRNA-seq analyses, especially for the clustering analysis. However, none of the methods mentioned in the last paragraph can incorporate the marker gene information in the clustering process.

CHAPTER 3

MULTI-OMICS SCRNA-SEQ MODEL – SCMDC

3.1 Introduction

In this chapter, we introduce a multimodal deep learning model, Single Cell Multimodal Deep Clustering (scMDC), for the clustering analysis of multimodal single-cell data. The network architecture of scMDC is shown in **Figure 3.1**. scMDC employs a multimodal autoencoder (Simidjievski et al., 2019), which applies one encoder for the concatenated data from different modalities and two decoders to separately decode the data from each modal. Following scDeepCluster (Tian et al., 2019), we apply ZINB loss as the reconstruction loss. The bottleneck layer is used for a deep K-means clustering (Xie, Girshick, & Farhadi, 2016). To further improve latent feature learning, we introduce a Kullback-Leibler divergence-based loss (KL loss), which attracts similar cells and separates dissimilar cells (L. Chen, Wang, Zhai, & Deng, 2020). The whole model, including the autoencoder, the KL-loss, and the deep K-means clustering, are optimized simultaneously. scMDC is an end-to-end multimodal deep learning clustering method for modeling different multi-omics data. Taking advantage of graphics processing units (GPU), scMDC is very efficient in the analysis of large datasets. In addition, by employing a conditional autoencoder framework, scMDC can correct batch effect when analyzing multi-batch data. To our knowledge, scMDC is the first end-to-end deep clustering method that can both integrate multimodal data and remove the batch effect for different types of multimodal data. The superior performance of scMDC is observed from the extensive experiments on both CITE-seq and SMAGE-seq data. After clustering, for a given cluster, we also detect the markers (gene or ADT) by transplanting an ACE model (Lu, Yu, Bonora, & Noble, 2021) to scMDC and conduct a gene set enrichment analysis based on the gene ranks from ACE. The meaningful results of these downstream analyses further support the superior clustering performance of scMDC. We conclude that scMDC is a promising tool for clustering multimodal single-cell data.

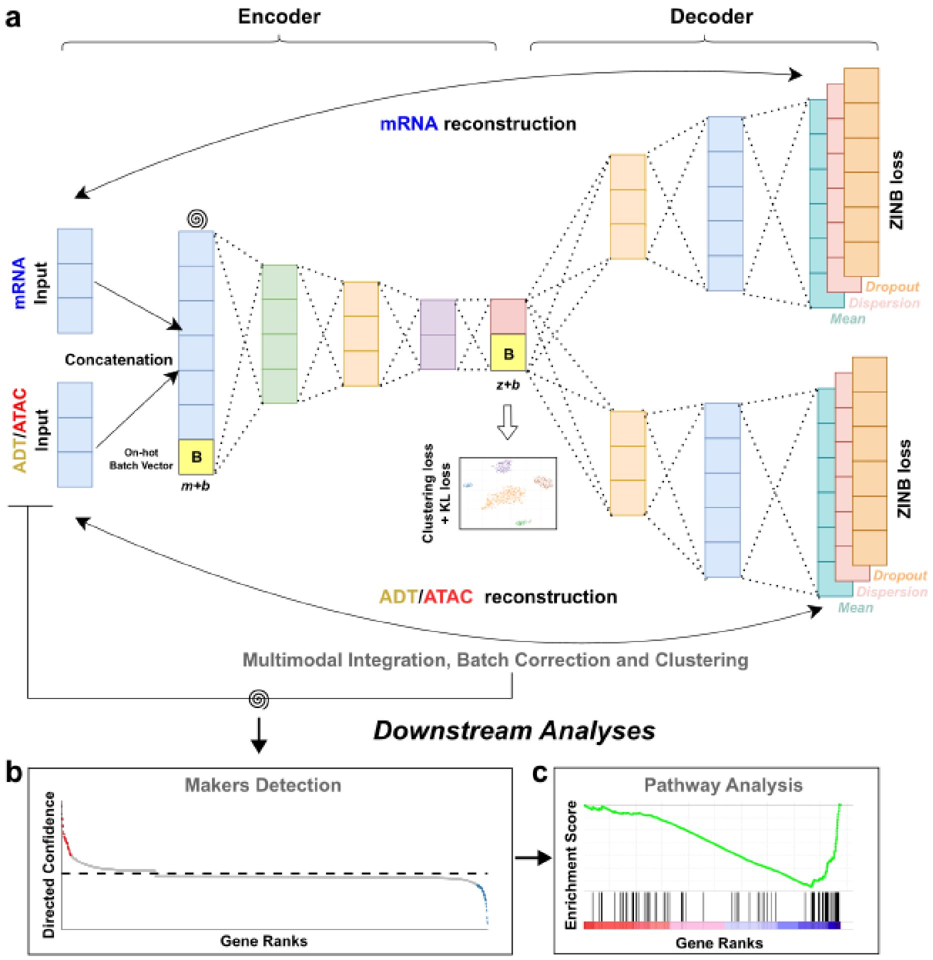


Figure 3.1 The architecture of scMDC. (a) scMDC has one encoder for the concatenated data and two decoders for each modal in the multimodal data. It can be used for clustering CITE-seq data and 10x Single-Cell Multiome ATAC + Gene Expression (SMAGE-seq) data. The spiral symbols indicate the artificial noises added to the data. For multi-batch datasets, scMDC will work in a conditional autoencoder manner. A one-hot batch vector B (in dimension *b*) will be concatenated to the input feature of the encoder (with raw feature dimension, *m*) and the decoders (with latent feature dimension, *z*). This is designed for batch effect correction. scMDC learns a latent representation Z (in dimension *z*) of data on which different modalities are integrated. A deep K-means algorithm and a KLD loss are implemented on Z. (b) Based on the clustering results, scMDC employs an ACE model to detect markers in different clusters. (c) Then, pathway analyses can be conducted based on the gene ranks calculated by ACE.

3.2 Experiments and Results

3.2.1 Real CITE-seq data evaluation

We first evaluate the clustering performance of scMDC on CITE-seq datasets in comparison with nine competing methods. The competing methods include the models designed for multimodal data clustering (BREM-SC, CiteFuse, and SeuratV4), the models developed for learning an embedding for single or multimodal data (SCVIS and TotalVI), and some general clustering tools for single-cell data (SC3 and Tscan). We test these tools on seven single-batch CITE-seq datasets and two multi-batch CITE-seq datasets. Of these ten methods under comparison, only scMDC, Seurat, and TotalVI can correct batch effect before clustering. We hypothesize that scMDC can boost the clustering performance in all the CITE-seq real datasets. **Figure 3.2** shows the performance (AMI, NMI, and ARI) of all the methods for different datasets. Overall, the multimodal methods have shown clear advantage over the single-modal methods (IDEC, SCVIS, SC3, Tscan, and Kmeans + PCA). As shown in **Figure 3.2a**, scMDC has demonstrated superior performance over competing methods across two metrics for most single-batch datasets except the BMNC dataset, in which Seurat has comparable performance. For the two multi-batch datasets, scMDC outperforms all the competing methods (see **Figure 3.2b**); TotalVI and Seurat are inferior to scMDC but outperform the other competing methods, thanks to their capability of correcting batch effect. The differences between the performance of scMDC and the competing methods are summarized in **Figure 3.2c**. A positive difference means a higher performance in scMDC than the competing methods. We find that scMDC has a steady advantage over all the competing methods in multiple datasets. We then rank all competing methods for each dataset based on their performance metrics. **Figure 3.2d** shows the averaged rank of each method for the nine datasets. We can see that scMDC constantly ranks number 1 in all datasets for all three metrics. In contrast, the second-best methods, Seurat for AMI and NMI and Specter for ARI, have an averaged rank of 3. Using one-sided paired t-tests on the clustering metrics (AMI, NMI, and ARI), we confirm that the improvements of scMDC over competing methods are all significant (see **Appendix Table D.1**). In summary, our results on multiple real datasets reveal that scMDC has stable and robust clustering performance on the CITE-seq datasets.

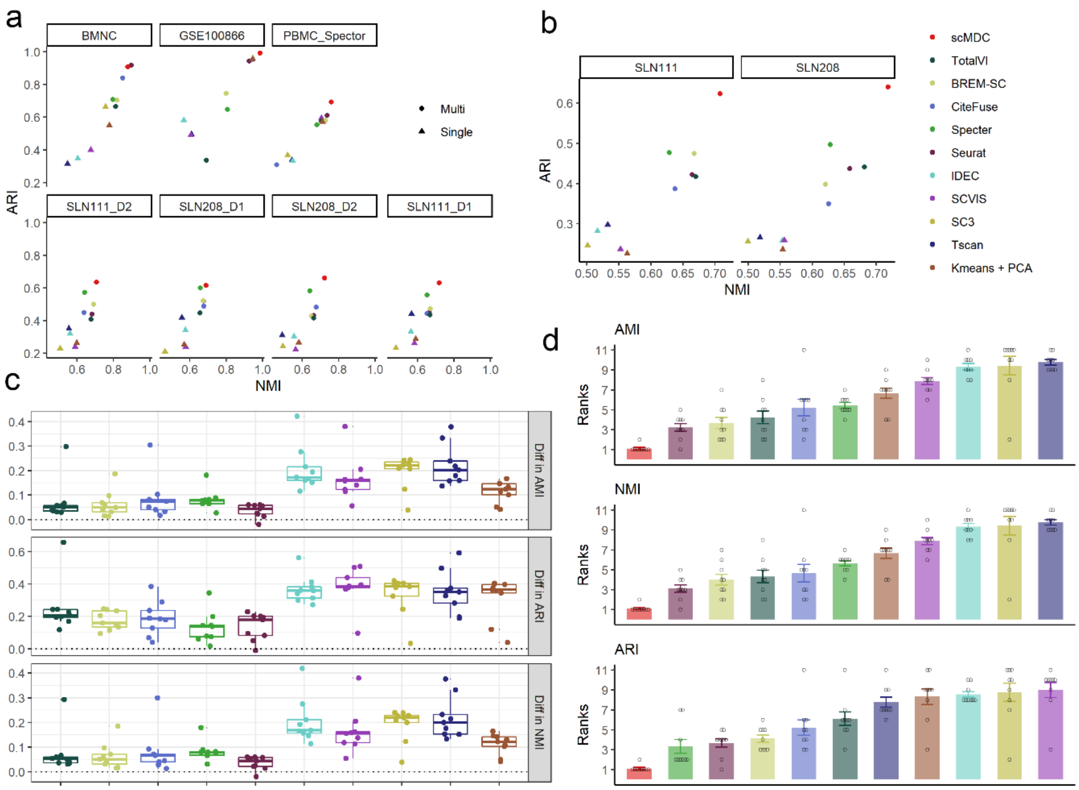


Figure 3.2 Clustering performance of scMDC and the competing methods on different CITE-seq datasets. All the methods are tested on (a) seven one-batch datasets (n=7) and (b) two two-batch datasets (n=2). In panel a and b, clustering performance is illustrated in a two-dimensional manner with ARI as the Y axis and NMI as the X axis. Circles stand for the results of the multi-omics methods and triangles stand for the results of the single-omics methods. (c) The differences between the performance of scMDC and the competing methods are shown in boxplots (n=9). Each boxplot shows the minimum, first quartile (Q1), median, third quartile (Q3), and maximum of data. The minimum and maximum are Q1 -1.5*IQR and Q1 + 1.5*IQR, respectively. Each data point (a difference of performance in a dataset) is shown by a dot. (d) We also summarized the performance of each method by showing the averaged ranks (n=9). Each data point (a rank of a method in a dataset) is shown by a dot and the standard errors are shown by the error bars. In panel c and d, clustering performance is evaluated by AMI, NMI, and ARI.

3.2.2 Real SMAGE-seq data evaluation

We then test the clustering performance of scMDC on the SMAGE-seq data. Here we compare scMDC with four competing methods: Cobolt, scMM, SeuratV4, and K-means + PCA. Cobolt and scMM are designed for multi-omics data embedding learning. SeuratV4 is developed for CITE-seq data but here we apply the WNN algorithm to the SMAGE-seq data. We test these methods on three real SMAGE-seq datasets from 10X genomics, including two PBMC datasets and one embryonic mouse brain dataset. We also conducted a multi-batch experiment by combining two PMBC datasets (denoted as PBMC13K). For scATAC-seq data, we use a cell-to-gene matrix as input for scMDC, scMM, Seurat, and Kmeans. This matrix is built by mapping ATAC reads onto the gene regions (See method for details). Cobolt uses the peak count matrix as the input. **Figure 3.3** shows the clustering performance of scMDC and the competing methods in (a) single-batch datasets and (b) multi-batch datasets. We find that scMDC has superior performance in both single- and multi-batch datasets from all the metrics (NMI and ARI). Cobolt is the second-best method in the tests and has a comparable performance with scMDC on the E18 dataset in NMI, but its performance is inferior to that of scMDC in other datasets. **Figure 3.3c** summarizes the differences of clustering performance between scMDC and the competing methods. We find that the median differences are around 0.1 in AMI and NMI, and around 0.3 in ARI for all the competing methods, which illustrates the superiority of scMDC. We then rank all competing methods for each dataset based on their performance metrics. **Figure 3.3d** shows the averaged rank of each method for the four datasets. We can see that scMDC ranks best in all three metrics, while Cobolt is the second-best for AMI and ARI, and Seurat is the second-best for ARI. Using one-sided paired t-tests done on the three raw performance metrics, we confirm that the improvements of scMDC over competing methods are all significant (See **Appendix Table D.2**).

Taking the results from CITE-seq and SMAGE-seq experiments together, we conclude that scMDC is a general and promising clustering model for various single-cell multimodal data.

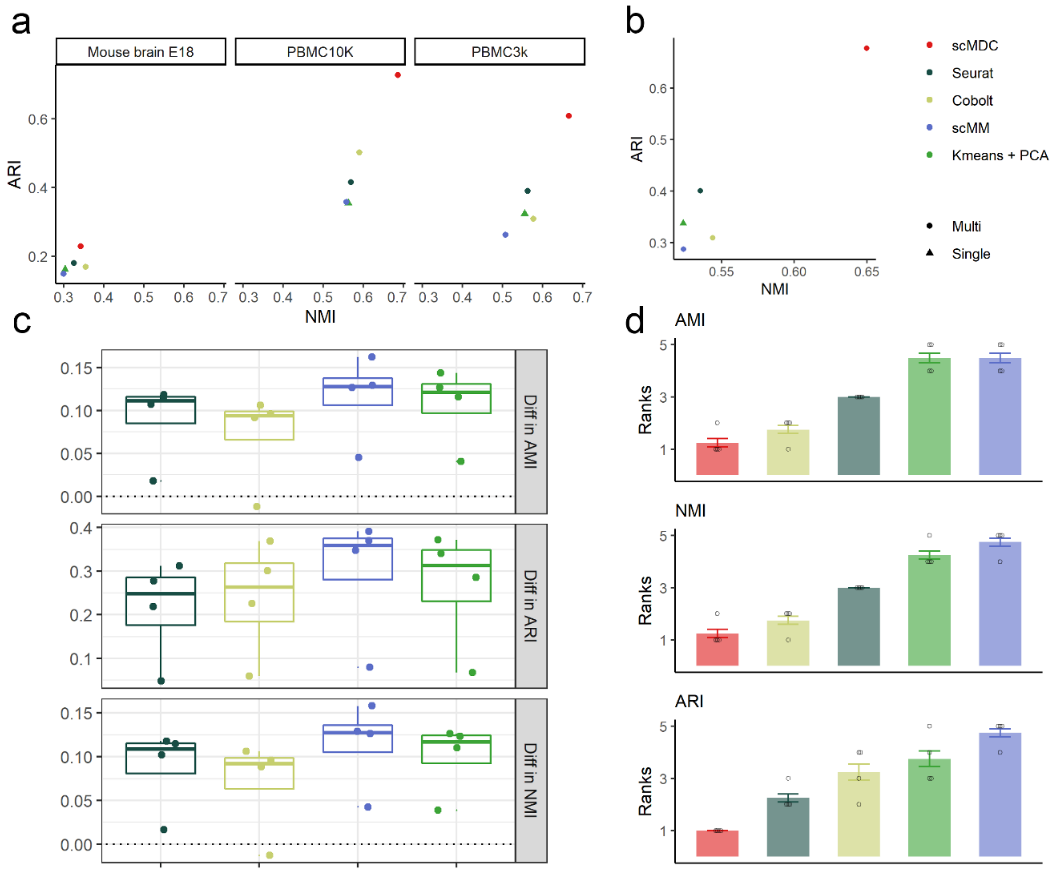


Figure 3.3 Clustering performance of scMDC and the competing methods on different SMAGE-seq datasets. All the methods are tested on (a) three one-batch datasets (n=3) and (b) one two-batch dataset (n=1). In panel a and b, clustering performance is illustrated in a two-dimensional manner with ARI as the Y axis and NMI as the X axis. Circles stand for the results of the multi-omics methods and triangles stand for the results of the single-omics methods. (c) The differences between the performance of scMDC and the competing methods are shown in boxplots (n=4). Each boxplot shows the minimum, first quartile (Q1), median, third quartile (Q3), and maximum of data. The minimum and maximum are Q1 - 1.5 * IQR and Q1 + 1.5 * IQR, respectively. Each data point (a difference of performance in a dataset) is shown by a dot. (d) We also summarized the performance of each method by showing the averaged ranks (n=4). Each data point (a rank of a method in a dataset) is shown by a dot and the standard errors are shown by the error bars. In panel c and d, clustering performance is evaluated by AMI, NMI, and ARI.

3.2.3 Simulation experiments

To test the robustness of scMDC under different scenarios, we conducted two simulation experiments with various clustering signals and dropout rates. We generate all the simulation datasets using SymSim package (v0.0.0.9) in R. **Figures 3.4 a**, **b**, and **c** show the performance of scMDC and the competing methods on the simulated CITE-seq data with low, medium, and high clustering signals, respectively. scMDC has demonstrated superior performance across all levels of clustering signals, especially in terms of AMI and NMI. TotalVI has comparable performance with scMDC in ARI, but it is outperformed by scMDC in other metrics. Besides, when the clustering signal is low, scMDC shows a greater advantage over other methods, revealing its capability to handle datasets with low signal-to-noise ratios. **Figures 3.4 d**, **e**, and **f** show the clustering results of all the methods with low, medium, and high dropout rates, respectively. We can see that scMDC yields the optimal performance under various dropout rates, followed by TotalVI. We also observe that the higher the dropout rate, the larger improvement scMDC brings, in comparison with its competing methods. Such a result is compelling because most real single-cell datasets exhibit high dropout rates. The robust performance under high dropout events makes scMDC to be a superior clustering method. This result also consolidates our statement that scMDC is a better tool to cluster the datasets with low signal-to-noise ratios than the competing methods. For multi-batch data, we compare scMDC with TotalVI and Seurat, the only two competing methods that can correct batch effect. Medium dropout rate and clustering signal are used for simulating the multi-batch dataset. scMDC outperforms the two competing methods in all three metrics (see **Figure 3.4g**). The differences between the performance of scMDC and each competing method are summarized in **Figure 4h**. Although the distribution of differences varies across different methods, all the medians of differences are greater than zero indicating a consistent superiority of scMDC over all the competing methods. Similarly, we rank all methods in analysis of these simulated datasets. scMDC and TotalVI constantly rank No. 1 and No. 2, respectively (see **Figure 3.4i**). Like the results in the real datasets, multi-omics methods have better overall performance than the single-source methods. Using one-sided paired t-tests done on the three raw performance metrics, we confirm that the improvements of scMDC over competing methods are all significant (see **Appendix Table D.3**). These simulation results demonstrate that scMDC has robust clustering performance under various scenarios.

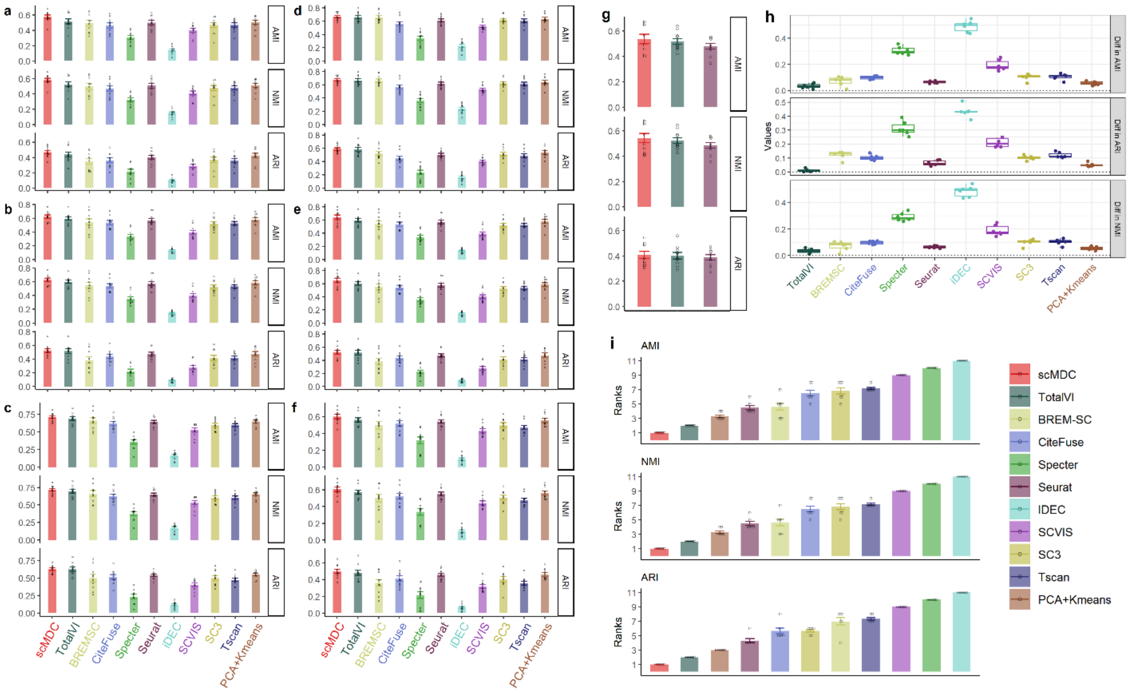


Figure 3.4 Clustering performance of scMDC and the competing methods on the simulation datasets. The first simulation experiment is to test the clustering performance of scMDC with (a) low, (b) medium, and (c) high clustering signals. The second simulation experiment is to test the clustering performance of scMDC with (d) low, (e) medium, and (f) high dropout rates. (g) Since scMDC, Seurat, and TotalVI can correct the batch effect, we also test their clustering performance on a multi-batch simulation dataset. In panel a-f, bars stand for the mean values, dots stand for the data points, and error bars stand for the standard errors. We generate ten replicates for each experimental setting (n=10). (h) The differences between the averaged performance of scMDC and the competing methods over all simulation datasets are shown in boxplots (n=6). Each boxplot shows the minimum, first quartile (Q1), median, third quartile (Q3), and maximum of data. The minimum and maximum are Q1 - 1.5 * IQR and Q1 + 1.5 * IQR, respectively. Each data point (a difference of averaged performance in a dataset) is shown by a dot. (i) We also summarized the performance of each method by showing the averaged ranks (n=7). Each data point (an average rank of a method in a setting) is shown by a dot and the standard errors are shown by the error bars. In all panels, the clustering performance is evaluated by AMI, NMI, and ARI.

3.2.4 Latent representations of real data

Figure 3.5 shows the t-SNE plots of the embedding of scMDC (a) and four competing methods, IDEC (b), scVIS (c), TotalVI (d), and Seurat (e), on the BMNC dataset. We also show the expression pattern of three marker genes in the t-SNE plots. They are *LYZ* (the first column) for CD1*4* monocyte cells, *CD8A* (the second column) for CD8 cells, and *NKG7* (the third column) for NK cells. True labels (cell types) are shown in the fourth column. We find that scMDC can divide most cell types in the latent space. In contrast, scVIS, totalVI, and Seurat fail to separate many cell types, including large cell types, such as CD14 monocyte and CD4 memory cells, which are connected or mixed with other cell types in the latent spaces. IDEC divides large cell types into many small clusters. Many of them are mixed with other cell types. It is noted that scMDC fails to divide some sub-cell types, such as CD8 effect 1, CD8 effect 2, CD8 memory 1, and CD8 memory 2, on the latent space. This problem is also observed on the t-SNE plots of other methods. In the latent space of scMDC, the marker genes are only expressed in some isolated clusters. However, in the latent space of other methods, the marker genes are either expressed in multiple clusters or in a part of a huge cluster. These are all unsatisfactory expression patterns. Similar results are observed in the expression pattern of ADT markers (see **Appendix Figure A.1**). We then build t-SNE plots of the embeddings of a multi-batch dataset SLN111 with two batches of data (see **Figure 3.6**). This dataset contains 35 cell types including some large ones (>1000 cells, such as CD4 and CD8 T cells) and tiny ones (<100 cells, such as erythrocytes and plasmacytoid dendritic cells - pDCs). An ideal model should be capable of 1) dividing different cell types on the latent space, and 2) removing batch effect and mixing the cells from different batches on the latent space. In other words, biological variations should be captured while technical variations are omitted during the embedding learning. **Figure 3.6** shows the latent representations of (a) scMDC and four competing methods including (b) IDEC, (c) scVIS, (d) TotalVI, and (e) Seurat. We find that scMDC can separate most cell types in the latent space. In addition, it mixes the cells from two batches in most clusters. IDEC can separate the large cell types but fails to divide the small cell types. scVIS, TotalVI, and Seurat show inferior performance in dividing different cell types in the latent space. Like scMDC, TotalVI and Seurat also have satisfactory performance on batch effect correction. scVIS and IDEC cannot address batch effect, so the cells from two batches are totally separated on the latent space. In summary, scMDC is the only method that has superior performance on both cell type partition and batch effect removal. Similar results can be found on the t-SNE plots of the multi-batch SMAGE-seq dataset (PBMC13K, see **Appendix Figure A.2**).

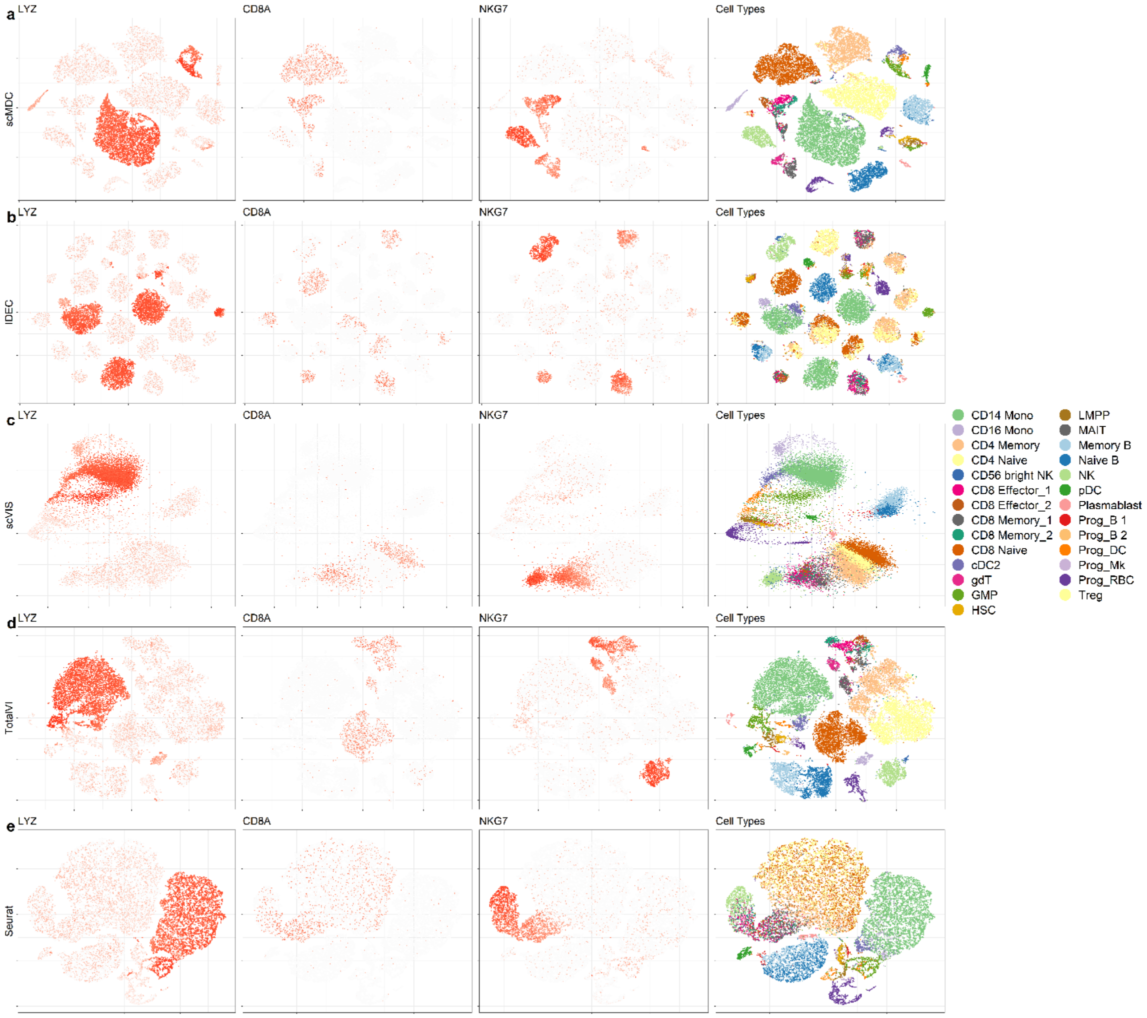


Figure 3.5 Low-dimension representation of scMDC and the competing methods on the BMNC dataset. The t-SNE plots of the embeddings from (a) scMDC and four competing methods including (b) IDEC, (c) scVIS, (d) TotalVI, and (e) Seurat are shown in different rows. The first three columns show the expression pattern of genes *LYZ*, *CD8A*, and *NKG7*. The last column shows the true labels (cell types) on the latent space of each method.

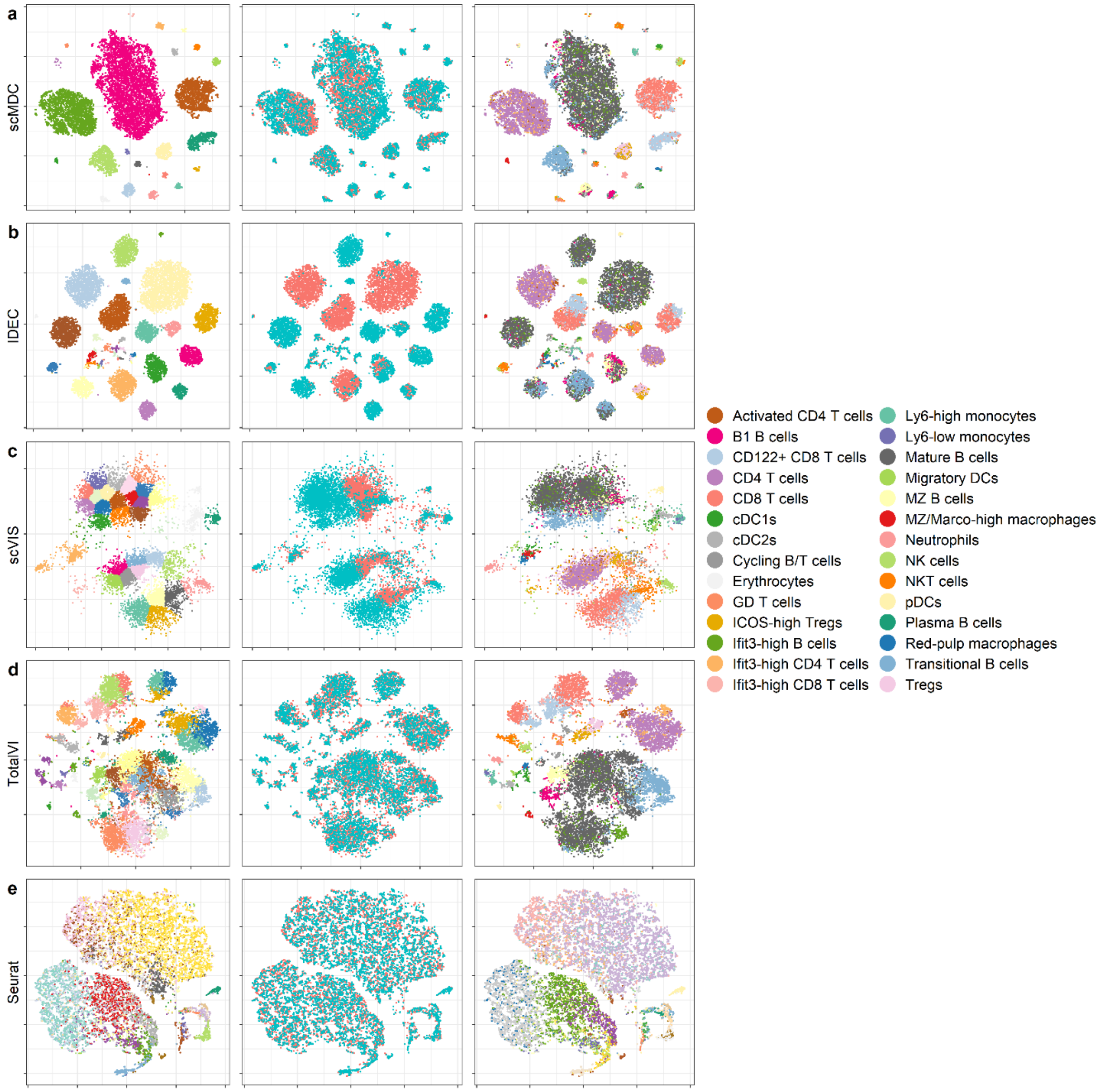


Figure 3.6 Low-dimension representation of scMDC and the competing methods on the SLN111 dataset. The t-SNE plots of the embeddings from (a) scMDC and four competing methods including (b) IDEC, (c) scVIS, (d) TotalVI, and (e) Seurat are shown in different rows. The three columns show the predicted labels, the batch IDs, and the true labels on the latent space of each method.

3.2.5 The advantage of using multimodal data

As described in the introduction, different omics of data provide different and complementary information for cell clustering and cell typing. Therefore, using multi-omics data in clustering should be able to achieve better performance than using single-source data. In this experiment, we conducted two tests. In the first test, we compare the performance of scMDC with two variant models: a sub-model of scMDC with only mRNA input and reconstruction loss (named scMDC-RNA) and another sub-model of scMDC with only ADT/ATAC input and reconstruction loss (named scMDC-ADT/scMDC-ATAC). We also compare scMDC to a model with concatenated mRNA and ADT data as input but with only one reconstruction loss (named scMDC-Concat). **Figures 3.7 a** and **b** show the performance of scMDC and three variant models in CITE-seq and SMAGE-seq data, respectively. We find that scMDC outperforms the variant models in all the datasets. For CITE-seq data, scMDC-ADT has the second-best performance in all datasets. This is consistent with our expectation because most ADTs are strong markers for identifying some cell types. On the other hand, scMDC-ATAC has inferior performance in two SMAGE-seq datasets. The differences between the performance of scMDC and each variant model are summarized in **Figure 3.7c**. We find a stable advantage of scMDC over all the variant models. Using one-sided paired t-test, we find that scMDC significantly outperforms most variant models for both CITE-seq and SMAGE-seq data (see **Appendix Table D.4**). The only exception is the scMDC-ATAC model (P-value = 0.07), because of the low sample size of SMAGE-seq data (n=4). Considering that the sub-models of scMDC are not optimized for clustering scRNA-seq data, we compare scMDC with scDeepCluster, a state-of-art tool for clustering scRNA-seq data. It is noted that scMDC uses multi-omics data as input (either mRNA + ADT or mRNA + ATAC), while scDeepCluster only uses mRNA-seq data as input. We find that scMDC outperforms scDeepCluster in all datasets (see **Figures 3.7d** and **e**), indicating that scMDC can integrate the information from multimodal data to boost clustering performance. We also build the t-SNE plots of the embeddings from scMDC and three variant models (see **Appendix Figure A.3**). Consolidating our expectations in the introduction, scMDC-RNA correctly separates some tiny cell types but falsely combines some large cell types. In constrast, scMDC-ADT separates most large cell types but fails to detect some small cell types. scMDC-Concat exhibits similar performance as scMDC-RNA, which suggests a predominant role of mRNA data in the concatenate input. The t-SNE plots of SMAGE-seq data (PBMC13K) from scMDC and three variant models are shown in **Appendix Figure A.4**. scMDC also outperforms the variant models in cell type partition on the latent space. In addition, we compare the single-modal scMDC (scMDC-RNA and scMDC-ADT/scMDC-ATAC) to other single-modal methods (see **Appendix Figures B.1-8**). We find that in most datasets, the single-modal scMDC models also have the best or close-to-best performance. Based on these single-modal methods, the multimodal scMDC further boosts the clustering performance by integrating the information from two omics of data.

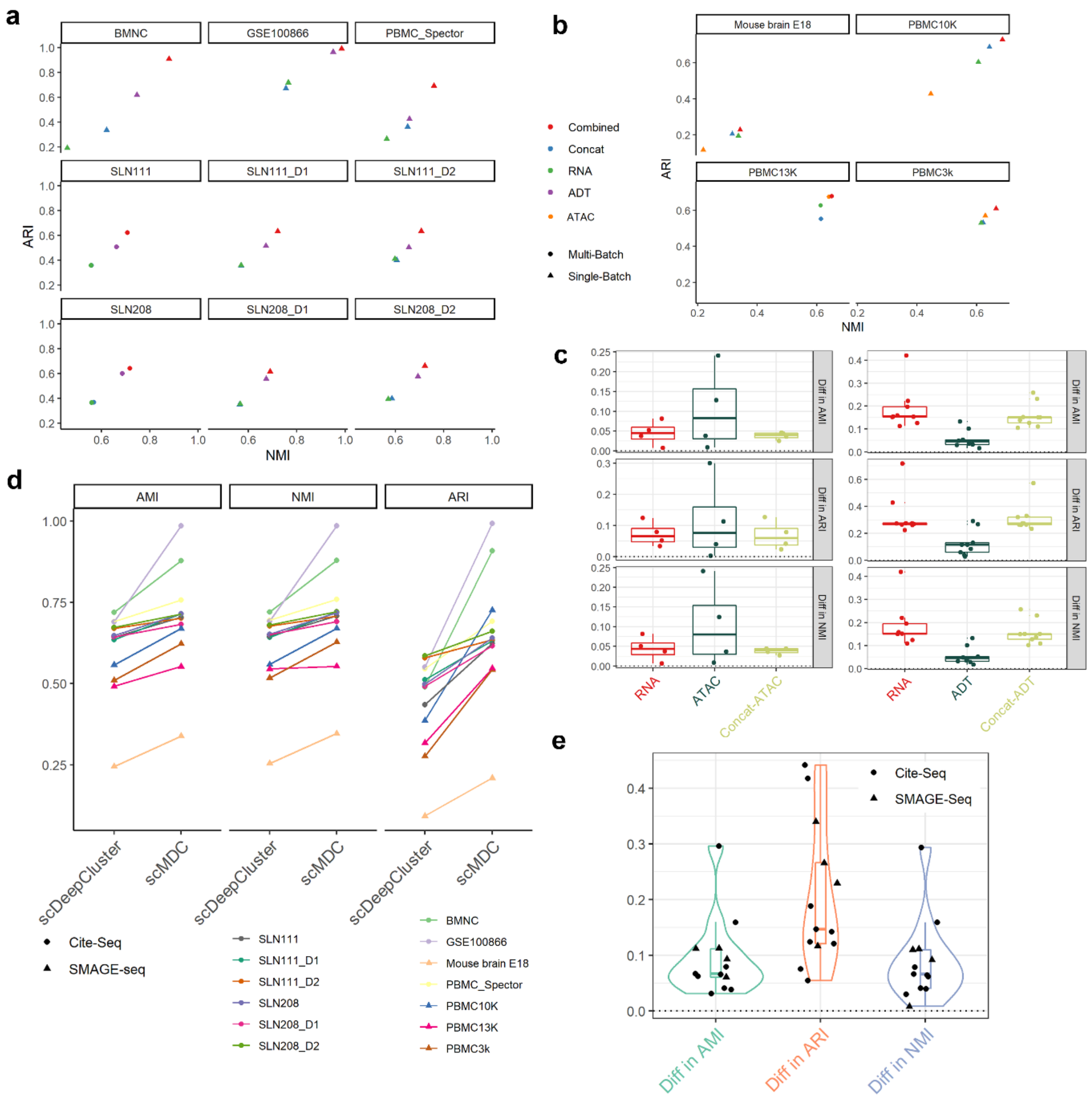


Figure 3.7 Clustering performance of scMDC and the variant models on the multimodal datasets. (a) scMDC, scMDC-RNA, scMDC-ADT, and scMDC-Concat are tested on the CITE-seq data (n=9) and (b) scMDC, scMDC-RNA, scMDC-ATAC, and scMDC-Concat are tested on the SMAGE-seq data (n=4). In panel a and b, clustering performance is illustrated in a two-dimensional manner with ARI as the Y axis and NMI as the X axis. Circles stand for the results of multi-batch datasets and triangles stand for the results of single-batch dataset. (c) The differences between the performance of scMDC and the competing methods in CITE-seq data (left, n=9) and SMAGE-seq data (right, n=4) are shown in boxplots. (d) The comparisons between scMDC and scDeepCluster are shown in dots (n=13). The paired performance for each dataset from two methods are connected by lines. (e) The differences between the performance of scMDC and the scDeepCluster are shown in boxplots and violin plots (n=13).

3.2.6 Downstream analysis

Based on the clustering results, we perform two popular downstream analyses, differential expression (DE) analysis and gene set enrichment analysis (GSEA). We employ the algorithm from ACE (Lu et al., 2021) which ranks genes based on the confidence of them to be assigned to this cluster. The DE analysis can be performed between two clusters or between one cluster and the rest of the clusters. Then, we calculate the log-fold change of each gene to get the directions of differential expression (namely upregulation or downregulation) based on the normalized mRNA counts. With gene ranks and directions, we perform GSEA to find the enriched pathways in the target clusters. Here, we show the results of the BMNC dataset (**Figure 3.8**). We conduct DE and GSEA for the four largest clusters in the BMNC data. All comparisons are performed between the target cluster and the rest of the clusters. **Figure 3.8a** shows the DE genes for CD14 monocyte, CD4 memory T cells, CD4 naive T cells, and CD8 naive T cells. We find many proven marker genes for each cell type. For example, *LYZ*, *CST3*, *HLA-DRA*, *CD74*, and *CD14* have been shown highly expressed in the monocyte cells (Schlachetzki et al., 2018). *CD27* and *CCR7* are the marker gene for naive cells (Caccamo, Joosten, Ottenhoff, & Dieli, 2018). They are in the top ranks in both CD4 naive and CD8 naive clusters. *IL7R* and *S100A4* have been demonstrated to be highly expressed in memory T cells (Harding et al., 2018). **Figure 3.8b** shows the GSEA results of the Hallmark pathways based on the DE analysis. Hierarchical clustering is performed on both pathways and cell clusters. We find that two naive cell types are clustered together and have many common enriched pathways. The MYC targets are enriched in CD4 naive, CD4 memory, and CD8 naive clusters. Their important functions in CD4 and CD8 T cells have been demonstrated by Marchingo et al.(Marchingo, Sinclair, Howden, & Cantrell, 2020) The complement system has the highest enrichment score in CD14 monocytes. It is an essential pathway for the phagocytosis of mesenchymal stromal cells by monocytes (Gavin et al., 2019). The hypoxia pathway is enriched in the CD4 memory T cells. It has been widely shown that hypoxia has a significant influence on the metabolism and differentiation of memory CD4 T cells(Cho et al., 2019; Dimeloe et al., 2016; Hasan, Chiu, Shaw, Wang, & Yee, 2021). IL2 signaling is enriched in CD4 memory T cells. Its dynamic roles in CD4 T cells have been demonstrated in the previous studies(Jones, Read, & Oestreich, 2020; Ross & Cantrell, 2018). The enrichment plots of the significant Hallmark pathways are shown in **Appendix Figures C.1-4**. These downstream analyses further consolidate the correctness of the clustering results of scMDC.

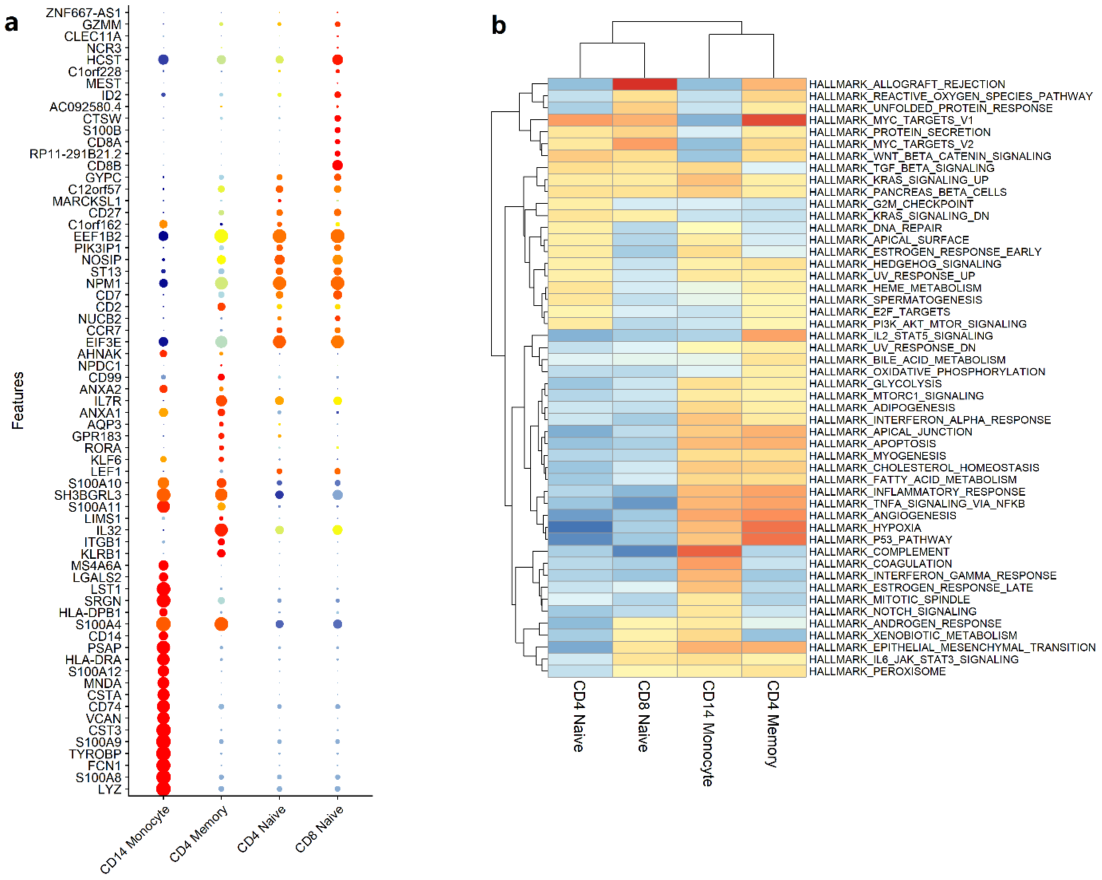


Figure 3.8 Downstream analyses of scMDC in BMNC dataset. (a) Differential expression analysis and (b) Hallmark gene set enrichment analysis are conducted for four large cell clusters in the BMNC dataset based on the clustering result of scMDC. In panel a, dot size shows the percentage of a gene expressed in a cell type and colors show the average expression of a gene in a cell type with blue as low and red as high.

3.2.7 Hyperparameter tuning and time complexity

scMDC has two key hyperparameters φ (Phi) and γ (Gamma) that control the KL loss and clustering loss, respectively. **Figures 3.9a** and **b** show the clustering performance of scMDC on both CITE-seq and SMAGE-seq datasets with various φ and γ, respectively. We find that when φ is lower than 0.01 and γ is lower than

10, scMDC is insensitive to these parameters. When φ goes beyond 0.01 and γ goes beyond 10, scMDC's performance drops dramatically. It is noted that the clustering loss has a clear contribution to the performance of most datasets (P<0.05 from one-sided paired t-test between $\gamma = 0.1$ and $\gamma = 0$). On the other hand, the KL loss contributes slightly to the performance of some CITE-seq data but boosts the performance of SMAGE-seq data, especially in ARI. The statistical tests of the hyperparameter tuning results are listed in **Appendix Table D.5**.

To test the running time of scMDC, we simulate datasets with cell numbers ranging from 1000 to 100,000. **Figure 3.9c** shows the running time of scMDC with ascending cell numbers. We find a linear relationship between the cell numbers and the running time of scMDC. When the cell number is ten thousand, scMDC only needs about 7 minutes to finish the clustering analysis. Even when the cell number is as large as a hundred thousand, scMDC just takes about 1 hour to finish the clustering analysis. All results are obtained on the Nvidia Tesla P100 with 16Gb memory.

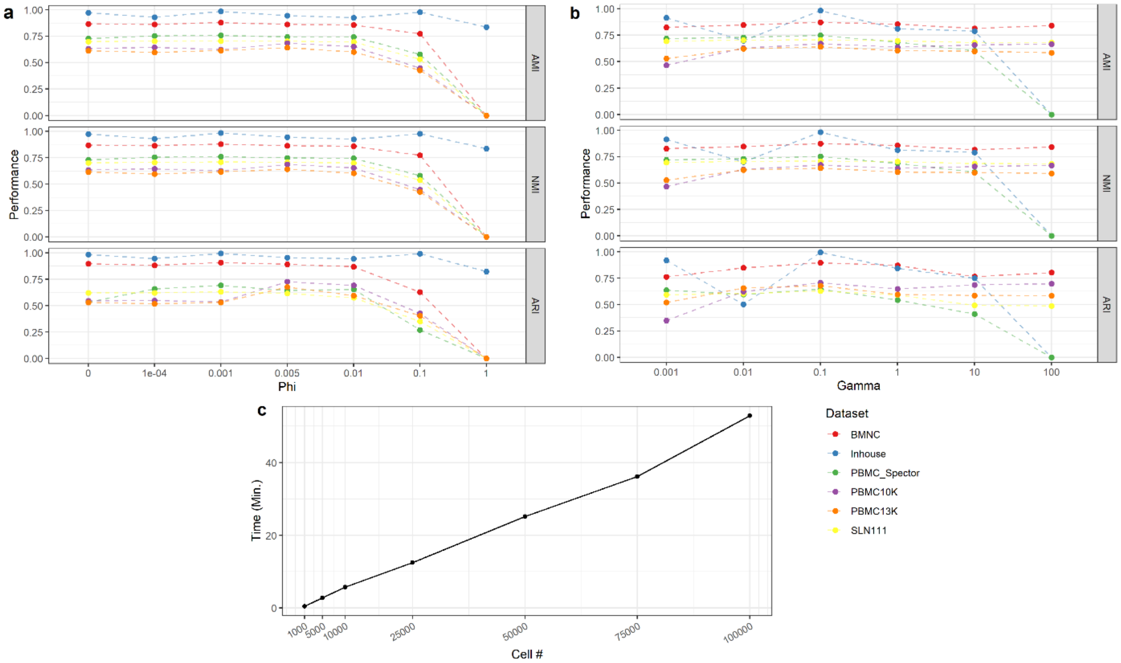


Figure 3.9 Hyperparameter tuning and running time testing of scMDC. This experiment is conducted on six real datasets (n=6). (a) Phi and (b) Gamma are set ranging from 0 to 1 and 0 to 100, respectively. (c) We test the running time of scMDC by increasing the cell numbers in the simulated datasets from 1000 to 100000 (n=7).

3.3 Discussion

We have introduced scMDC - a multimodal deep clustering method for clustering analysis of different single-cell multi-omics data. scMDC jointly models both mRNA and ADT/ATAC data by employing a multimodal autoencoder. Deep K-means clustering is conducted on the bottleneck of the autoencoder, and a KL-loss is employed to facilitate separating distinct cell groups. scMDC is an end-to-end deep model, and all components are optimized simultaneously. Current existing clustering methods for CITE-seq data either apply a shallow Bayes model, such as BREM-SC, or combine two distance-based graphs of mRNA and

ADT, such as CiteFuse and Seurat, to leverage information from different data sources. These methods do not explicitly model dropout events and overdispersions in mRNA and/or ADT count data. Our real-data results demonstrate that the multimodal-based deep learning approach can characterize different sources of count data of CITE-seq and SMAGE-seq more effectively and efficiently.

The clustering results are essential for the downstream analyses, such as differential expression and gene set enrichment analysis. We employ a deep learning-based differential expression algorithm (Lu et al., 2021) to rank genes in a target cluster based on their confidence of being assigned to the target cluster. Given the ranked list of genes, GSEA can be performed to profile cell types at a functional level. The advantages of this deep differential expression method over the traditional methods, such as Wilcoxon-test and DEseq2 (Love, Huber, & Anders, 2014), have been demonstrated by Lu et al (Lu et al., 2021). With the acceleration of GPU, scMDC is very efficient for large multi-omics datasets. Taking all results together, we conclude that scMDC is a promising method for the clustering analysis of single-cell multi-omics data.

3.4 Methods and Materials

3.4.1 Count data preprocessing

The raw CITE-seq data is preprocessed and normalized by the Python package SCANPY (Wolf, Angerer, & Theis, 2018). mRNA and ADT data are normalized separately but using the same method. Specifically, the genes and ADTs with no count are filtered out. The counts of a cell are normalized by a size factor s_i (specifically, s_i^p for ADT data and s_i^r for mRNA data), which is calculated as dividing the library size of that cell by the median of the library size of all cells. In this way, all cells will have the same library size and become comparable. Finally, the counts are transformed into logarithms and scaled to have unit variance and zero mean. The treated count data of mRNA and ADT are used in our denoising multi-modal autoencoder model. We use the raw count matrix to calculate the ZINB loss (Eraslan et al., 2019; Lopez et al., 2018). Before processing the Single-cell Multiome ATAC Gene Expression (SMAGE-seq) data, we map all the reads from scATAC-seq to the gene regions (see details below). Then we use the same methods to preprocess and normalize SMAGE-seq data as for CITE-seq data. The size factor s_i^a for ATAC data is also calculated.

3.4.2 Denoising hierarchical multi-modal autoencoder

The autoencoder is a neural network that is able to learn nonlinear representations efficiently (Hinton & Salakhutdinov, 2006). There are various types of autoencoder models. The denoising autoencoder receives corrupted data with artificial noises and reconstructs the original data (Vincent, Larochelle, Bengio, & Manzagol, 2008). It is widely used for noisy datasets to learn robust latent representation. We use the denoising autoencoder for the mRNA, ADT, and ATAC data since they are very noisy. Let's denote the preprocessed counts of mRNA, ADT, and ATAC as $\mathbf{X^r}$, $\mathbf{X^p}$, and $\mathbf{X^a}$ and the corrupted mRNA, ADT and ATAC data as $\mathbf{X_c^r}$, $\mathbf{X_c^p}$, and $\mathbf{X_c^a}$, formally:

$$\mathbf{X_c^r} = \mathbf{X^r} + \sigma_r * \mathbf{n_r} \tag{3.1}$$

$$\mathbf{X_c^p} = \mathbf{X^p} + \sigma_P * \mathbf{n_p} \tag{3.2}$$

$$\mathbf{X_c^a} = \mathbf{X^a} + \sigma_a * \mathbf{n_a} \tag{3.3}$$

where $\mathbf{n_r}$, $\mathbf{n_p}$ and $\mathbf{n_a}$ are the artificial gaussian noise (with mean=0 and variance=1) for mRNA, ADT and ATAC data respectively, and σ_r, σ_p, and σ_a controls the weights of n_r, n_p and n_a. We set σ_r and σ_a as 2.5 and σ_p as 1.5.

Next, ADT/ATAC and mRNA data are reduced to latent spaces by an autoencoder model. Our autoencoder model contains one encoder (*E*) for the concatenated data and two decoders (*D*) for different omics of data. Both the encoder and decoders are multi-layered fully connected neural networks. We denote encoder $\mathbf{Z} = E_{\mathbf{w}}(\mathbf{X_c^r} \odot \mathbf{X_c^p})$ for the concatenated mRNA and ADT data, encoder $\mathbf{Z} = E_{\mathbf{w}}(\mathbf{X_c^r} \odot \mathbf{X_c^a})$ for the concatenated mRNA and ATAC data, and decoder $\mathbf{X^{a\prime}} = D^a_{\mathbf{w_a'}}(\mathbf{Z_a})$ for ATAC data, decoder $\mathbf{X^{p\prime}} = D^p_{\mathbf{w_p'}}(\mathbf{Z_p})$ for ADT data, and decoder $\mathbf{X^{r\prime}} = D^r_{\mathbf{w_r'}}(\mathbf{Z_r})$ for mRNA data. $\mathbf{w}$ and $\mathbf{w}'$ stand for the learnable weights of the encoder end decoders, respectively. $\odot$ indicates the concatenation of two matrices. The ELu activation function (Clevert, Unterthiner, & Hochreiter, 2015) is used for all the hidden layers in the encoder and decoders and batch normalization is performed on the output of all the hidden layers. The reconstruction loss functions of our autoencoder model are:

$$L_{ADT} = L(\mathbf{X^p}, D^p_{\mathbf{w_p'}}(E_{\mathbf{w}}(\mathbf{X_c^{con}}))) \tag{3.4}$$

$$L_{ATAC} = L(\,\mathbf{X^a}, D^{a}_{\mathbf{w_a'}}(E_{\mathbf{w}}(\mathbf{X_c^{con}}))) \tag{3.5}$$

$$L_{mRNA} = L(\,\mathbf{X^r}, D^{r}_{\mathbf{w_r'}}(E_{\mathbf{w}}(\mathbf{X_c^{con}}))) \tag{3.6}$$

where $\mathbf{X_c^{con}}$ stands for the concatenated data from either mRNA + ADT or mRNA + ATAC. For all the omics of data, we employ the zero-inflated negative binomial (ZINB) models as the reconstruction loss function (Tian et al., 2019). Note, the raw count data is used in the ZINB models (Eraslan et al., 2019; Lopez et al., 2018; Tian et al., 2019). Let X_{ij}^{p} be the count for cell *i* and protein *j* in the raw count matrix of ADT, X_{ij}^{a} be the count for cell *i* and gene *j* in the raw count matrix of ATAC, and X_{ij}^{r} be the count for cell *i* and gene *j* in the raw count matrix of mRNA. The NB distributions are parameterized by means μ_{ij}^{p} , μ_{ij}^{a} and μ_{ij}^{r} , dispersions θ_{ij}^{p}, θ_{ij}^{a} and θ_{ij}^{r} , for ADT, ATAC and mRNA respectively. Formally:

$$NB\left(X_{ij}^{p}\middle|\mu_{ij}^{p},\theta_{ij}^{p}\right) = \frac{\Gamma\left(X_{ij}^{p}+\theta_{ij}^{p}\right)}{X_{ij}^{p}!\Gamma\left(\theta_{ij}^{p}\right)}\left(\frac{\theta_{ij}^{p}}{\theta_{ij}^{p}+\mu_{ij}^{p}}\right)^{\theta_{ij}^{p}}\left(\frac{\theta_{ij}^{p}}{\theta_{ij}^{p}+\mu_{ij}^{p}}\right)^{X_{ij}^{p}} \tag{3.7}$$

$$NB\left(X_{ij}^{a}\middle|\mu_{ij}^{a},\theta_{ij}^{a}\right) = \frac{\Gamma\left(X_{ij}^{a}+\theta_{ij}^{a}\right)}{X_{ij}^{a}!\Gamma\left(\theta_{ij}^{a}\right)}\left(\frac{\theta_{ij}^{a}}{\theta_{ij}^{a}+\mu_{ij}^{a}}\right)^{\theta_{ij}^{a}}\left(\frac{\theta_{ij}^{a}}{\theta_{ij}^{a}+\mu_{ij}^{a}}\right)^{X_{ij}^{a}} \tag{3.8}$$

$$NB\left(X_{ij}^{r}\middle|\mu_{ij}^{r},\theta_{ij}^{r}\right) = \frac{\Gamma\left(X_{ij}^{r}+\theta_{ij}^{r}\right)}{X_{ij}^{r}!\Gamma\left(\theta_{ij}^{r}\right)}\left(\frac{\theta_{ij}^{r}}{\theta_{ij}^{r}+\mu_{ij}^{r}}\right)^{\theta_{ij}^{r}}\left(\frac{\theta_{ij}^{r}}{\theta_{ij}^{r}+\mu_{ij}^{r}}\right)^{X_{ij}^{r}} \tag{3.9}$$

ZINB distribution is parameterized by the negative binomial of count data and an additional coefficient π_{ij}^{p}, π_{ij}^{a} and π_{ij}^{r} for the probabilities of dropout events:

$$ZINB\left(X_{ij}^{p}\middle|\mu_{ij}^{p},\theta_{ij}^{p},\pi_{ij}^{p}\right) = \pi_{ij}^{p}\delta_0\left(X_{ij}^{p}\right) + (1-\pi_{ij}^{p})NB(X_{ij}^{p}|\mu_{ij}^{p},\theta_{ij}^{p}) \tag{3.10}$$

$$ZINB\left(X_{ij}^{a}\middle|\mu_{ij}^{a},\theta_{ij}^{a},\pi_{ij}^{a}\right)=\pi_{ij}^{a}\delta_{0}\left(X_{ij}^{a}\right)+(1-\pi_{ij}^{a})NB(X_{ij}^{a}|\mu_{ij}^{a},\theta_{ij}^{a}) \tag{3.11}$$

$$ZINB\left(X_{ij}^{r}\middle|\mu_{ij}^{r},\theta_{ij}^{r},\pi_{ij}^{r}\right)=\pi_{ij}^{r}\delta_{0}\left(X_{ij}^{r}\right)+(1-\pi_{ij}^{r})NB(X_{ij}^{r}|\mu_{ij}^{r},\theta_{ij}^{r}) \tag{3.12}$$

To estimate these parameters in the ZINB loss functions, we add three independent fully connected layers $\mathbf{M}$, $\boldsymbol{\theta}$, and $\boldsymbol{\Pi}$ to the last hidden layer of each decoder. The layers are defined as:

$$\mathbf{M_{ADT}}=diag\left(s_{i}^{p}\right)\times\exp\left(\mathbf{w_{p(\mu)}}\mathbf{X^{p\prime}}\right);\ \boldsymbol{\Theta}_{\mathbf{ADT}}=\exp\left(\mathbf{w_{p(\theta)}}\mathbf{X^{p\prime}}\right);$$

$$\boldsymbol{\Pi}_{\boldsymbol{ADT}}=\exp\left(\mathbf{w_{p(\pi)}}\mathbf{X^{p\prime}}\right) \tag{3.13}$$

$$\mathbf{M_{ATAC}}=diag(s_{i}^{a})\times\exp\left(\mathbf{w_{a(\mu)}}\mathbf{X^{a\prime}}\right);\ \boldsymbol{\theta}_{\mathbf{ATAC}}=\exp\left(\mathbf{w_{a(\theta)}}\mathbf{X^{a\prime}}\right)$$

$$\boldsymbol{\Pi}_{\boldsymbol{ATAC}}=\exp\left(\mathbf{w_{a(\pi)}}\mathbf{X^{a\prime}}\right) \tag{3.14}$$

$$\mathbf{M_{RNA}}=diag(s_{i}^{r})\times\exp\left(\mathbf{w_{r(\mu)}}\mathbf{X^{r\prime}}\right);\ \boldsymbol{\theta}_{\mathbf{RNA}}=\exp\left(\mathbf{w_{r(\theta)}}\mathbf{X^{r\prime}}\right)$$

$$\boldsymbol{\Pi}_{\boldsymbol{RNA}}=\exp\left(\mathbf{w_{r(\pi)}}\mathbf{X^{r\prime}}\right) \tag{3.15}$$

where $\mathbf{M_{ADT}}$, $\boldsymbol{\theta}_{\mathbf{ADT}}$ and $\boldsymbol{\Pi}_{\mathbf{ADT}}$ are the matrices of estimated mean, dispersion and drop-out probability for the ZINB loss of ADT data, $\mathbf{M_{ATAC}}$, $\boldsymbol{\theta}_{\mathbf{ATAC}}$ and $\boldsymbol{\Pi}_{\mathbf{ATAC}}$ are the matrices of estimated mean, dispersion and drop-out probability for the ZINB loss of ATAC data, and $\mathbf{M_{RNA}}$, $\boldsymbol{\theta}_{\mathbf{RNA}}$ and $\boldsymbol{\Pi}_{\mathbf{RNA}}$ are the matrices of estimated mean, dispersion, and drop-out probability for the ZINB loss of mRNA data. $\mathbf{w_{p(\mu)}}$, $\mathbf{w_{p(\theta)}}$, $\mathbf{w_{p(\pi)}}$, $\mathbf{w_{a(\mu)}}$, $\mathbf{w_{a(\theta)}}$, $\mathbf{w_{a(\pi)}}$, $\mathbf{w_{r(\mu)}}$, $\mathbf{w_{r(\theta)}}$ and $\mathbf{w_{r(\pi)}}$ are the learnable weights. The size factor s_{i}^{p} , s_{i}^{a} and s_{i}^{r} for ADT, ATAC and mRNA are calculated in the preprocessing step. The loss function of ZINB-based autoencoder is defined as:

$$L_{ADT} = \sum_{ij} -\log\left(ZINB(X_{ij}^{p} | \mu_{ij}^{p}, \theta_{ij}^{p}, \pi_{ij}^{p})\right) \tag{3.16}$$

$$L_{ATAC} = \sum_{ij} -\log\left(ZINB(X_{ij}^{a} | \mu_{ij}^{a}, \theta_{ij}^{a}, \pi_{ij}^{a})\right) \tag{3.17}$$

$$L_{mRNA} = \sum_{ij} -\log\left(ZINB(X_{ij}^{r} | \mu_{ij}^{r}, \theta_{ij}^{r}, \pi_{ij}^{r})\right) \tag{3.18}$$

for ADT, ATAC and mRNA data.

3.4.3 Conditional autoencoder

Conditional autoencoder (CAE) has been designed to integrate the data from different batches (Gayoso et al., 2021). Based on the traditional autoencoder model, we add a matrix **B** on the input of the encoder and decoders. **B** is the one-hot coding from a batch vector b of cells. If there are M batches in b, the dimension of **B** would be $N \times M$. So, the encoder becomes $\mathbf{Z} = E_{\mathbf{w}}(\mathbf{X}_{\mathbf{c}}^{\mathbf{con}} \odot \mathbf{B})$ and the decoders become $\mathbf{X}^{\mathbf{p}\prime} = D_{\mathbf{w}_p^{\prime}}^{p}(\mathbf{Z} \odot \mathbf{B})$ for ADT, $\mathbf{X}^{\mathbf{a}\prime} = D_{\mathbf{w}_a^{\prime}}^{a}(\mathbf{Z} \odot \mathbf{B})$ for ATAC , and $\mathbf{X}^{\mathbf{r}\prime} = D_{\mathbf{w}_r^{\prime}}^{r}(\mathbf{Z} \odot \mathbf{B})$ for mRNA data.

3.4.4 Model architecture

Our model can be used for clustering CITE-seq data and SMAGE-seq data. For CITE-seq data, the encoder is set as {256, 64, 32, 16}, the decoder for mRNA is set as {16, 64, 256} and the decoder for ADT is set as {16 20}. For SMAGE-seq data, the encoder is set as {256, 128, 64} and the decoders for both mRNA and ATAC data are set as {64, 128, 256}. So, the latent space of CITE-seq and SMAGE-seq data has 16 and 64 dimensions respectively. The overall architecture of the scMDC model is shown in **Figure 3.1**.

3.4.5 KL divergence on the latent layer

In the clustering analysis, similar points should be grouped into the same cluster. According to the method described by Chen et al. (L. Chen et al., 2020), we employ a KL divergence loss function to enhance the association between similar cells and prevent squeezing the centroids of clusters in the latent space. Following t-SNE (Maaten & Hinton, 2008), the t-distribution kernel function is used to describe the pairwise similarity among two cells *i* and *i'* in latent space of the high-level autoencoder:

$$q_{ii'} = \frac{(1+||\mathbf{Z_i}-\mathbf{Z_{i'}})||^2)^{-1}}{\sum_{l\neq i}(1+||\mathbf{Z_i}-\mathbf{Z_l})||^2)^{-1}} \tag{3.19}$$

where $q_{ii} = 0$. The **P** is the target distribution in training, which strengthens and weakens the affinities between the cells with high and low similarities, respectively. **P** is defined as the square of **Q** then normalized:

$$p_{ii'} = \frac{q_{ii'}^2 / \sum_{i\neq i'} q_{ii'}}{\sum_{l\neq i}(q_{il}^2 / \sum_{i\neq l} q_{il})} \tag{3.20}$$

With the two similarity distributions, we construct the KL loss function by the Kullback-Leibler (KL) divergence between **Q** and the derived target distribution **P**:

$$L_{kl} = KL(\mathbf{P} \parallel \mathbf{Q}) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3.21}$$

which measure the probability-distance between the two distributions. During the training process, $\mathbf{P}$ and $\mathbf{Q}$ are calculated per batch.

3.4.6 Deep K-means clustering

We perform unsupervised clustering on the latent space of the autoencoder (L. Chen et al., 2020). Our multimodal autoencoder learns a non-linear mapping for each cell *i*, which transfers two input matrices to a low dimensional space **Z**. The clustering loss function is defined as:

$$L_c = \sum_{i=1}^{N} \sum_{j=1}^{K} w_{ij} \tau f(\mathbf{Z_i}, V_j) \tag{3.22}$$

where $\mathbf{V}$ stands for the *K* clustering centroids and f calculates the Euclidean distance between a cell (in latent space) and a centroid. τ is a hyperparameter. We set τ as 1 for CITE-seq data and 0.1 for SMAGE-seq data. The Gaussian kernel function is applied in weight measuring to smooth the gradient descent optimization process:

$$\widetilde{w}_{ij} = \frac{\exp(-f(\mathbf{Z_i}, V_j))}{\sum_{k=1}^{K} \exp(-f(\mathbf{Z_i}, V_k))} \tag{3.23}$$

Then, to speed up the convergence, an inflation operation is applied on the weights:

$$w_{ij} = \frac{\widetilde{w}_{ij}^{\alpha}}{\sum_{k}^{K=1} \widetilde{w}_{ik}^{\alpha}} \tag{3.24}$$

where the hyperparameter α is set to 2.

The total loss of scMDC is defined as:

$$\underset{\mathbf{w}, \mathbf{w}'_{\boldsymbol{p}}, \mathbf{w}'_{\boldsymbol{r}}, \mathbf{U}}{\operatorname{argmin}} L_{total}(\mathbf{X^p}, \mathbf{X^r} | \mathbf{w}, \mathbf{w}'_{\boldsymbol{p}}, \mathbf{w}'_{\boldsymbol{r}}, \mathbf{U}) = L_{mRNA}(\mathbf{X^r} | \mathbf{w}, \mathbf{w}'_{\boldsymbol{r}}) + L_{ADT}(\mathbf{X^p} | \mathbf{w}, \mathbf{w}'_{\boldsymbol{p}}) + \gamma * L_c(\mathbf{X^r}, \mathbf{X^p} | \mathbf{w}, , \mathbf{U}) + \varphi * L_{kl}(\mathbf{X^r}, \mathbf{X^p} | \mathbf{w}) \tag{3.25}$$

for CITE-seq data, and

$$\underset{\mathbf{w},\mathbf{w}'_{\boldsymbol{a}},\mathbf{w}'_{\boldsymbol{r}},U}{\operatorname{argmin}} L_{total}(\mathbf{X}^{\mathbf{a}},\mathbf{X}^{\mathbf{r}}|\mathbf{w},\mathbf{w}'_{\boldsymbol{a}},\mathbf{w}'_{\boldsymbol{r}},\mathbf{U}) = L_{mRNA}(\boldsymbol{X}^{\boldsymbol{r}}|\mathbf{w},\mathbf{w}'_{\boldsymbol{r}}) + L_{ATAC}(\mathbf{X}^{\mathbf{a}}|\mathbf{w},\mathbf{w}'_{\boldsymbol{a}}) + \gamma * L_c(\mathbf{X}^{\mathbf{r}},\mathbf{X}^{\mathbf{a}}|\mathbf{w},\mathbf{U}) + \varphi * L_{kl}(\mathbf{X}^{\mathbf{r}},\mathbf{X}^{\mathbf{a}}|\mathbf{w}) \tag{3.26}$$

for SMAGE-seq data. $\mathbf{w}$ is the weight matrix of the encoder. $\mathbf{w}'_{\boldsymbol{a}}, \mathbf{w}'_{\boldsymbol{p}}, and\ \mathbf{w}'_{\boldsymbol{r}}$are the weights of mRNA decoder, ADT decoder and ATAC decoder, respectively. $\mathbf{U}$ is a set of centroids. Here, γ and φ are the hyper-parameters that control weights for the clustering loss and the KL loss, respectively. Value of γ is set as 0.1 for all experiments. φ is set to 0.001 for CITE-seq data and 0.005 for SMAGE-seq data.

3.4.7 Marker gene detection

We employ an approach proposed by Lu et al. (Lu et al., 2021) to find marker genes in each cluster against another cluster or the rest of the clusters. Briefly, for each gene, this algorithm will find the minimal perturbation that alters the group assignment from a source group (s) to the target group(s) (t). The objective function for one-to-one comparison is:

$$\min_{\delta} \parallel \delta \parallel + \ \lambda \max\left(0, \alpha + m_s(\mathbf{x}+\delta) - m_t(\mathbf{x}+\delta)\right) \tag{3.27}$$

where the tradeoff coefficient λ and the margin α are set to 100 and 1, respectively. $\mathbf{x} \in \mathbf{X}$ is the normalized data of a cell. $\delta \in \mathbb{R}^P$ is the perturbation for altering the cluster assignment of cells. L1 norm of δ is used to encourage sparsity and non-redundancy. The objective function for one-to-rest comparison is:

$$\min_{\delta} \parallel \delta \parallel + \ \lambda \max\left(0, \alpha + m_s(\mathbf{x} + \delta) - \max_{t \neq s} m_t(\mathbf{x} + \delta)\right) \tag{3.28}$$

It is equal to comparing a source cluster to a target cluster for which cell x has the highest confidence. The confidence from a cell x to a cluster c is defined as:

$$m_c(\mathbf{x}) = \log\left(\frac{\exp\left(-\beta \| E_{\mathbf{w}}(\mathbf{x}) - \mu_c \|\right)}{\sum_k \exp\left(-\beta \| E_{\mathbf{w}}(\mathbf{x}) - \mu_k \|\right)}\right) \tag{3.29}$$

where μ_c is the centroid of cluster c and β is set to 1. Besides the mRNA matrix, this algorithm can also be applied to ADT and ATAC matrix. The gene rank learned from ACE is then multiply by a direction vector to get the directed gene rank. The direction vector of genes is calculated based on the log fold change between clusters by changing positive values to 1 and negative values to -1. Based on the directed gene rank, gene set enrichment analysis (GSEA) is performed by the package fgsea (v1.19.4) and msigdbr (v7.4.1) in R.

3.4.8 Model implementation

The model is implemented in Python3 using PyTorch (Paszke et al., 2017). Adam with AMSGrad variant (Kingma & Ba, 2014; Reddi, Kale, & Kumar, 2018) with initial learning rate = 0.001 is used for the pretraining stage. We Adadelta optimizer(Zeiler, 2012) with learning rate = 1 and rho = 0.95 is used in the clustering stage. The batch size is set as 256. We pretrain the autoencoders for 400 epochs before entering the clustering stage. In the pretraining stage, we optimize the reconstruction losses in the first 200 epochs. The KL loss (L_{kl}) on the bottleneck layer is added to the training in the remaining 200 epochs. After pretraining, the users need to specify the number of clusters (K). In the beginning of clustering stage, we initialize K centroids by k-means algorithm. During the clustering stage, all loss functions including clustering loss (L_c) are optimized simultaneously, and the centroids are also continuously updated by the learning process. The convergence threshold for the clustering stage is that clustering labels are changed less than 0.1% per epoch. All experiments of scMDC in this study are conducted on Nvidia Tesla P100 (16G) GPU.

3.4.9 Competing methods

BREM-SC (v0.2.0, https://github.com/tarot0410/BREMSC) (X. Wang et al., 2020), CiteFuse (v1.0.0, https://github.com/SydneyBioX/CiteFuse) (Kim et al., 2020), Seurat (v4.0.4, https://github.com/satijalab/seurat) (Butler et al., 2018), IDEC (https://github.com/XifengGuo/IDEC) (Xie et al., 2016), k-means (sklearn v0.22.2, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html), SC3 (v1.21.0, https://github.com/hemberg-lab/SC3) (Kiselev et al., 2017), SCVIS (v0.1.0, https://github.com/shahcompbio/scvis) (Ding, Condon, & Shah, 2018), Tscan (v1.31.0, https://github.com/zji90/TSCAN) (Ji & Ji, 2016) , TotalVI (scvi-tools v0.15.0, https://scvi-tools.org/), Cobolt (v1.0.0, https://github.com/epurdom/cobolt) (Gong et al., 2021), scMM (v1.0.0, https://github.com/kodaim1115/scMM)(Minoura et al., 2021) and Specter (https://github.com/canzarlab/Specter) (Ringeling & Canzar, 2021) are used as competing methods. For the multimodal methods, ADT/ATAC and mRNA data are used as input, and standard normalization is applied if authors described. For single data source methods, ADT/ATAC and mRNA matrices are preprocessed and normalized separately then concatenated as a single input. To keep consistency, all the methods use the same highly variable genes in RNA and ATAC data and use full ADTs in the CITE-seq data. If the methods require normalized data as inputs without defining a specific way of normalization, we apply the same normalization method as that for scMDC (described in section 3.4.1). Before doing K-means clustering, PCA is performed on the normalized mRNA data and the top 20 PCs are used for clustering. BREM-SC uses the raw count matrix as input directly. The data normalization for Citefuse follows the vignette (https://sydneybiox.github.io/CiteFuse/articles/CiteFuse.html). Specifically, mRNA counts are normalized by the function “logNormCounts” in the Scater package(McCarthy, Campbell, Lun, & Wills, 2017) with default settings. ADT counts are normalized and log-transformed by the function “normaliseExprs” from the CiteFuse package. Seurat uses the raw count matrices as input. Following the CITE-seq tutorial of Seurat, we use “LogNormalize” for mRNA and “centered log-ratio transformation” for ADT data normalization. Then the function “ElbowPlot” is used to find the best PCs (principal components) for clustering. The resolution in “FindClusters” function of Seurat is adjusted for different datasets in order to estimate a satisfactory number of clusters that are close to the real *K*. For the single-omics and multi-omics clustering, the function ‘FindNeighbors’ and ‘FindMultiModalNeighbors’ (Hao et al., 2021) are used to find the neighbors of cells by the SNN (shared nearest-neighbor) and WNN (weighted nearest-neighbor) algorithms, respectively. For IDEC and TScan, normalized data are provided as inputs. SC3 needs both the raw data and the normalized data as input. When the cell number is higher than 5000, SC3 runs a SVM to estimate the cell types of the extra cells in a supervised manner. SCVIS is a variational autoencoder-based model aimed to reduce the dimension of scRNA-seq data. According to the author's protocol (Ding et al., 2018), the count data are firstly processed as log2(CPM/10 + 1), where 'CPM' means the 'counts per million'. Next, we concatenate CPMs of mRNA and ADT. Then the 100 PCs are extracted from the CPM matrix by PCA and used as the input for SCVIS analysis. K-means clustering is performed on the latent output of SCVIS. For TotalVI, we keep the default setting for all the datasets according to the official pipeline (https://docs.scvi-tools.org/en/stable/tutorials/notebooks/totalVI.html). We then perform Kmeans clustering on the latent space of datasets from TotalVI since the number of clusters is supposed to be known. Specter(Ringeling & Canzar, 2021) uses the normalized RNA and ADT expression data as the input. We used the default setting for Specter's multimodal analysis. For SMAGE-seq datasets, we compare our model to four competing methods: k-means + PCA, Seurat, scMM, and Cobolt. All the methods use the top 2000 highly variable mRNA and ATAC data from the SMAGE-seq data. If the methods need normalized data as input, we apply the same normalization method for it as that for scMDC. Before doing K-means, PCA is performed on both mRNA and ATAC data and the top 20 PCs of each are used for clustering. For Seurat, the ATAC data, which is mapped to the gene regions, is processed in the same way as for the mRNA data. Then WNN algorithm is used for integrating multimodal data as described before. For Cobolt, we follow the official pipeline (https://github.com/epurdom/cobolt/blob/master/docs/tutorial.ipynb) to produce the data embeddings. We then perform K-means clustering on the latent space of datasets since the number of clusters is supposed to be known. We followed the tutorial provided of scMM (v1.0.0)(Minoura et al., 2021) and used the default parameters. The embeddings of scMM are obtained and used for the K-means clustering.

3.4.10 Evaluation metrics

Adjust Rand Index (ARI)(Hubert & Arabie, 1985), Adjusted Mutual Information (AMI)(Vinh, Epps, & Bailey, 2010), and Normalized Mutual Information (NMI)(Alexander & Joydeep, 2003) are used as metrics to evaluate the clustering performance.

Adjust Rand Index measures the agreements between two sets **C** and **G**. Assuming a is the number of pairs of two objects in the same group in both **C** and **G**; b is the number of pairs of two objects in different groups in both **C** and **G**; c is the number of pairs of two objects in the same group in **C** but in different groups in **G**; and d is the number of pairs of two objects in different groups in **C**, but in the same group in **G**. The ARI is defined as:

$$ARI = \frac{\binom{n}{2}(a+d)-[(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}-[(a+b)(a+c)+(c+d)(b+d)]} \tag{3.30}$$

Let **C** = {*C1, C2, …, C*$_{tc}$}and **G** = {*G1, G2, …, G*$_{tg}$} be the predicted and ground truth labels on a dataset with n cells. NMI is defined as:

$$NMI = \frac{I(\mathbf{C},\mathbf{G})}{\max\{H(\mathbf{C}),H(\mathbf{G})\}} \tag{3.31}$$

where $I(\mathbf{C},\mathbf{G})$ represents the mutual information between **C** and **G** and is defined as:

$$I(\mathbf{C},\mathbf{G}) = \sum_{p=1}^{tc} \sum_{q=1}^{tg} |C_p \cap G_q| \log \frac{n|C_P \cap G_q|}{|C_p| \times |G_q|} \tag{3.32}$$

and H(C) and H(G) are the entropies:

$$H(\mathbf{C}) = -\sum_{p=1}^{tc} |C_p| \log \frac{|C_p|}{n} \tag{3.33}$$

$$H(\mathbf{G}) = -\sum_{p=1}^{tg} |G_p| \log \frac{|G_p|}{n} \tag{34}$$

Similarly, AMI is defined as:

$$AMI(\mathbf{C},\mathbf{G}) = \frac{I(\mathbf{C},\mathbf{G}) - E\{I(\mathbf{C},\mathbf{G})\}}{\max\{\mathrm{H}(\mathbf{C}),\mathrm{H}(\mathbf{G})\} - E\{I(\mathbf{C},\mathbf{G})\}} \tag{35}$$

The extra component $E\{I(\mathbf{C},\mathbf{G})\}$ is the expected mutual information between two random clusters (Vinh et al., 2010).

To illustrate the superiority of scMDC over the competing methods in multiple datasets, we rank the methods based on their clustering performance (AMI, NMI, and ARI) on each dataset. The lower the rank, the better the performance. Besides, a one-sided paired t-test is conducted to test if the clustering metrics (NMI, AMI, and ARI) of scMDC are significantly higher than that of the competing methods, which is implemented by the “t.test()” function in R. Nominal p-value <0.05 is considered to indicate a significant difference.

3.4.11 Public real datasets

The real CITE-seq datasets used in this study are summarized in **Table 1.1**. The GSE100866 dataset is downloaded from GEO (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100866). The cells in this dataset are cord blood mononuclear (CBMN) cells and annotated by Wang et al. from marker genes and ADTs (X. Wang et al., 2020). Cells with 'Unknown' cell types were filtered out. The bone marrow mononuclear cells (BMNC, GSE128639) and the cell type labels are downloaded from the "bmcite" dataset in "SeuratData" package (v0.2.1). The mouse spleen lymph node datasets (SLN208 and SLN111, GSE150599) and the cell type labels are provided by TotalVI (Gayoso et al., 2021) on GitHub (https://github.com/YosefLab/totalVI_reproducibility). Cells are also filtered by the author. PBMC dataset is available on the 10X website (https://support.10xgenomics.com/single-cell-gene-expression/datasets). We downloaded the preprocessed data and the cell type labels from the GitHub of Specter (https://github.com/canzarlab/Specter)(Ringeling & Canzar, 2021).

The real Single-cell Multiome ATAC Gene Expression (SMAGE-seq) datasets used in this study are summarized in **Table 1.2**. All the SMAGE-seq datasets are downloaded from the 10X Genomics website (https://www.10xgenomics.com/resources/datasets). The first and second datasets are from human peripheral blood mononuclear cells (PBMCs) with about 3k and 10k cells. We denote them as PBMC3K and PBMC10K respectively. The third dataset is from the E18 mouse brain. We denote it as E18.

For each dataset, mRNA counts are downloaded directly while the ATAC gene counts are generated by us. Specifically, after filtering the reads by ATAC peak region fragments, nucleosome signal, and TSS enrichment, we mapped each read to a gene region by the function ‘GeneActivity’ in Signac (v1.4.0) (Stuart, Srivastava, Lareau, & Satija, 2020). All the steps are referred to the official pipeline from Satija lab. Then, the PBMC cells are annotated by the label transferring method in Seurat V3(Stuart et al., 2020) with the reference datasets “pbmc_10k_v3.rds” (https://www.dropbox.com/s/zn6khirjafoyyxl/pbmc_10k_v3.rds?dl=0) provided by Satija lab. For the E18 dataset, we transfer the labels from another mouse brain dataset (GSE126074 P0 mouse brain cortex) and the cell type labels are provided by the author of the SNARE-seq paper(S. Chen et al., 2019).

3.4.12 Simulation

The simulated data are generated by the R package SymSim (0.0.0.9000)(Zhang, Xu, & Yosef, 2019). The overall setting for simulation is from the Online vignettes of SymSim (https://github.com/YosefLab/SymSim). This setting was estimated from the Zeisel 2015 dataset (Zeisel et al., 2015). We lower the parameter “n_de_evf” to 5 to keep about 50% differential expressed genes/ADTs in the dataset. We performed three experiments to test the clustering performance of scMDC and generate 10 datasets in each experiment. In the first experiment, we adjusted the parameter “Sigma” in the function SimulateTrueCounts() to 0.6, 0.7, and 0.8 in mRNA and 0.3, 0.4, and 0.5 in ADT to simulate the high, medium, and low clustering signal among clusters (cell types). We give a lower sigma (higher signal) to ADT data than mRNA data since it has a higher signal-to-noise ratio in the real datasets (X. Wang et al., 2020). In the second experiment, we adjust the parameter “alpha_mean” in function True2ObservedCounts() to 0.001, 0.00075, 0.0005 in mRNA and 0.05, 0.045, 0.04 in ADT data to simulate low, medium, and high dropout rates. These settings are also consistent with that in the real datasets since mRNA has higher dropout rates than ADT data. In the third experiment, we added a batch effect in the data to test the model’s performance in batch effect correction. Medium signal and dropout rate are used in this data and the parameter “batch_effect_size” in function DivideBatches() is set to 1. All the simulated datasets have 8 groups, 1000 cells, 2000 genes, and 30 ADTs.

3.4.13 Data availability

The GSE100866 data used in this study are available in the GEO database under accession code GSE100866 [https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100866]. Cell type labels are downloaded from the GitHub of BREM-SC (https://github.com/tarot0410/BREMSC). The BMNC dataset and the cell type labels are downloaded from the “bmcite” dataset in “SeuratData” package (https://github.com/satijalab/seurat-data). The mouse spleen lymph node datasets (SLN208 and SLN111) and the cell type labels are provided by TotalVI (Gayoso et al., 2021) on GitHub (https://github.com/YosefLab/totalVI_reproducibility). These datasets are sequenced in two batches. PBMC dataset is available on 10x Genomics website (https://support.10xgenomics.com/single-cell-gene-expression/datasets) and the cell type labels are downloaded from the GitHub of Specter (https://github.com/canzarlab/Specter). All SMAGE-seq datasets (PBMC3K, PBMC10K, and mouse brain E18) are downloaded from the 10X Genomics website (https://www.10xgenomics.com/resources/datasets). Labels are transferred by Signac (v1.4.0) from the annotated datasets.

3.4.14 Code availability

Codes supporting this study are available on GitHub:

https://github.com/xianglin226/scMDC/releases/tag/v1.0.0.

CHAPTER 4

SPATIAL-RESOLVED SCRNA-SEQ MODEL – DSSC

4.1 Introduction

In this chapter, we introduce a novel clustering approach for sp-scRNA-seq data, DSSC (**D**eep **S**patial-constrained **S**ingle-cell **C**lustering). DSSC integrates the prior information from both the physical organization of cells and the expression of the spatial dependent marker genes into the clustering process by a denoising graphical autoencoder with cell-to-cell constraints. Our extensive experiments indicated that DSSC outperforms the state-of-the-art methods in both simulated and real datasets, revealing that it is a promising tool for spatial-resolved single-cell data clustering.

4.2 Experiments and Results

4.2.1 Simulation experiments

DSSC is developed for clustering spatial-resolved single-cell data by integrating the prior knowledge from cell/spot location and marker genes. The overall architecture of the DSSC model is shown in **Figure 4.1**. In the simulation experiments, we test the performance of DSSC on the data in different cell-type spatial organizations and dependencies. We simulated the scRNA-seq data by Splatter and placed them in the spatial locations from two real datasets from 1) osmFISH data (see **Figure 4.2a**); 2) sample 151673 from spatialLIBD data (see

Figure 2b); We adjust the cell-type spatial dependencies by perturbing the spatial coordinates of 10%, 15%, and 20% of total cells (see details in the method section). Constraints are built based on the true labels with 5% perturbations. We compare DSSC with seven existing clustering methods including SpaGCN, stLearn, Seurat, Giotto, BayesSpace, *k*-means + PCA, and SC3. We compare both the clustering performance (measured by AC, NMI, and ARI) and the predicted label's spatial heterogeneity (denoted as PLSH, measured by KNN ACC and Moran's I) of these methods. The results of simulation experiments are shown in **Figure 4.4**. Generally, we find that the spatial-based clustering methods (DSSC, SpaGCN, stLearn, BayesSpace, and Giotto) have higher clustering performance and PLSH than the traditional scRNA-seq clustering methods (Seurat, SC3, and *k*-means). Cell-type spatial-dependency is negatively correlated with the performance of the spatial-based clustering methods, but it has no influence on the performance of the traditional clustering methods. BayesSpace cannot encode the spatial coordinates of the osmFISH data, so the clustering performance and PLSH of it are much higher in spatial organization 2 (see **Figure 4.2b**) than in spatial organization 1 (see **Figure 4.2a**). Although DSSC outperforms the competing methods in both spatial organizations, its advantage is much higher in spatial organization 1 than in spatial organization 2. In summary, these results reveal that DSSC's performance is not affected by the sequencing technologies and cell type spatial organizations, while other methods may prefer the sequencing-based technologies (such as the 10x Visium). Besides, DSSC can keep a superior performance over the competing methods under low, medium, and high cell-type dependencies. Therefore, these experiments demonstrate the robustness of DSSC's performance. The statistical tests of the clustering performance between DSSC and the competing methods are shown in **Appendix Tables D.1, D.2, and D.3**.

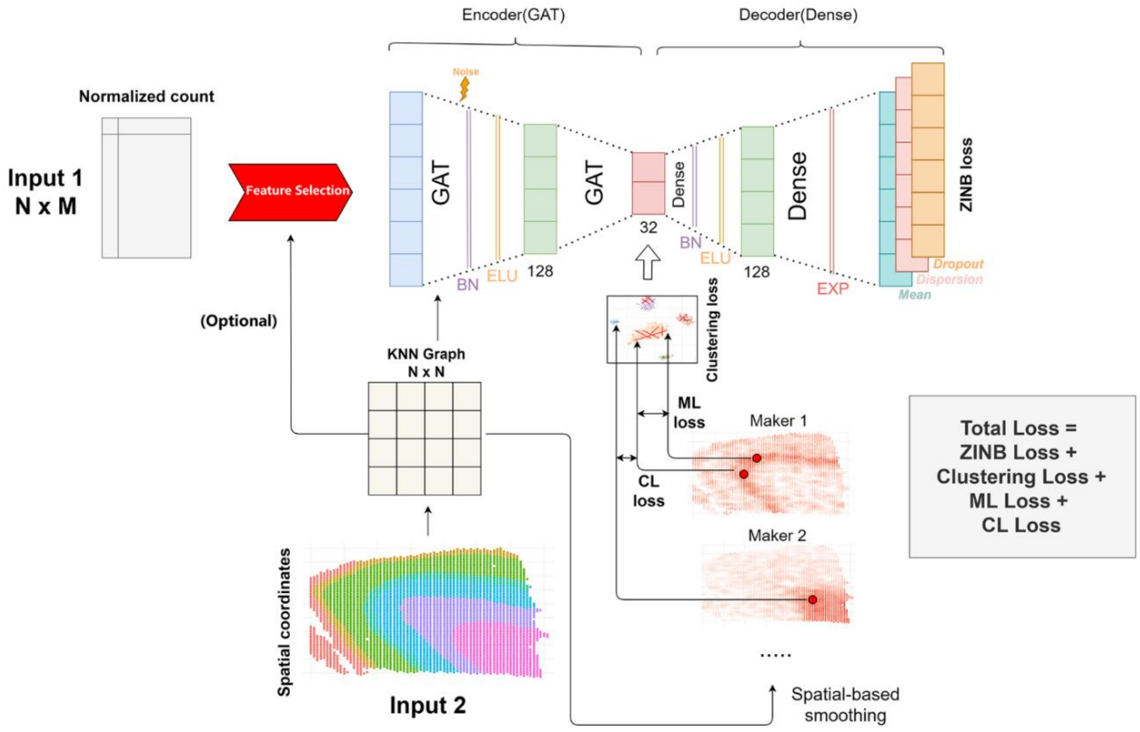


Figure 4.1 DSSC model architecture. The inputs of DSSC are the gene expression matrix and the cell coordinates. The outputs of DSSC are the low-dimension latent space (32D) and the predicted labels. Briefly, DSSC learns a low-dimensional representation of the gene expression matrix while simultaneously leveraging the prior knowledge from the spatial coordinates of cells/spots and the marker genes. Clustering is performed on latent space. Constraint loss, reconstruction loss, and clustering loss are optimized simultaneously. ML loss and CL loss are optimized alternately. Notations: BN stands for the batch normalization; ELU stands for the ELU activation; ML indicates the must-links constraints; CL indicates the cannot-link constraints; ZINB means the zero-inflated negative binominal.

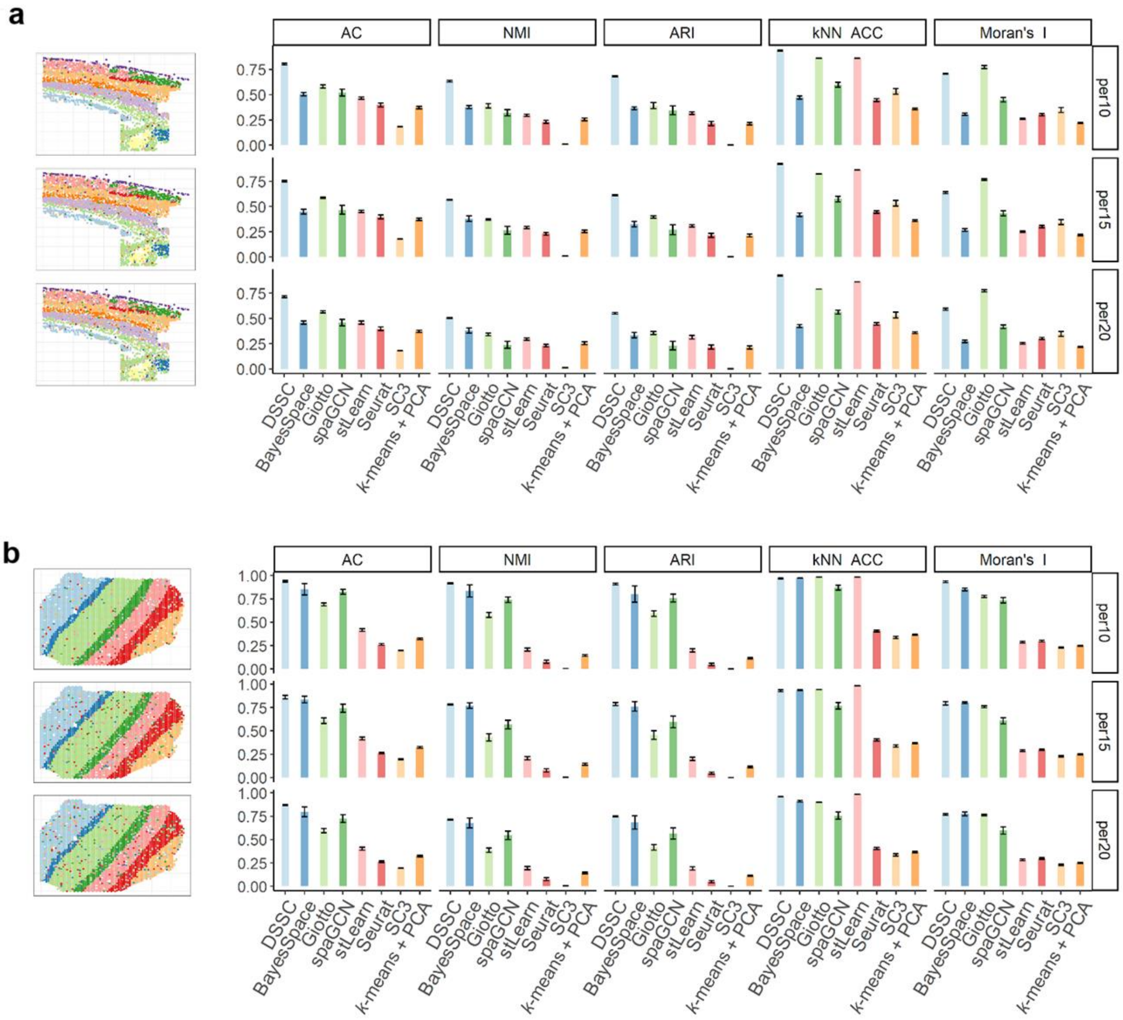


Figure 4.2 Simulation results from the (a) spatial organization 1 (from osmFISH data) and (b) spatial organization 2 (from spatialLIBD sample 151507). True labels with 10%, 15%, and 20% perturbed coordinates are shown on the physical spaces (left). The corresponding clustering results are shown in the bar plots (right).

4.2.2 Real datasets

We then tested the performance of DSSC in three studies including 25 real datasets with 1 dataset from osmFISH (mouse cortex), 12 datasets from spatialLIBD (human cortex), and 12 datasets from 10x Genomics (Mouse brain, denoted as 10xMBAD). In all datasets, we compare DSSC with seven competing methods as described in section 4.2.1. For the data from spatialLIBD and 10xMBAD, we use the markers from the original paper of spatialLIBD (Pardo et al., 2022). Since osmFISH data only has 33 genes, we only use the genes with the top Moran's I.

The results of the osmFISH dataset are shown in **Figure 4.3**. Since the latent dimension of SpaGCN is larger than the feature dimension of this data, we exclude SpaGCN from the competing methods for this experiment. BayesSpace cannot recognize the neighbors from the hybridization technologies, so the spatial information is not used by it for this dataset. The marker genes used here for DSSC *are Rorb* and *Syt6* (see **Figure 4.3c**). As expected, the expression of these genes have high spatial dependency. We find that DSSC can identify the layer structures in the cortex (see **Figure 4.3a**). These layers are not clearly profiled by the competing methods **(**see **Figure 4.3b**). Besides, DSSC outperforms the competing methods in both clustering performance and PLSH (see **Figure 4.3b**). Some spatial-based methods, such as Giotto and stLearn, have very high KNN accuracy, but their clustering performance is much lower than DSSC. A potential reason for this result is that the spatial information overwhelms the clustering signal from the gene expression during the clustering process, resulting in the high spatial dependence but low clustering performance.

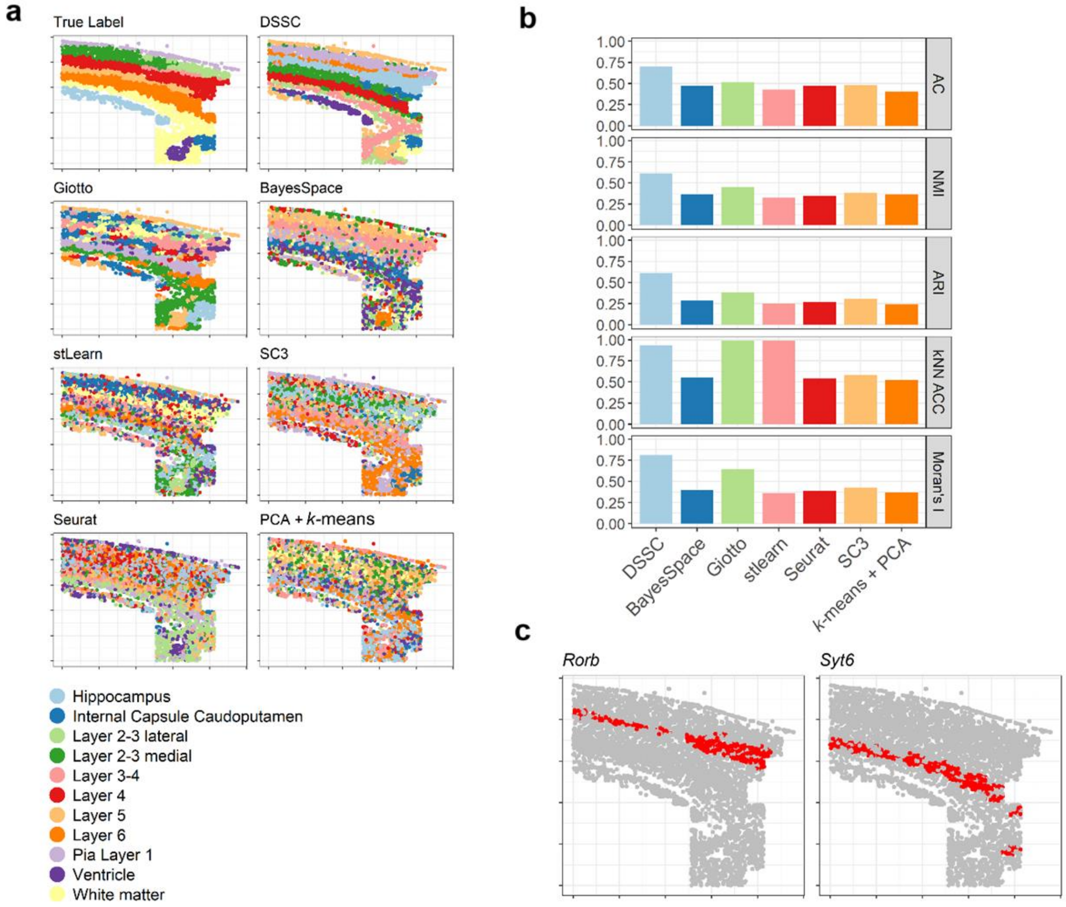


Figure 4.3 Results of osmFISH dataset. (a) predicted labels; (b) clustering performance; and (c) marker genes used for DSSC.

We then test all the methods on the spatialLIBD datasets (see **Figure 4.4**). The marker genes used in this dataset are *PCP4* and *MOBP* (see **Figure 4.4c**) for layer 5 and WM respectively from the paper of spatialLIBD. These genes show strong spatial dependencies. So, they can be used to guide the clustering process. **Figure 4.4a** shows that DSSC is the only method that can identify 5 layers in sample 151673. Some other spatial-based methods, such as SpaGCN, and BayesSpace, cluster some cells in clumps, not in layers. **Figure 4.4b** shows that DSSC outperforms all the competing methods in the 12 spatialLIBD samples in both clustering performance and PLSH. Spatial-based methods have overall better performance than the traditional scRNA-seq clustering methods, revealing the benefits from using the spatial information. BayesSpace has the second-best performance in this dataset since it can recognize the spatial neighbors for each cell in this dataset. The statistical tests of the clustering performance between DSSC and the competing methods are shown in **Appendix Table D.4**.

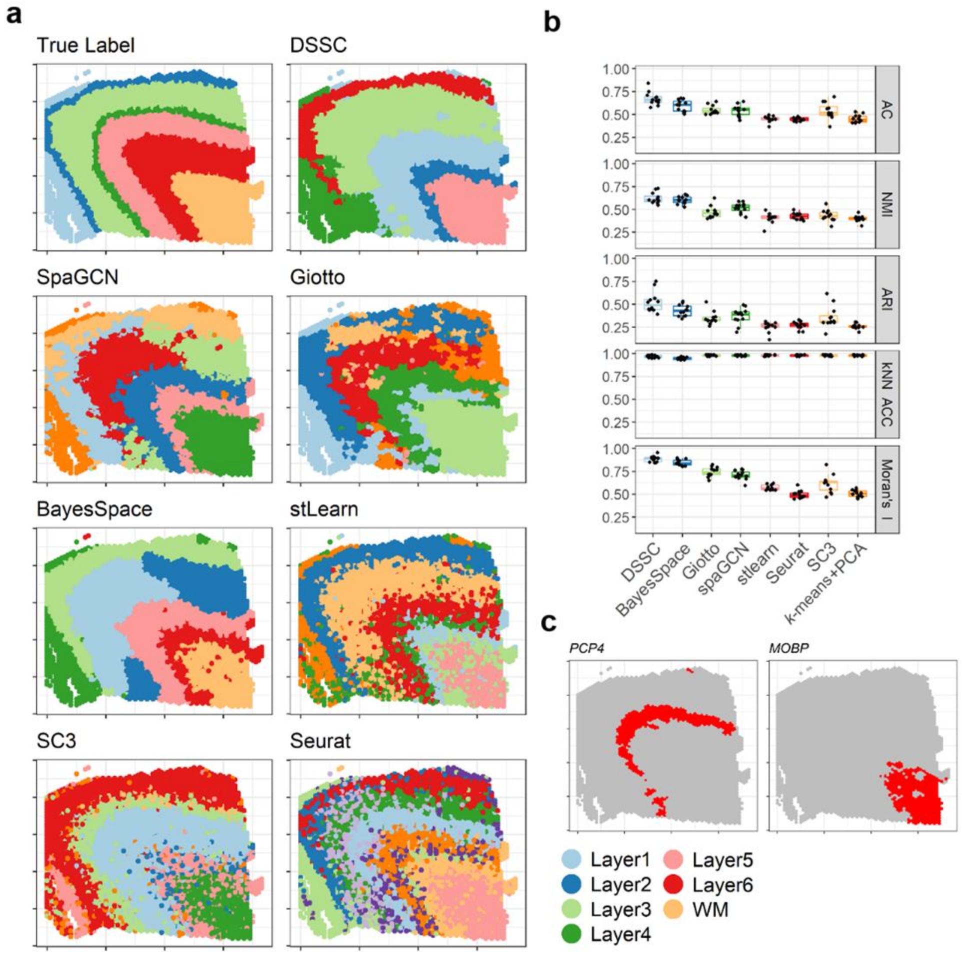
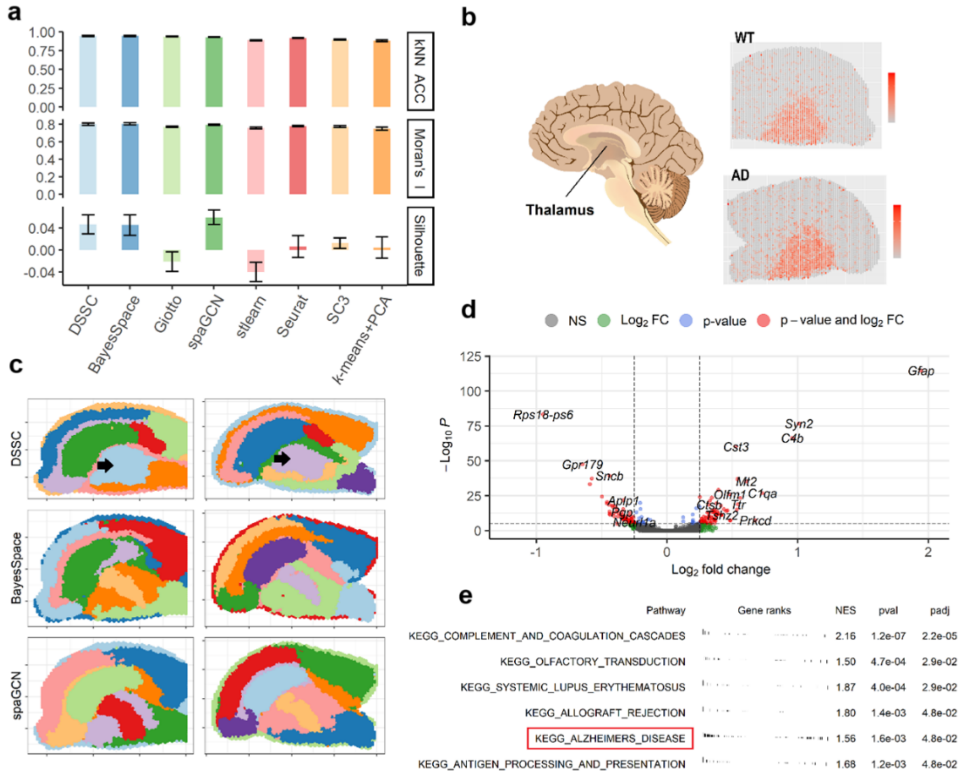


Figure 4.4 Results of spatialLIBD datasets. (a) visualization of the predicted label for sample 151673; (b) the clustering performance of the 12 samples; and (c) the marker gene used in this experiment.

We then apply DSSC on the 10xMBAD dataset (see **Figure 4.5**). Since this dataset has no true labels, we use silhouette score (SS) to evaluate the clustering performance. We find that all the methods have similar predicted labels' spatial heterogeneity on this dataset (see **Figure 4.5a**). DSSC, BayesSpace, and SpaGCN have higher SS than other methods. To further prove the accuracy of clustering of DSSC, we identify the cluster of thalamus in a wild-type (WT) sample and an Alzheimer's Disease (AD) sample by a marker gene *Tcf7l2* (see **Figure 4.5b**)(Lipiec et al., 2020) and then perform a different expression analysis (DE) between the two groups of cells. We select thalamus since it has been widely demonstrated to be associated with the memory and cognition loss during AD (Pardilla-Delgado et al., 2021; Van De Mortel, Thomas, Van Wingen, & Initiative, 2021). BayesSpace and SpaGCN fail to identify the region of thalamus in the corresponding WT and AD samples (see **Figure 4.5c**). The DE results are shown in **Figure 4.5d**. Many genes that overexpress in the AD group have been proved by previous studies. For example, *Olfm1* has been shown as a potential neuroprotective agent in Alzheimer's disease (Takahama, Nakaya, & Tomarev, 2014); *Cst3* has contributions in increasing the neuronal vulnerability and impaired neuronal ability to prevent neurodegeneration (Kaur & Levy, 2012); *Syn2* is related to the onset and progression of Alzheimer's disease (Kumar & Reddy, 2020). As a result, in the pathway analysis of the KEGG geneset from the DE results (see **Figure 4.5e**), the Alzheimer's disease pathway is significantly enriched in the thalamus of the AD sample. Another significant pathway, olfactory transduction, is also shown to be associated with AD from the previous studies (Zou, Lu, Liu, Zhang, & Zhou, 2016). Spliceosome is also demonstrated to be altered in the Alzheimer transcriptomes (Koch, 2018), which is significantly down-regulated in the AD sample. These downstream analyses further consolidate the clustering results of DSSC. The statistical tests of the clustering performance (SS) between DSSC and the competing methods are shown in **Appendix Table D.5**.



Pathway	Gene ranks	NES	pval	padj
KEGG_COMPLEMENT_AND_COAGULATION_CASCADES		2.16	1.2e-07	2.2e-05
KEGG_OLFACTORY_TRANSDUCTION		1.50	4.7e-04	2.9e-02
KEGG_SYSTEMIC_LUPUS_ERYTHEMATOSUS		1.87	4.0e-04	2.9e-02
KEGG_ALLOGRAFT_REJECTION		1.80	1.4e-03	4.8e-02
KEGG_ALZHEIMERS_DISEASE		1.56	1.6e-03	4.8e-02
KEGG_ANTIGEN_PROCESSING_AND_PRESENTATION		1.68	1.2e-03	4.8e-02

Figure 5 Results of 10xMBAD datasets. (a) clustering performance (without true labels); (b) a cartoon of brain showing the position of thalamus (from www.flintrehab.com) and the expression of a marker gene, Tcf7l2, for thalamus in a WT and an AD sample; (c) predicted labels for a wild type sample and an Alzheimer's disease sample from DSSC, BayesSpace, and SpaGCN; the black arrows indicate the thalamus regions; (d) volcano plot from the differential expression analysis (DE) between the cells in thalamus from the wild type and the Alzheimer's disease samples; (e) KEGG pathway analysis from the DE results in panel D. The pathway of Alzheimer's disease is highlighted by the red box.

4.2.3 Model test

We test three parameters in DSSC: 1) the number of constraints (ML and CL respectively); 2) the parameter that controls the clustering loss (gamma); 3) the number of neighbors in the *k*NN graph for GAT layers on the 12 spatialLIBD datasets (see **Figure 4.6a**). We find that when the constraint number is 0 (no constraints) or 6000 (too many constraints), the performance of DSSC becomes unstable. A suitable number of constraints (here we suggest setting the constraint number around the cell number) will not only improve the clustering performance but also makes the model more stable. Compared to the model without clustering loss (gamma=0), DSSC's performance is improved when gamma is 0.01. However, a too high gamma (>1) will seriously impact the model's performance. When the numbers of neighbors are higher than 10, DSSC's performance is not sensitive to them. However, a model without considering neighbors (K=0) has much lower performance revealing the contributions from using the spatial information in clustering analysis. The results of the statistical tests of the parameter tuning experiments are in **Appendix Tables D.6, D.7, and D.8**. We then test DSSC on the simulated datasets with incremental numbers of cells (see **Figure 4.6b**). We find that DSSC has a linearly ascending running time with the increased cell numbers. Thus, it can be easily used for analyzing large datasets. All experiments here are performed on the NVIDIA Tesla P100 with 16Gb memory.

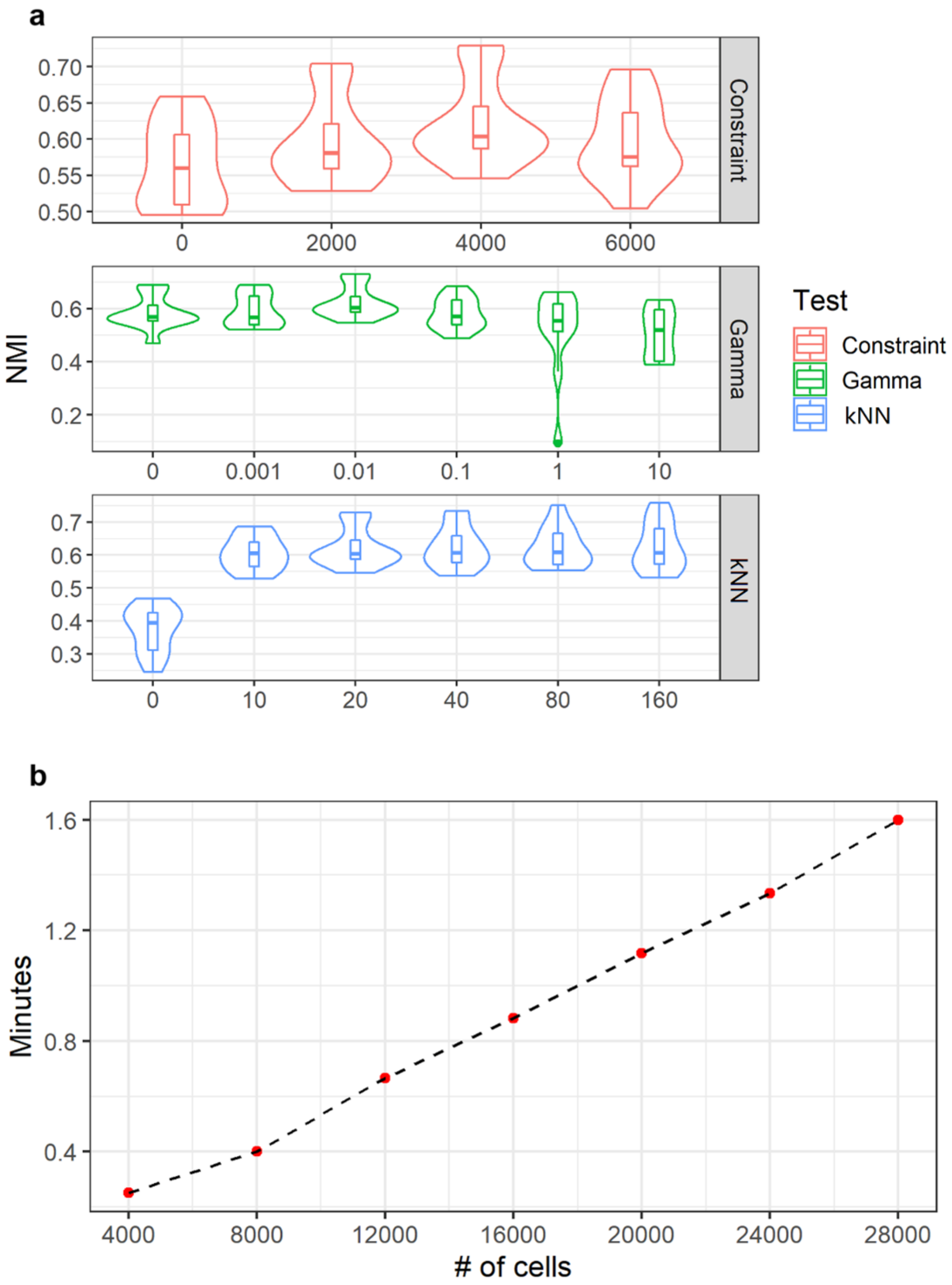


Figure 4.6 Parameter tuning of DSSC. (a) Parameter tuning on the 12 spatialLIBD datasets and (b) running time test on the simulated data with incremental cell numbers.

4.3 Discussion

In this chapter, we have introduced a novel deep learning approach, DSSC, for clustering sp-scRNA-seq data. DSSC utilizes a denoising graphical autoencoder to learn a nonlinear representation of data. Spatial information is integrated into the clustering approach in two ways: 1) constraints from marker genes; and 2) GAT encoders. To our knowledge, DSSC is the first model that can encoder the information from both spatial coordinates and marker genes for guiding the clustering. More broadly, DSSC is a flexible model in which its reconstruction loss function can be switched depending on the data structure. The available reconstruction loss includes ZINB loss, NB loss, and MSE loss to deal with various scenarios. In this study, DSSC has been tested on both simulated and real datasets. The aim of our experiments is to test the robustness of DSSC's clustering performance over the data with different cell type spatial organization and cell type spatial dependency. The evaluation has been conducted regarding two aspects, clustering performance, and space heterogeneity. Our results show that DSSC outperforms the state-of-art methods over different datasets.

Recently, a new general-purpose density estimator has been introduced by employing a symmetrical and paired generative adversarial network (GAN) architecture (Liu, Xu, Jiang, & Wong, 2021). Adopting this GAN architecture, a new method scDEC enables simultaneous learning of latent features and cell clustering and shows its superiority over competing methods in scATAC-seq analysis (Liu, Chen, Jiang, & Wong, 2021). If spatial information could be accommodated in this GAN architecture, we may expect similar promising improvement in analysis of sp-scRNA-seq data. We leave such exploration to future work.

One limit of the current model is its compatibility with the datasets with low spatial dependency. DSSC employs the spatial information of cells to boost the clustering performance, while not all tissue types have a high spatial dependency. Besides, for approaches like 10x Visium, our model is dependent on the assumption that all the cells in one spot are in the same cell type. In the future investigation, this issue can be solved by doing the decomposition of spots. The latent representation of DSSC can be used for many downstream analyses, such as cell-to-cell communication and trajectory analysis.

4.4 Methods and Materials

4.4.1 Denoising autoencoder

The autoencoder is a neural network for learning a nonlinear representation of data (Hinton & Salakhutdinov, 2006). It receives corrupted data with artificial noises and reconstructs the original data (Vincent et al., 2008). It is able to learn a robust latent representation for noisy data. We use the denoising autoencoder for the highly noisy count data of cells. Let's denote the preprocessed counts data as X and the corrupted data as X_c, formally:

$$X_c = X + \sigma * n \tag{4.1}$$

where n is the artificial noise in standard Gaussian distribution (with mean=0 and variance=1), and σ controls the weights of n. We set σ as 0.1.

Next, we use an autoencoder to reduce the dimension of count data. Encoders (*E*) are graphical attention networks (GAT) layers and decoders (*D*) are fully connected neural networks. Denoting the latent space as Z and the learnable weights of encoder as w, the encoder can be shown as $Z = E_w(X_c)$. The GAT layers in E can be formalized as:

$$X_i = \begin{cases} ELU(BatchNorm(GAT_i^{(K)}(X_c, A))) \qquad if\ i = 1 \\ ELU(BatchNorm(GAT_i^{(K)}(ELU(X_{i-1}), A))) \qquad if 1 < i < L \\ GAT_i^{(K)}(ELU(X_{i-1}), A) \qquad if\ i = L \end{cases} \tag{4.2}$$

where X_i is the output of the i*th* layer. $GAT_i^{(K)}$ is the ith GAT layer with K heads. *L* is the total layers of encoder. A is the adjacent matrix of a kNN graph G built based on the spatial coordinates of cells. Specifically, the distance between two cells i and j is measured by Euclidean distance:

$$M_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4.3}$$

where x and y indicate the coordinates of cells i and j in a two-dimensional physical space. Then A_{ij} $(i, j \ \epsilon \ 1, 2, 3, \dots, N)$ is built by:

$$A_{ij} = \begin{cases} 1,\ if\ i\ is\ the\ K\ nearest\ neighbor\ of\ j\ on\ the\ physical\ space \\ 0,\ otherwise \end{cases} \tag{4.4}$$

A is then normalized by $\tilde{A} = \bar{A} \cdot A \cdot \bar{A}$, where $\tilde{A}$ is the normalized graph, $\bar{A}$ is $diag(power\left(\sum_j^N A_j, -0.5\right))$ and $(\cdot)$ means dot product. Then $\tilde{A}$ is used as the input for the GAT encoder. In this study, we set the number of heads as 3. The decoder is $X' = D_{w\prime}(Z)$, where w' are the learnable weights for the decoder and X' is the reconstructed counts from the decoder. The ELu activation function

(Nair & Hinton, 2010) and batch normalization are used for all the hidden layers in the encoder and decoder except the bottleneck layer. In the default setting, we use two layers of encoder and decoder. The default bottleneck layer is set as 32.

We employ a zero-inflated negative binomial (ZINB) model in the reconstruction loss function to characterize the zero-inflated and over-dispersed count data (Tian et al., 2019). Note, the raw count data, not the normalized data, is used in the ZINB model (Eraslan et al., 2019; Lopez et al., 2018; Tian et al., 2019). Let X_{ij} be the count for cell i and gene j in the raw count matrix. The NB distributions are parameterized by μ_{ij} and θ_{ij} as means and dispersions respectively. Formally:

$$NB\left(X_{ij}\middle|\mu_{ij},\theta_{ij}\right)=\frac{\Gamma\left(X_{ij}+\theta_{ij}\right)}{X_{ij}!\Gamma\left(\theta_{ij}\right)}\left(\frac{\theta_{ij}}{\theta_{ij}+\mu_{ij}}\right)^{\vartheta_{ij}}\left(\frac{\theta_{ij}}{\theta_{ij}+\mu_{ij}}\right)^{X_{ij}} \tag{4.5}$$

Then, ZINB distribution is parameterized by the negative binomial and an additional coefficient π_{ij} for the probability of dropout events (zero mass):

$$ZINB\left(X_{ij}\middle|\mu_{ij},\theta_{ij},\pi_{ij}\right)=\pi_{ij}\delta_0\left(X_{ij}\right)+(1-\pi_{ij})NB(X_{ij}|\mu_{ij},\theta_{ij}) \tag{4.6}$$

The loss function of ZINB-based autoencoder for the count data is defined as:

$$L_{ZINB}=\sum_{ij}-\log\left(ZINB(X_{ij}\,|\mu_{ij},\theta_{ij}\,,\pi_{ij})\right) \tag{4.7}$$

We use independent fully connected layers to estimate these parameters in ZINB loss functions. We add three independent fully connected layers M, Θ, and Π after the last hidden layer of the decoder which outputs the reconstructed matrix X'. The parameter layers are defined as:

$$M = diag(s_i) \times \exp(w_\mu X'); \tag{4.8}$$

$$\Theta = \exp(w_\theta X'); \tag{4.9}$$

$$\Pi = \exp(w_\pi X'); \tag{4.10}$$

where M, Θ, and Π are the matrix of estimated mean, dispersion, and drop-out probability for the ZINB loss of count data. w_μ, w_θ, and w_π are the learnable weights for them, respectively. The size factor s_i for the cell i was calculated in the preprocessing step.

The sizes of layers are set to (128, 32) for the GAT encoder and (32, 128) for the fully connected decoder.

4.4.2 Deep embedded clustering

Our model has two learning stages, a pretraining stage and a clustering stage. In the pretraining stage, we only train the autoencoder without considering the clustering loss and the constraint loss (see details below). Then, in the clustering stage, we simultaneously optimize the autoencoder and the clustering results. We perform unsupervised clustering on the latent space of the autoencoder (Xie et al., 2016). Our autoencoder transfers the input matrix to a low dimensional space *Z*. The clustering loss is defined as the Kullback-Leibler (KL) divergence between the soft label distribution *Q'* and the derived target distribution *P'*:

$$L_{Clustering} = KL(P' \parallel Q') = \sum_i \sum_k p'_{ik} \log \frac{p'_{ik}}{q'_{ik}} \tag{4.11}$$

where the soft label q'_{ik} measures the similarity between z_i and cluster center μ_k by Student's t-kernel (Maaten & Hinton, 2008). The cluster center μ_k is initialized by applying a *k*-means on the bottleneck layer from the pretraining stage, and then updated per batch in the clustering stage. Formally, q'_{ik} is defined as:

$$q'_{ik} = \frac{(1+\|z_i - \mu_k\|^2)^{-1}}{\sum_{k'}(1+\|z_i - \mu_{k'}\|^2)^{-1}} \tag{4.12}$$

The target distribution P' which emphasizes the more certain assignments is derived from *Q'*. Formally p'_{ik} is defined as:

$$p'_{ik} = \frac{q'^2_{ik}/\sum_i q'_{ik}}{\sum_{k'}(q'^2_{ik'}/\sum_i q'_{ik'})} \tag{4.13}$$

During the training process, Q' and clustering loss are calculated per batch and P' is updated per epoch. This clustering loss will improve the initial estimate (from *k*-means) in each iteration by learning from the high-confident cell assignments, which in turn helps to improve the low-confident ones (Xie et al., 2016).

4.4.3 Autoencoder with pairwise constraints

Based on the autoencoder architecture, we add pairwise constraints of cells (Tian, Zhang, Lin, Wei, & Hakonarson, 2021a) on the latent space according to the expression of the marker genes. Similar to scDCC (Tian et al., 2021a), we employ the must-link constraints which pull two cells to have similar soft labels if they have similar expression patterns of one or more marker genes, and cannot-link constraints which encourage two cells to have different soft labels if they have different expression patterns of one or more marker genes.

Constraints are built by six steps, considering both the spatial coordinates and the gene expression of the cells: 1) select the marker genes from literatures;

2) for each marker, say gene A, smooth the expression of A by averaging the normalized count data of the k (k is defined according to the technology, we set it as 6 in this study) spatial neighbors of each cells; 3) define the cells with the top 5% (cutoff1) expression of A as high, otherwise as low; 4) collect the cells as the confident cells if more than half (cutoff2) of its neighbors (and itself) have the high smoothed expression of A; 5) repeat step 2-4 for all the marker genes; 6) since each marker gene represents a cell type (or a layer in cortex), we connect two confident cells by a must-link if they are selected by the markers for the same cell type (or layer); otherwise, we connect two confident cells by a cannot-link if they are selected by the markers for different cell types (or layers). It is noted that there is a tradeoff between the coverage and the reliability of constraints. A higher cutoff will decrease the coverage of constraints but also reduce the false positive links. We denote the constraints sampled here as the pool of constraints.

The must-link and cannot-link constraints loss are defined as:

$$L_{ml} = \sum_{(i,j)\in ML} log \sum q_i \times q_j \tag{4.14}$$

$$L_{cl} = \sum_{(i,j)\in CL} \log\,(1 - \sum q_i \times q_j) \tag{4.15}$$

where q is the soft labels described in the clustering section. Must-links and cannot-links are used for training the model alternately and are updated (resampled) during the training. The number of constraints can be set according to the cell numbers. For example, for a dataset with 4000 cells, we sample 4000 must-links and cannot-links, respectively.

Combining the pairwise constraint loss, reconstruction loss, and clustering loss, the total loss of the DSSC is:

$$L = L_{ZINB} + \gamma * L_{Clustering} + \beta * L_{ml} + \lambda * L_{cl} \qquad (4.16)$$

where γ, β, and λ are the coefficients for the clustering loss, must-link loss, and cannot-link loss respectively. In the experiments of this study, γ is set to 0.01, β and λ are set to 0.1 and 1 respectively (see parameter tuning in the result section).

4.4.4 Model implementation

This model is implemented in Python3 using PyTorch (Paszke et al., 2017). Adam with AMSGrad variant (Kingma & Ba, 2014; Reddi et al., 2018) with an initial learning rate = 0.001 is used for the pretraining stage and the clustering stage. The kNN graph is calculated by the “kneighbors_graph” function from the scikit-learn package. The top 2000 HVGs are selected to train the model. We pretrain the autoencoders for 200 epochs before entering the clustering stage. In the beginning of the clustering stage, we initialize *K* centroids by the *k*-means algorithm. During the clustering stage, reconstruction loss and clustering loss are optimized first. Then, constraint losses are optimized with reconstruction loss. ML and CL losses are optimized alternately. The centroids are also continuously updated by the learning process. Before each epoch, constraints are randomly sampled from the constraint pools. The soft label distribution *Q’* is calculated in each batch and the derived target distribution *P’* is updated after each epoch. The convergence threshold for the clustering stage is that less than 0.1% of labels are changed per epoch. The marker genes used in this study are from the original paper of the spatialLIBD datasets (Maynard et al., 2021), including *PCP4*, *MOBP*, *FABP7*, *AQP4, CARTPT, KRT17* and so forth. More markers can be added if necessary. It is noted that we test the Moran's I and check the expression pattern of each marker before using it (See **Appendix F** for details). If a marker has very low spatial dependency in a dataset, we exclude it for building constraints. For the osmFISH dataset with only 33 genes, we just use the genes with the highest spatial dependency (Moran' I) as the markers. All experiments of DSSC in this study are conducted on NVIDIA Tesla P100 with 16Gb memory.

4.4.5 Marker and gene selection

Before running the autoencoder model, we use Moran's I statistic (Miller et al., 2021; Moran, 1950) to measure the gene spatial heterogeneity. I_k^{gene} stands for the Moran's I of gene k, which is defined as:

$$I_k^{gene} = \frac{N}{\sum_{i=1}^{N}\sum_{j=1}^{N} A} \cdot \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{4.17}$$

where $\underline{x}$ is the mean value of the normalized counts of the gene K over all cells. A is the kNN graph from spatial information of cells. Marker genes with low Moran' I will not be used to build constraints. It is noted that gene filtering has a tiny influence on the performance of the osmFish dataset since it only has 33 genes. These genes are all selected by the researchers so all of them are important for all or a part of cells in the tissue. In our experiments, because of the low feature number, we only selected 30 HVGs out of 33 genes. On the other hand, the sequencing-based methods profile the whole transcriptome (>20000 genes). Many genes are not informative for clustering and even mislead the clustering. So, feature selection is essential for these datasets. In our experiments, we select the top 2000 highly variable genes (HVGs) for training DSSC. An optional feature selection approach is to use the genes with the top Moran’s I.

4.4.6 Evaluation metrics for clustering performance

Adjusted Rand Index (ARI) (Hubert & Arabie, 1985), Normalized Mutual Information (NMI)(Alexander & Joydeep, 2003), and Clustering Accuracy (AC) are used as metrics to evaluate the performance of different methods.

Adjusted Rand Index measures the agreements between two sets U and G. Assuming a is the number of pairs of two cells in the same group in both U and G; b is the number of pairs of two cells in different groups in both U and G; c is the number of pairs of two cells in the same group in U but in different groups in G; and d is the number of pairs of two cells in different groups in U, but in the same group in G. The ARI is defined as:

$$ARI = \frac{\binom{n}{2}(a+d)-[(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}-[(a+b)(a+c)+(c+d)(b+d)]} \tag{4.18}$$

Let U = {U1, U2, …, C$_{tu}$} and G = {G1, G2, …, G$_{tg}$} be the predicted and ground truth labels on a dataset with n cells. NMI is defined as:

$$NMI = \frac{I(U,G)}{\max\{H(U),H(V)\}} \tag{4.19}$$

where I(U,G) represents the mutual information between U and G and is defined as:

$$I(U,G) = \sum_{p=1}^{tu}\sum_{q=1}^{tg}|U_p \cap G_q|\log\frac{n|U_P \cap G_q|}{|U_p|\times|G_q|} \tag{4.20}$$

and H(U) and H(G) are the entropies:

$$H(U) = -\sum_{p=1}^{tu}|U_p|\log\frac{|U_p|}{n} \tag{4.21}$$

$$H(G) = -\sum_{p=1}^{tg}|G_p|\log\frac{|G_p|}{n} \tag{4.22}$$

AC is defined as the best matching between predicted and true clusters, which is given as:

$$AC = \max_{m}\sum_{i=1}^{n} 1\frac{\{\widehat{l_\imath} = m(l_i)\}}{n} \tag{4.23}$$

where $\widehat{l_\imath}$ are the true labels and l_i are the predicted labels from clustering algorithms. n is the number of cells and m is the number of all possible one-to-one mapping between $\widehat{l_\imath}$ and l_i. The best mapping is found by the Hungarian algorithm (Kuhn, 1955).

The silhouette score (SS) is used to measure the clustering performance without labels. It compares how similar a cell is to its own cluster compared to other clusters. The silhouette score ranges from −1 to +1, where a high value indicates a better clustering. Let’s denote the silhouette score of cell i as S_i, so we have:

$$S_i = \begin{cases} 1 - \frac{a_i}{b_i} & if\ a_i < b_i \\ 0 & if\ a_i = b_i \\ \frac{b_i}{a_i} - 1 & if\ a_i > b_i \end{cases} \tag{4.24}$$

where a_i stands for how well a cell I is assigned to its cluster based on the distance between this cell and all other cells in its cluster; b_i stands for the smallest mean distance of the cell I to the cells in any other clusters. Then we use the mean value of S_i over all the cells as the SS for a dataset.

4.4.7 Evaluation metrics for spatial heterogeneity and concentration

kNN accuracy measures the consistency of the labels between each cell and its spatial neighbors. It is defined as:

$$A_{KNN} = \frac{\sum_{i=1}^{N} y_i = \hat{y_i}}{N} \tag{4.25}$$

where y_i is the predicted label of cell i by clustering algorithms and $\hat{y}$ is the major label of its neighbors (K=20) on the physical space. We also employ a variant of Moran's I (Moran, 1950) to measure the cell type spatial concentration. Let I^{label} be the I score for the predicted labels $(y_1, y_2, y_3, \dots, y_N)$ defined as:

$$I^{label} = \frac{N}{\sum_{i=1}^{N} \sum_{j=1}^{N} A} \cdot \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} B_{ij}}{N} \tag{4.26}$$

where B_{ij} of cell i and j is defined as:

$$B_{ij} = \begin{cases} 1, if\ y_i = y_j \\ 0, otherwise \end{cases} \tag{4.27}$$

, and A is the kNN graph (with k=20) from spatial information of cells. The I^{label} measures the degree that the physically neighboring cells have the same label. Both metrics range from 0 to 1.

4.4.8 Data simulation

In order to test the model's performance to integrate spatial information for clustering, we simulate the single-cell RNA-seq data by Splatter package in R (Zappia, Phipson, & Oshlack, 2017). The parameters for scRNA-seq data simulation are estimated from a real scRNA-seq dataset (https://support.10xgenomics.com/spatial-gene-expression/datasets) and the parameter of clustering signal (de.scale) is fixed as 0.4. Besides simulating the count data, we place each cell on a 2D space with a coordinate (x,y). The physical space and coordinates are extracted from two real datasets (osmFISH and 151507 from spatialLIBD). The regions (domains) on the physical space in the real datasets are provided by the authors. Specifically, let's denote the spot number in a layer k (from true label) as n_k and the total layer number as K. During the simulation, for a layer k, we use splatter to simulate n_k cells and randomly assign these cells to the spatial coordinates of the spots in this layer. We do this for all K layers. So, the cell number in the simulated datasets should be the same as the spot number in the real dataset. Then, we perturb the spatial coordinate of 10%, 15%, and 20% of cells to control the cell type spatial dependency. We also use the spatial coordinates from two datasets (osmFISH (Codeluppi et al., 2018) and spatialLIBD 151507 (Maynard et al., 2021)) to simulate different spatial organizations. Therefore, our simulation experiments can test the robustness of DSSC's performance in the data with different cell type spatial dependencies and cell type spatial organizations. To simulate the constraints from markers, we randomly connect 3000 cells in the same cell type (from the true label) as the must-links. We then perturb the cells in 5% must-links to simulate the real accuracy (about 95%). Similarly, we randomly connect 3000 cells in the different cell types as the cannot-links.

4.4.9 Real datasets

We use data from three studies including 25 sp-scRNA-seq datasets in this study. The first dataset was measured by the osmFISH technology (Codeluppi et al., 2018), and the other two datasets were sequenced by the 10x Visium technology and provided by spatialLIBD (Pardo et al., 2022) and 10x Genomics website, respectively. Specifically, the osmFISH dataset of the somatosensory cortex was downloaded from the website of Linnarsson lab (http://linnarssonlab.org/osmFISH/). This dataset contains 33 genes and 4839 cells. We did not implement the feature selection for this dataset as the low dimension of features. All 10x Visium datasets are read by the 'Load10x_Spatial' function and preprocessed by the 'SCTransform' function by Seurat in R. The 10x mouse brain Alzheimer's disease dataset is downloaded from the website (https://www.10xgenomics.com/resources/datasets). This dataset contains 12 sp-scRNA-seq data with 6 wild-type samples and 6 CRND8 APP-overexpressing transgenic (Alzheimer's Disease, AD) samples. The mice brains were sampled in 2.5, 5.7, and 13.2 month of age. Per phenotype per time-point has two replicates resulting in 12 samples in total. The spatialLIBD dataset is downloaded from R package "spatialLIBD" (Pardo et al., 2022). This dataset contains 12 spatial-resolved RNA-seq datasets which can be grouped into three spatial organizations. Specifically, sample 151507-151510 have similar spatial organization, sample 151669-151672 have similar spatial organization, and sample 151673-151676 have similar spatial organization.

4.4.10 Count data preprocessing

The raw count data is preprocessed and normalized by the Python package SCANPY (Wolf et al., 2018). Specifically, the genes with no count are filtered out. The counts of a cell are normalized by a size factor s_i, which is calculated as dividing the library size of that cell by the median of the library size of all cells. In this way, all cells will have the same library size and become comparable. Then, the counts are logarithm transformed and scaled to have unit variances and zero means. The treated count data is used in our denoising autoencoder model. However, we use the raw count matrix to calculate the ZINB loss (Eraslan et al., 2019; Lopez et al., 2018).

4.4.11 Competing methods

For consistency, we use DSSC's data preprocessing and feature selection approaches for all the competing methods. Our competing methods include *k*-means (with PCA) (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html), Seurat (https://github.com/satijalab/seurat) (Butler et al., 2018), SC3 (https://github.com/hemberg-lab/SC3) (Kiselev et al., 2017), BayesSpace (https://github.com/edward130603/BayesSpace) (Zhao et al., 2021), Giotto (https://rubd.github.io/Giotto_site/) (Dries et al., 2021), SpaGCN (https://github.com/jianhuupenn/SpaGCN) and stlearn (https://github.com/BiomedicalMachineLearning/stLearn). For Seurat and Giotto, we adjusted the resolution in the Louvain algorithm for a better K estimation (same or close to the real K). All other parameters in all the competing methods are kept in the default setting or following the settings in the official pipelines. It is noted that the latent dimension of SpaGCN is higher than the feature dimension of osmFISH data. So SpaGCN cannot be used to analyze osmFISH data. For consistency, H&E images are not used for all the methods.

4.4.12 Statistical test

The differences between the clustering performance of DSSC and the competing methods are tested by the one-sided paired t-test.

4.4.13 Software availability

Source code of DSSC is available at GitHub (https://github.com/xianglin226/DSSC).

CONCLUSION

High throughput data generation, both multi-omics and spatial-resolved scRNA-seq, revealed the great demands of scalable computational methods that can take advantages of the multi-dimensional measurements to efficiently improve the downstream analyses, such as the clustering and differential expression analysis. However, the state-of-the-art methods developed for the multi-omics scRNA-seq data clustering and the spatial-resolved scRNA-seq data clustering still have some potential issues which prevent them from achieving good performance and/or scalabilities. Thus, the existing computational methods do not catch up with the rapid changes in technologies and fail to fully fulfil their potential. In my study, I developed two models, scMDC and DSSC, for analyzing multi-omics single-cell data and spatial-resolved single-cell data, respectively. The extensive experiments demonstrated the superior performance of these novel models. Therefore, they represent promising tools for application in
real-world genomic research.

APPENDIX A

EMBEDDINGS FROM SCMDC AND COMPETING METHODS

Figures A.1, A.2, A.3, and A.4 show the U-maps of the embeddings extracted from different models.

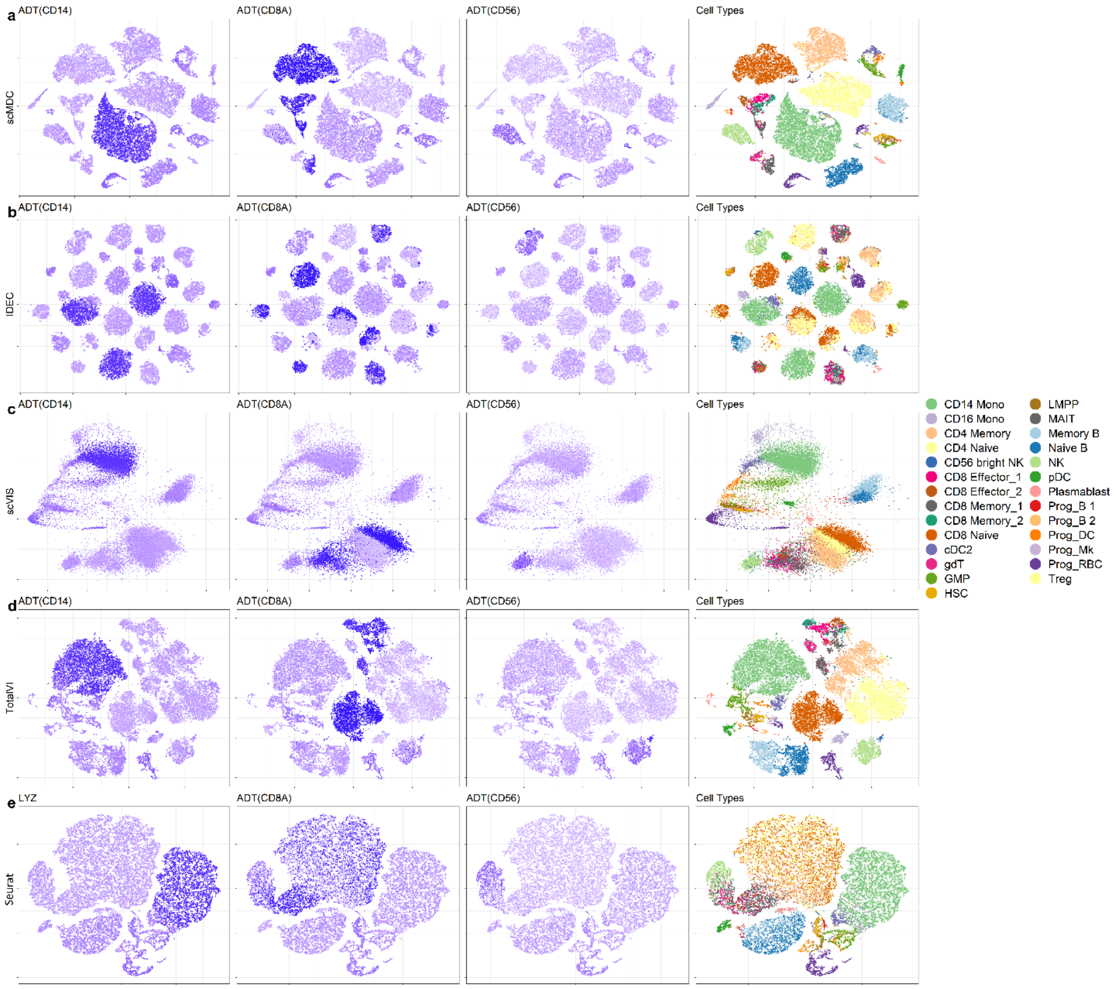


Figure A.1 Low-dimension representation of scMDC and the competing methods on the BMNC dataset. The t-SNE plots of the embeddings from (a) scMDC and four competing methods including (b) IDEC, (c) scVIS, (d) TotalVI, and (e) Seurat are shown in different rows. The first three columns show the expression pattern of ADT CD14, CD8A, and CD56. The last column shows the true labels (cell types) on the latent space of each method.

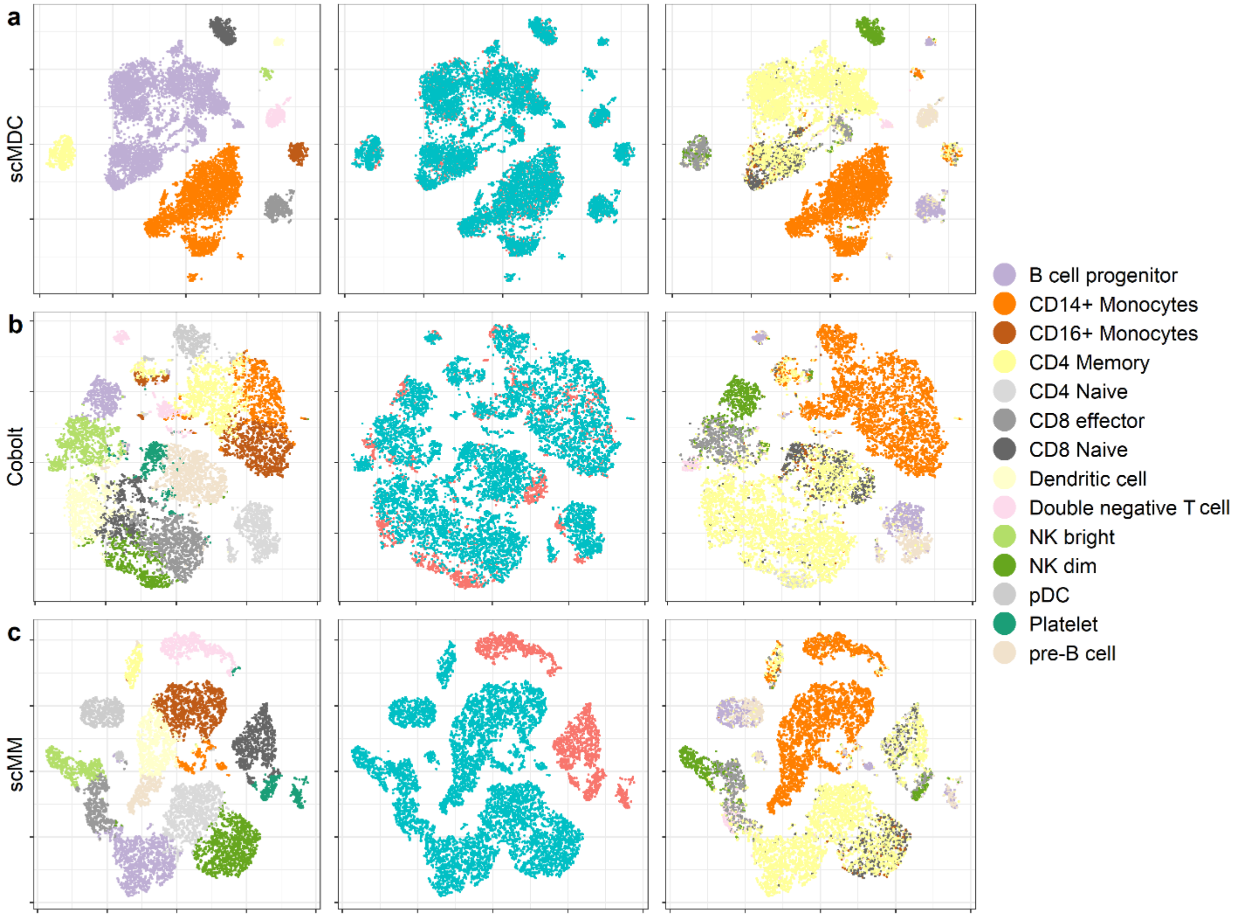


Figure A.2 Low-dimension representation of scMDC and the competing methods on the PBMC13K dataset. The t-SNE plots of the embeddings from (a) scMDC and two competing methods including (b) Cobolt and (c) scMM are shown in different rows. The three columns show the predicted labels, the batch IDs, and the true labels on the latent space of each method.

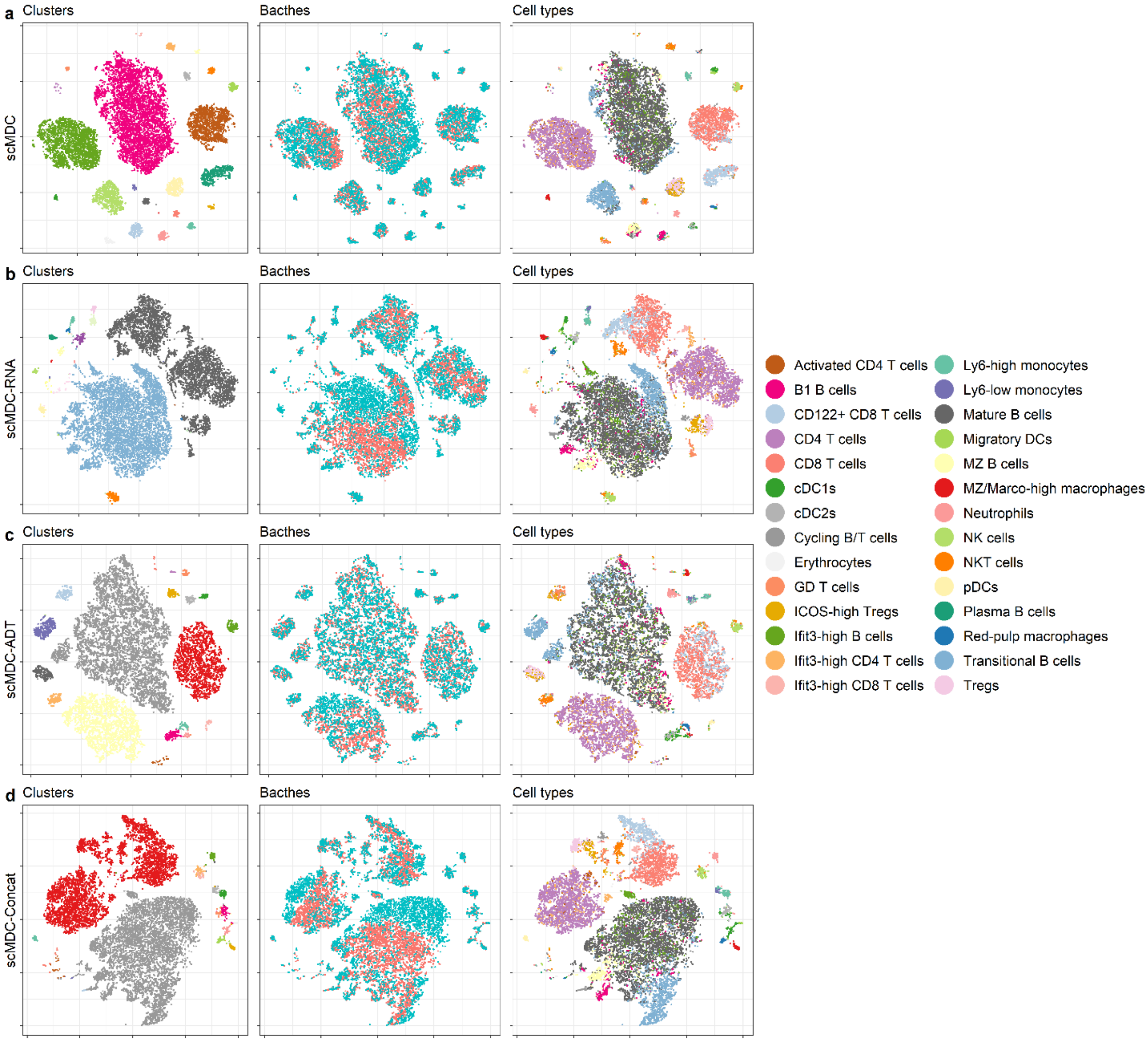


Figure A.3 Low-dimension representation of scMDC and the variant methods on the SLN111 dataset. The t-SNE plots of the embeddings from (a) scMDC and three competing methods including (b) scMDC-RNA, (c) scMDC-ADT, and (d) scMDC-Concat are shown in different rows. The three columns show the predicted labels, the batch IDs, and the true labels on the latent space of each method.

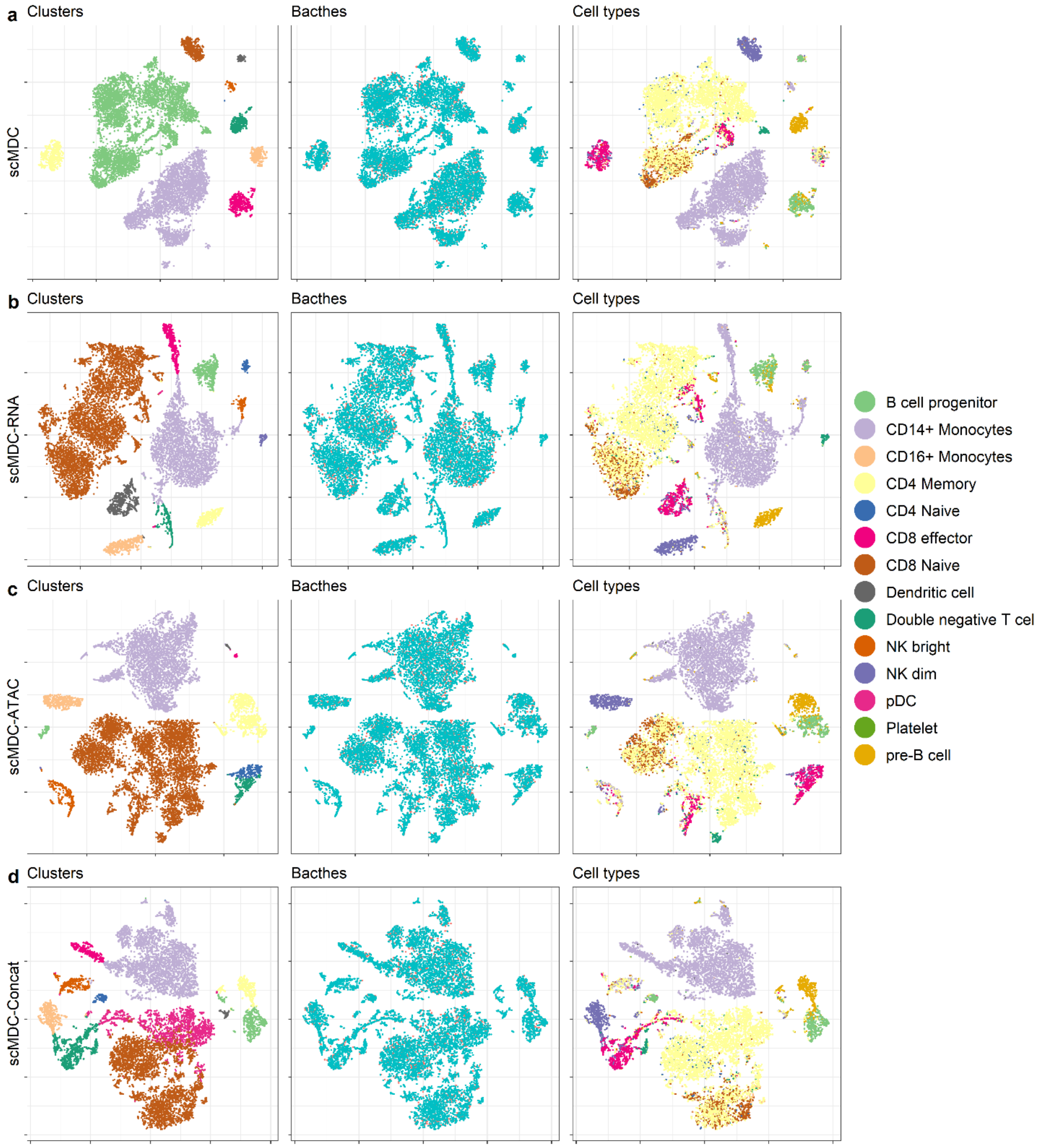


Figure A.4 Low-dimension representation of scMDC and the variant methods on the PBMC13K dataset. The t-SNE plots of the embeddings from (a) scMDC and three competing methods including (b) scMDC-RNA, (c) scMDC-ATAC, and (d) scMDC-Concat are shown in different rows. The three columns show the predicted labels, the batch IDs, and the true labels on the latent space of each method.

APPENDIX B

CLUSTERING PERFORMANCE OF SCMDC AND COMPETING METHODS ON SINGLE-MODAL DATASETS

Figures B.1-8 show the clustering performance of scMDC and competing methods on single-modal data with single and multiple batches.

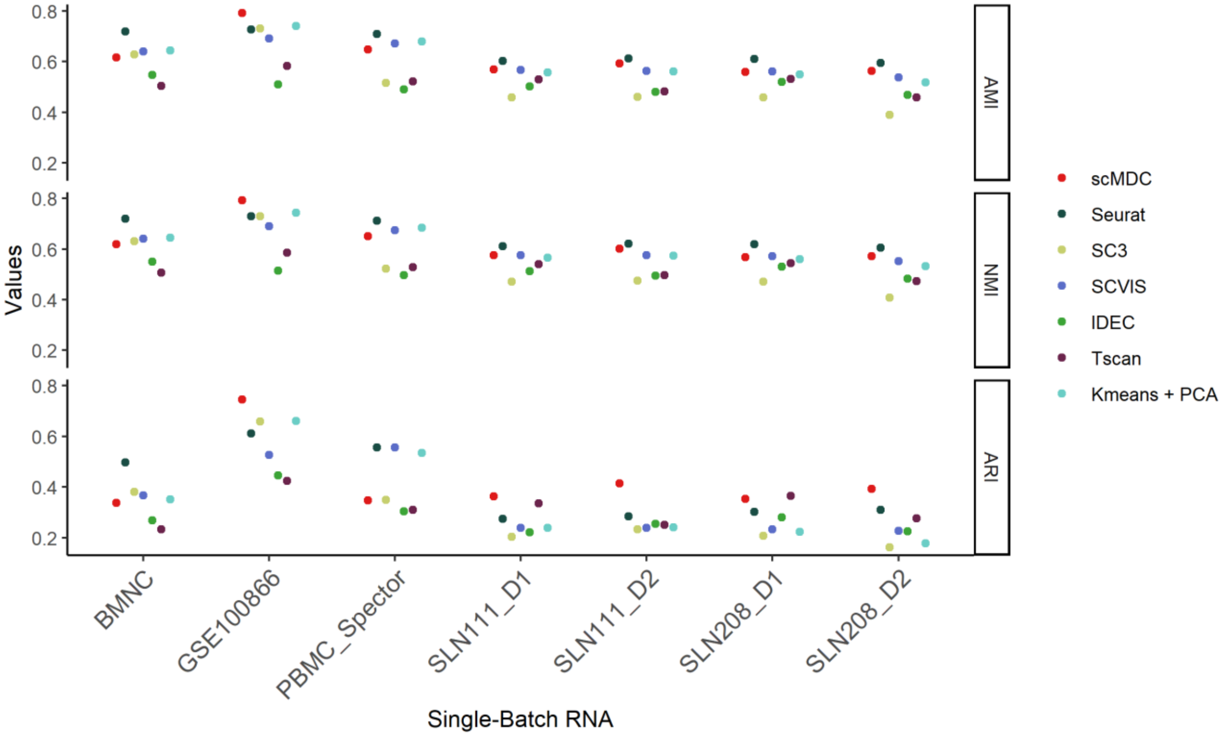


Figure B.1 Clustering performance of scMDC-RNA and six single-modal clustering methods on the single-batch CITE-seq datasets. All methods only take mRNA counts or normalized counts as input. Clustering performance is evaluated by AMI, NMI, and ARI.

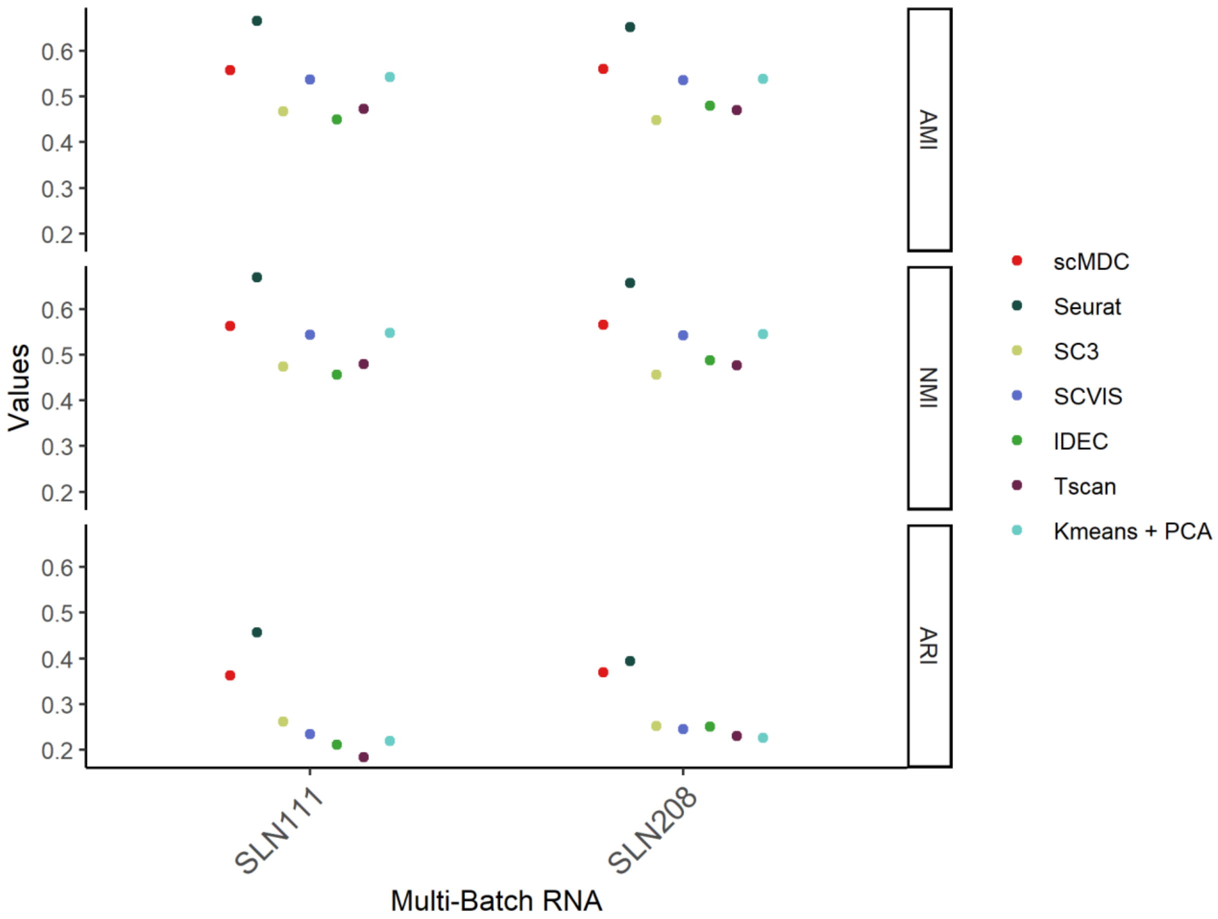


Figure B.2 Clustering performance of scMDC-RNA and six single-modal clustering methods on the multiple-batch CITE-seq datasets. All methods only take mRNA counts or normalized counts as input. Clustering performance is evaluated by AMI, NMI, and ARI.

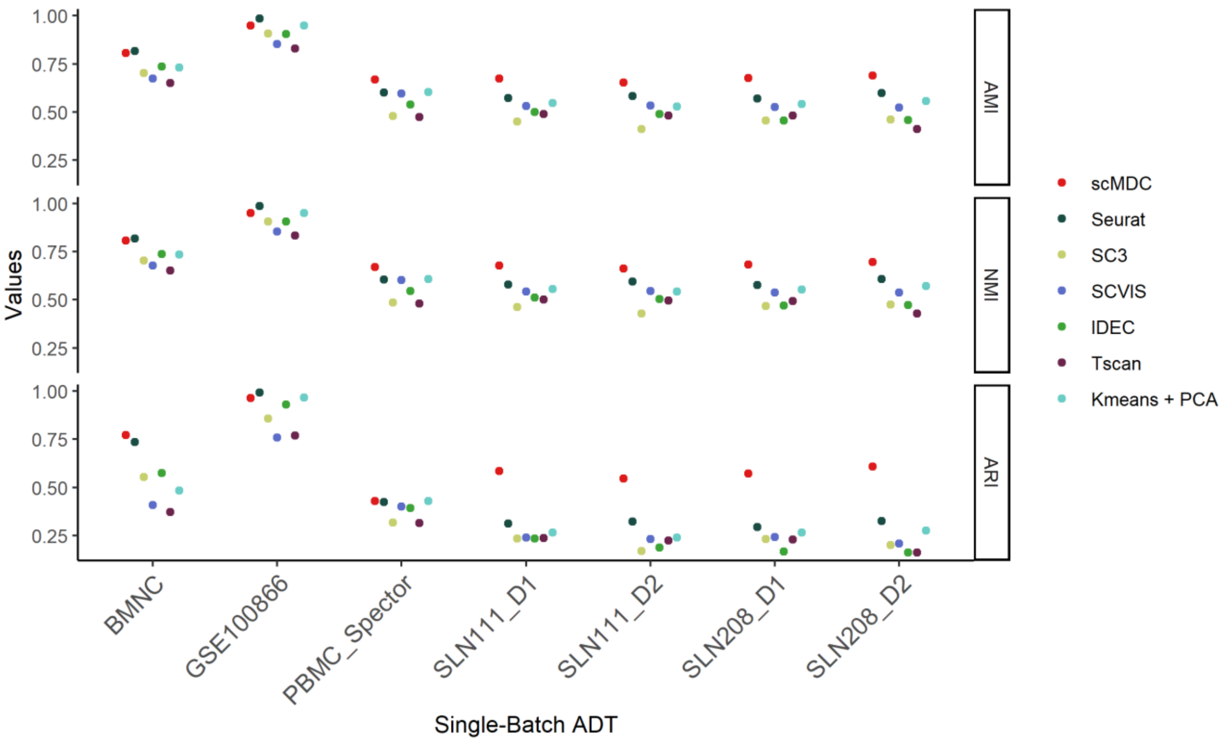


Figure B.3 Clustering performance of scMDC-ADT and six single-modal clustering methods on the single-batch CITE-seq datasets. All methods only take ADT counts or normalized counts as input. Clustering performance is evaluated by AMI, NMI, and ARI.

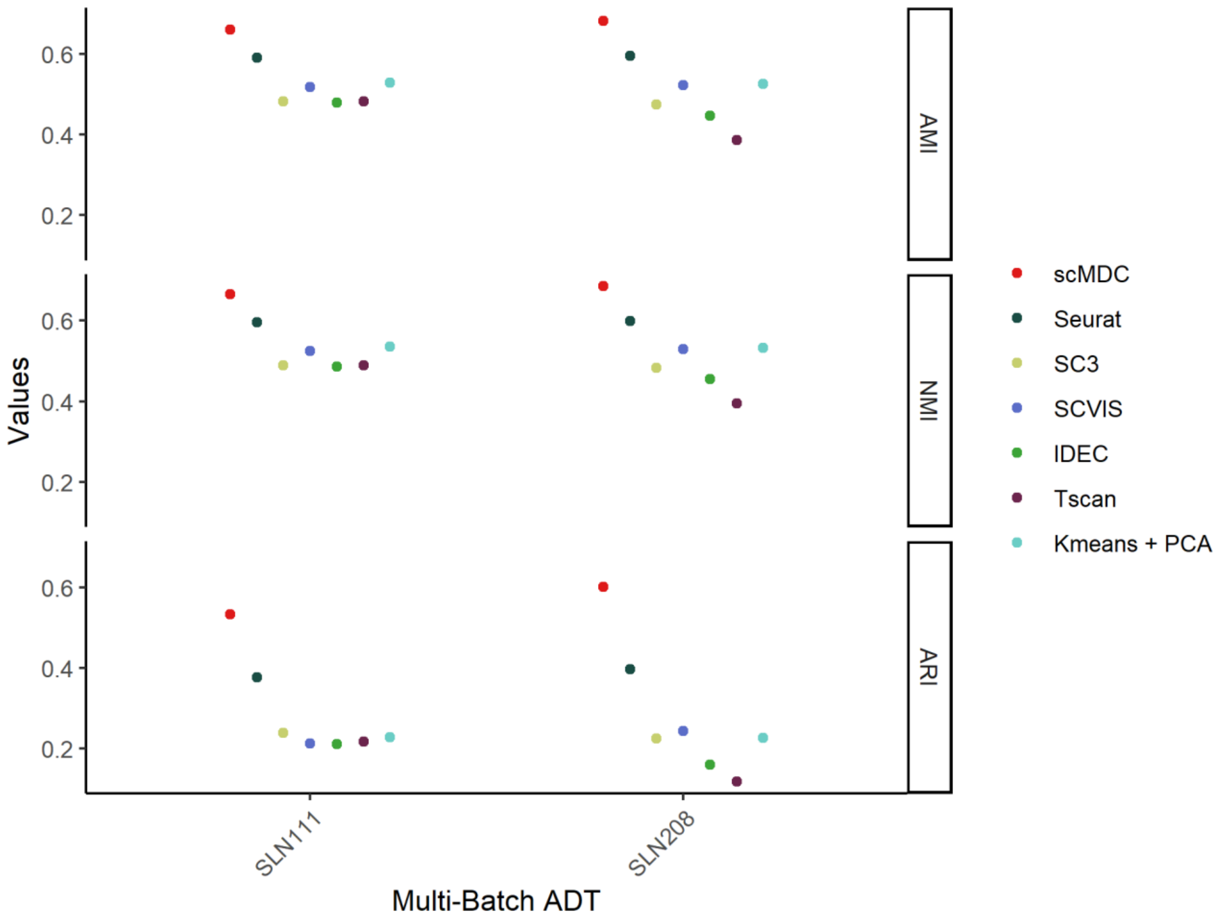


Figure B.4 Clustering performance of scMDC-ADT and six single-modal clustering methods on the multiple-batch CITE-seq datasets. All methods only take ADT counts or normalized counts as input. Clustering performance is evaluated by AMI, NMI, and ARI.

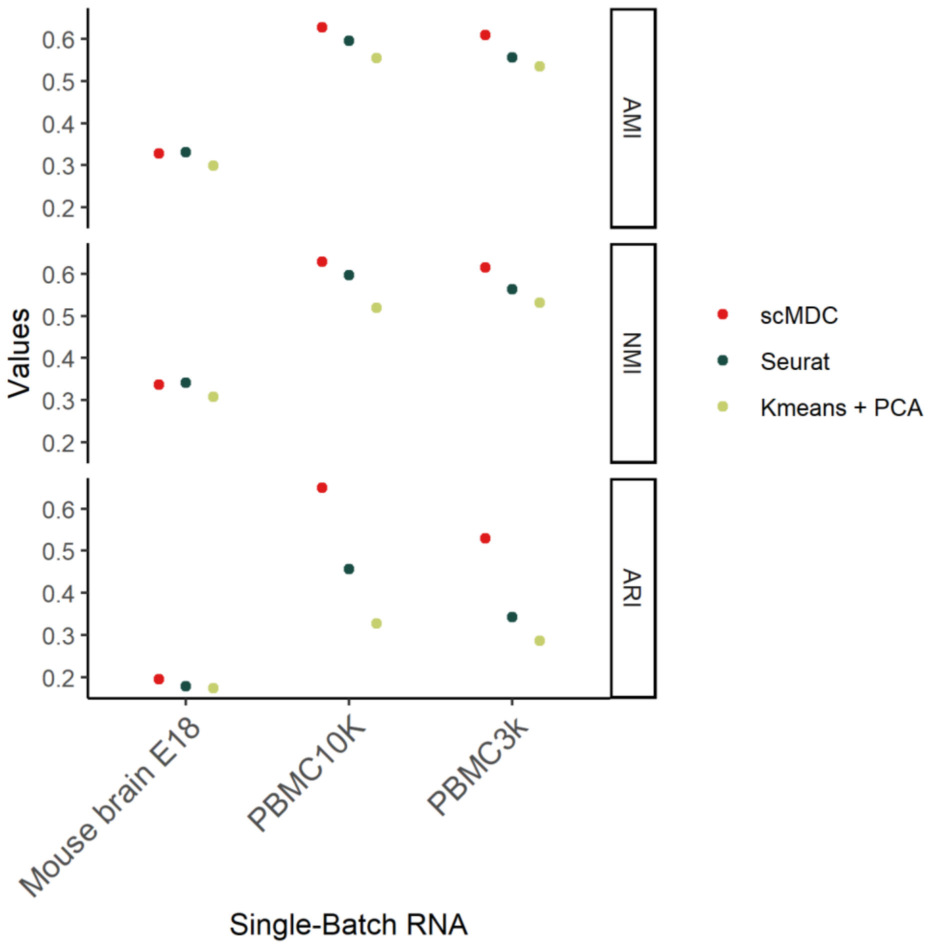


Figure B.5 Clustering performance of scMDC-RNA and two single-modal clustering methods on the single-batch SMAGE-seq datasets. All methods only take mRNA counts or normalized counts as input. Clustering performance is evaluated by AMI, NMI, and ARI.

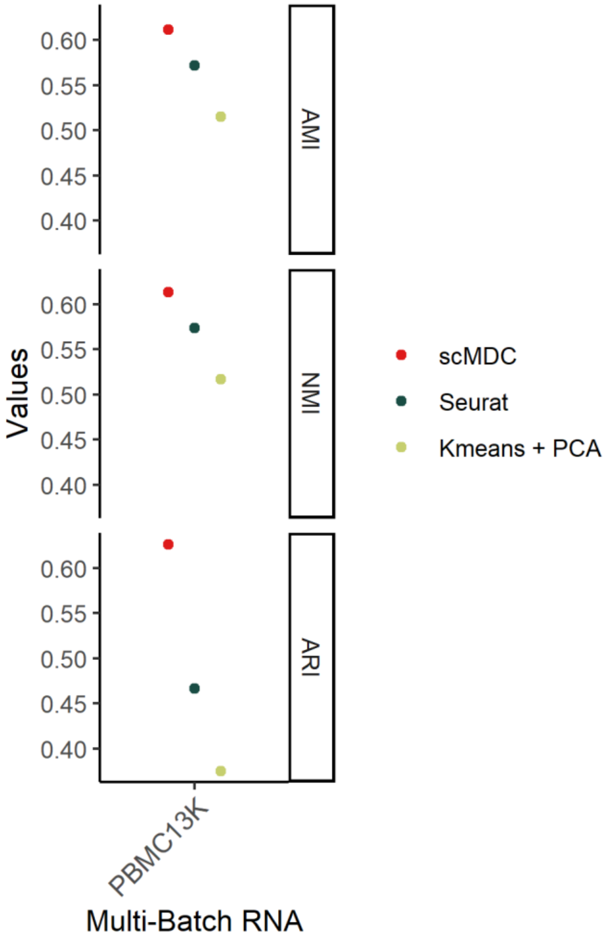


Figure B.6 Clustering performance of scMDC-RNA and two single-modal clustering methods on a multiple-batch SMAGE-seq dataset. All methods only take mRNA counts or normalized counts as input. Clustering performance is evaluated by AMI, NMI, and ARI.

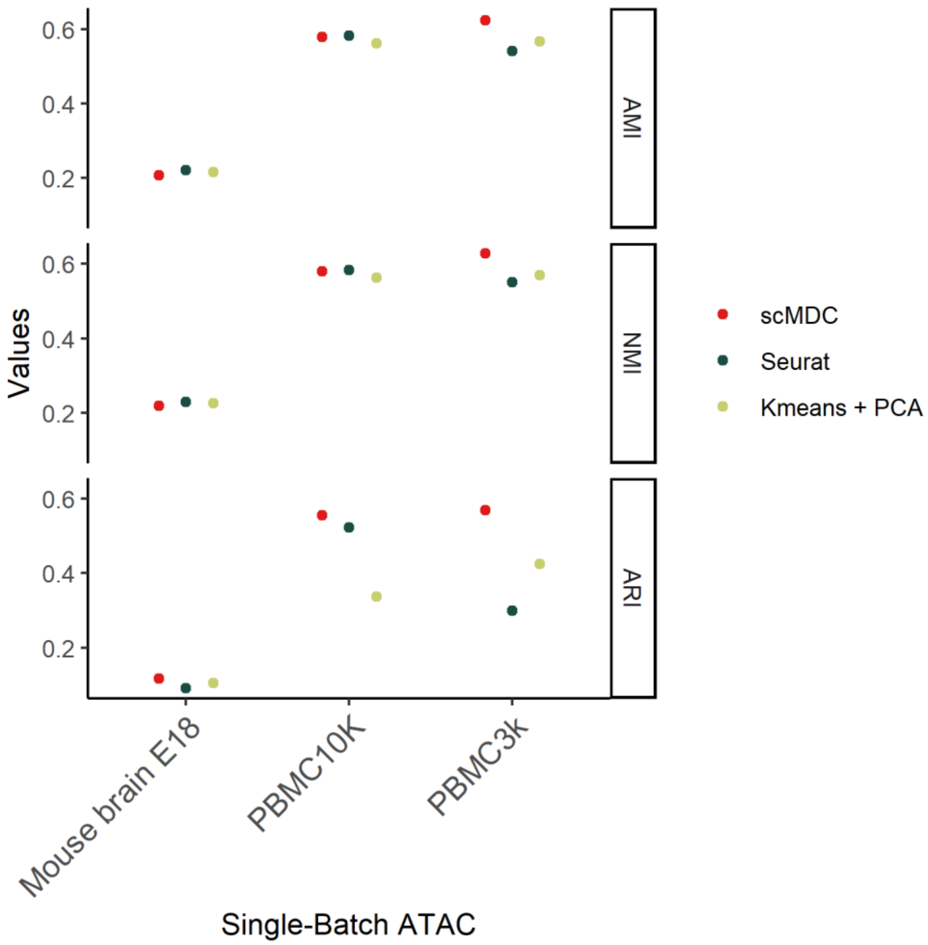


Figure B.7 Clustering performance of scMDC-ATAC and two single-modal clustering methods on the single-batch SMAGE-seq datasets. All methods only take ATAC counts or normalized counts as input. The ATAC counts are mapped to the gene regions. Clustering performance is evaluated by AMI, NMI, and ARI.

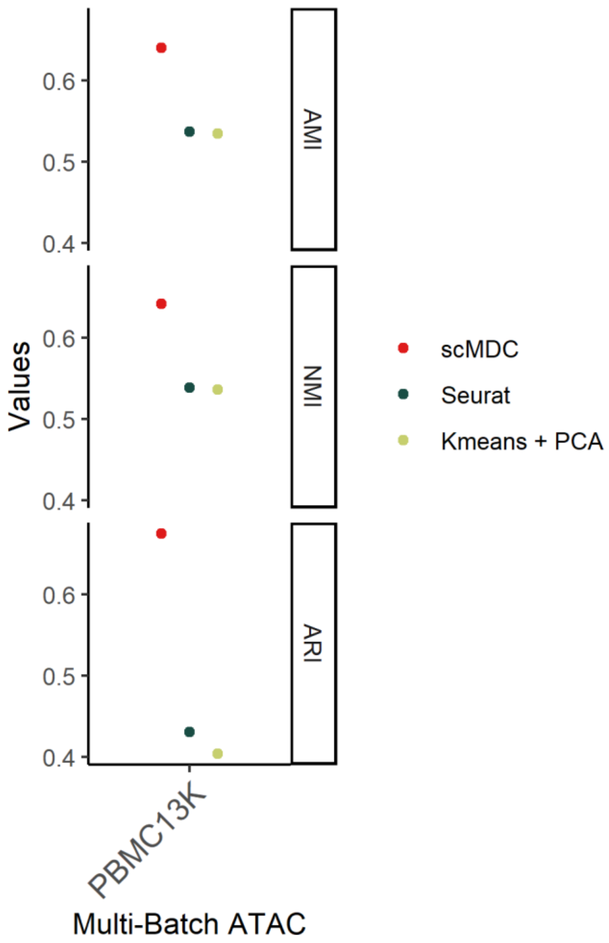
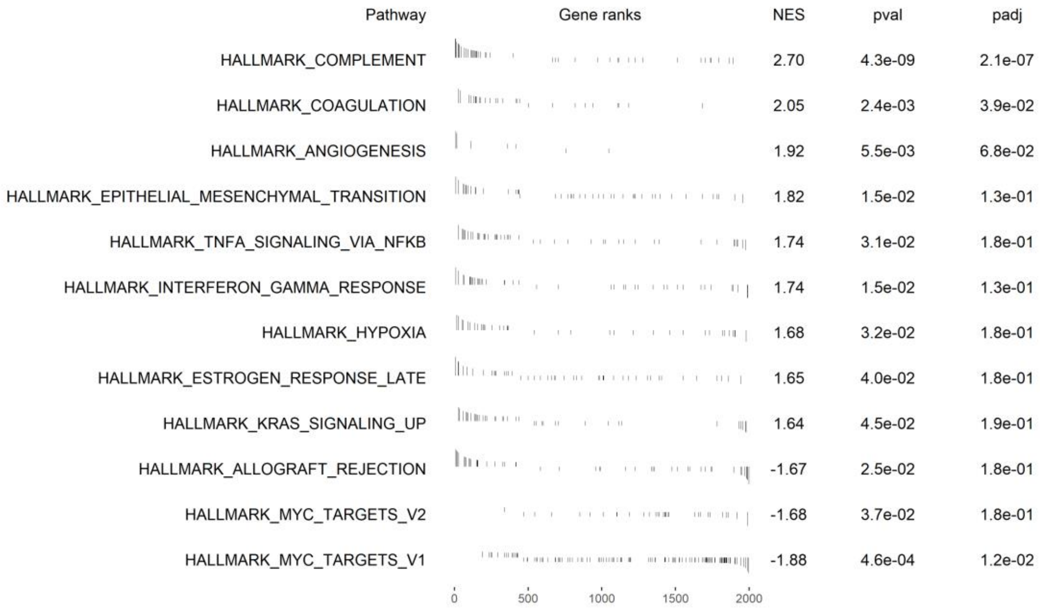


Figure B.8 Clustering performance of scMDC-ATAC and two single-modal clustering methods on a multiple-batch SMAGE-seq dataset. All methods only take ATAC counts or normalized counts as input. The ATAC counts are mapped to the gene regions. Clustering performance is evaluated by AMI, NMI, and ARI.

APPENDIX C

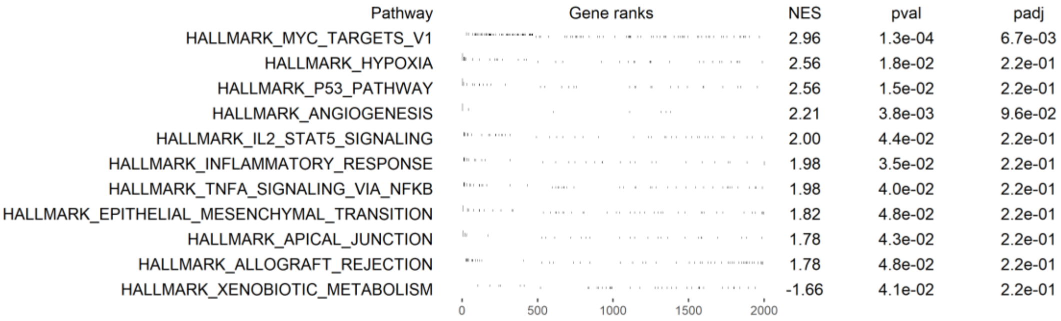
GSEA RESULTS OF THE BMNC DATASET

Figures C.1-4 show the GSEA results based on the clustering results from scMDC for different genesets.



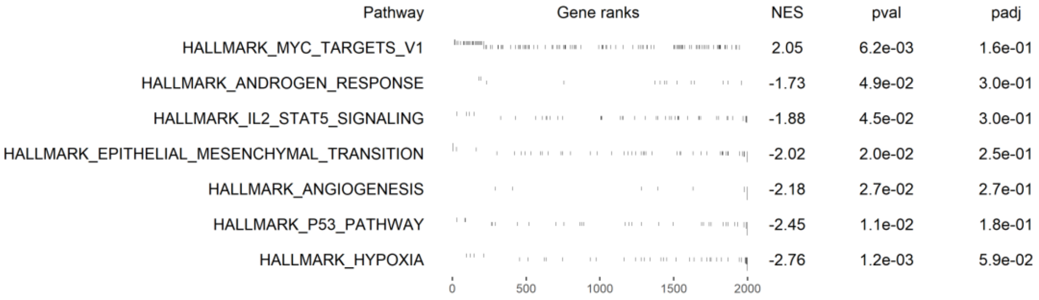
Pathway	Gene ranks	NES	pval	padj
HALLMARK_COMPLEMENT		2.70	4.3e-09	2.1e-07
HALLMARK_COAGULATION		2.05	2.4e-03	3.9e-02
HALLMARK_ANGIOGENESIS		1.92	5.5e-03	6.8e-02
HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION		1.82	1.5e-02	1.3e-01
HALLMARK_TNFA_SIGNALING_VIA_NFKB		1.74	3.1e-02	1.8e-01
HALLMARK_INTERFERON_GAMMA_RESPONSE		1.74	1.5e-02	1.3e-01
HALLMARK_HYPOXIA		1.68	3.2e-02	1.8e-01
HALLMARK_ESTROGEN_RESPONSE_LATE		1.65	4.0e-02	1.8e-01
HALLMARK_KRAS_SIGNALING_UP		1.64	4.5e-02	1.9e-01
HALLMARK_ALLOGRAFT_REJECTION		-1.67	2.5e-02	1.8e-01
HALLMARK_MYC_TARGETS_V2		-1.68	3.7e-02	1.8e-01
HALLMARK_MYC_TARGETS_V1		-1.88	4.6e-04	1.2e-02

Figure C.1 Enrichment plot of Hallmark pathways in CD14 monocyte cells from the BMNC dataset. The Kolmogorov-Smirnov test is used here, and the nominal P-values are adjusted for multiple comparisons (padj) by Benjamini & Hochberg (BH) method. Pathways with nominal P-values < 0.05 are shown.



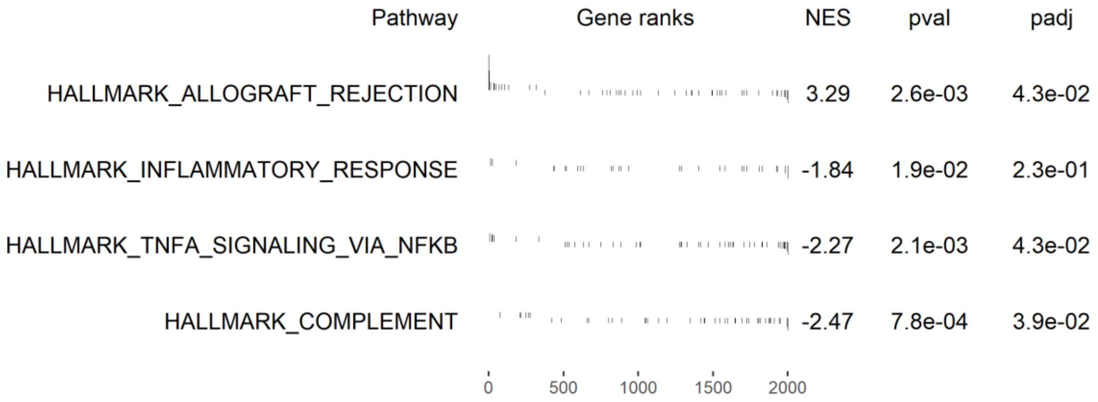
Pathway	Gene ranks	NES	pval	padj
HALLMARK_MYC_TARGETS_V1		2.96	1.3e-04	6.7e-03
HALLMARK_HYPOXIA		2.56	1.8e-02	2.2e-01
HALLMARK_P53_PATHWAY		2.56	1.5e-02	2.2e-01
HALLMARK_ANGIOGENESIS		2.21	3.8e-03	9.6e-02
HALLMARK_IL2_STAT5_SIGNALING		2.00	4.4e-02	2.2e-01
HALLMARK_INFLAMMATORY_RESPONSE		1.98	3.5e-02	2.2e-01
HALLMARK_TNFA_SIGNALING_VIA_NFKB		1.98	4.0e-02	2.2e-01
HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION		1.82	4.8e-02	2.2e-01
HALLMARK_APICAL_JUNCTION		1.78	4.3e-02	2.2e-01
HALLMARK_ALLOGRAFT_REJECTION		1.78	4.8e-02	2.2e-01
HALLMARK_XENOBIOTIC_METABOLISM		-1.66	4.1e-02	2.2e-01

Figure C.2 Enrichment plot of Hallmark pathways in CD4 memory cells from the BMNC dataset. The Kolmogorov-Smirnov test is used here, and the nominal P-values are adjusted for multiple comparisons (padj) by Benjamini & Hochberg (BH) method. Pathways with nominal P-values < 0.05 are shown.



Pathway	Gene ranks	NES	pval	padj
HALLMARK_MYC_TARGETS_V1		2.05	6.2e-03	1.6e-01
HALLMARK_ANDROGEN_RESPONSE		-1.73	4.9e-02	3.0e-01
HALLMARK_IL2_STAT5_SIGNALING		-1.88	4.5e-02	3.0e-01
HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION		-2.02	2.0e-02	2.5e-01
HALLMARK_ANGIOGENESIS		-2.18	2.7e-02	2.7e-01
HALLMARK_P53_PATHWAY		-2.45	1.1e-02	1.8e-01
HALLMARK_HYPOXIA		-2.76	1.2e-03	5.9e-02

Figure C.3 Enrichment plot of Hallmark pathways in CD4 naive cells from the BMNC dataset. The Kolmogorov-Smirnov test is used here, and the nominal P-values are adjusted for multiple comparisons (padj) by Benjamini & Hochberg (BH) method. Pathways with nominal P-values < 0.05 are shown.



Pathway	Gene ranks	NES	pval	padj
HALLMARK_ALLOGRAFT_REJECTION		3.29	2.6e-03	4.3e-02
HALLMARK_INFLAMMATORY_RESPONSE		-1.84	1.9e-02	2.3e-01
HALLMARK_TNFA_SIGNALING_VIA_NFKB		-2.27	2.1e-03	4.3e-02
HALLMARK_COMPLEMENT		-2.47	7.8e-04	3.9e-02

Figure C.4 Enrichment plot of Hallmark pathways in CD8 naive cells from the BMNC dataset. The Kolmogorov-Smirnov test is used here, and the nominal P-values are adjusted for multiple comparisons (padj) by Benjamini & Hochberg (BH) method. Pathways with nominal P-values < 0.05 are shown.

APPENDIX D

STATISTICAL TESTS OF SCMDC

Tables D.1-5 show the results of statistical tests of the experiment.

Table D.1 One-Sided Paired T-Test between the Clustering Performance of scMDC and the Competing Methods for the CITE-Seq Datasets

Methods	**p_AMI***	**p_NMI***	**p_ARI***
BREM-SC	0.00402824	0.00355908	9.1677E-06
CiteFuse	0.0079801	0.01111861	0.00036261
IDEC	6.7698E-05	7.57E-05	5.7372E-07
Kmeans + PCA	2.1861E-05	2.1894E-05	6.0185E-05
SC3	1.4569E-05	1.366E-05	2.2145E-05
SCVIS	0.00025911	0.00030163	5.887E-06
Seurat	0.00212642	0.00220737	0.00062321
Specter	0.00015003	0.00010859	0.00161893
TotalVI	0.01579666	0.0144765	0.00069109
Tscan	2.0401E-05	2.4565E-05	1.9785E-05

* p indicates P-value.

Table D.2 One-Sided Paired T-Test between the Clustering Performance of scMDC and the Competing Methods for the SMAGE-Seq Datasets

Methods	**p_AMI***	**p_NMI***	**p_ARI***
Cobolt	0.04201604	0.04327985	0.01847998
Kmeans + PCA	0.00932083	0.00834753	0.01511548
scMM	0.00944043	0.00970153	0.01339524
Seurat	0.01684468	0.01755079	0.01762545

* p indicates P-value.

Table D.3 One-Sided Paired T-Test between the Clustering Performance of scMDC and the Competing Methods for the Simulation Datasets

Methods	**p_AMI***	**p_NMI***	**p_ARI***
BREMSC	0.00205187	0.00194259	6.9106E-05
CiteFuse	3.9747E-06	3.9025E-06	2.7077E-05
iDEC	5.5039E-07	5.4942E-07	9.7328E-07
PCA+Kmeans	0.00012266	0.00012191	0.00012582
SC3	7.5575E-05	7.5267E-05	9.2593E-06
SCVIS	4.3007E-05	4.2824E-05	9.334E-06
Seurat	5.6212E-06	4.9064E-06	0.00022347
Specter	1.2021E-06	1.4494E-06	1.1547E-05
TotalVI	0.0028567	0.00282413	0.0248134
Tscan	5.0904E-05	5.1844E-05	1.4705E-05

* p indicates P-value.

Table D.4 One-Sided Paired T-Test between the Clustering Performance of scMDC and the Competing Methods for the Model Testing Experiments

Method1	**Method2**	**Pval_AMI***	**Pval_NMI***	**Pval_ARI***
scMDC	ATAC	0.07041006	0.07245205	0.09195784
scMDC	Concat-ATAC	0.00194839	0.00135246	0.0296167
scMDC	RNA	0.00015569	0.00016842	0.00013612
scMDC	ADT	0.00124744	0.0011954	0.00185413
scMDC	Concat-ADT	9.0239E-06	9.9314E-06	8.4946E-06

Table D.5 One-Sided Paired T-Test between the Clustering Performance of scMDC and the Competing Methods for the Parameter Tunning Experiments

Parameters	**Values**	**Pvals_ami***	**Pvals_nmi***	**Pvals_ari***
Fi	*0.0001*	*0.657522448*	*0.654359414*	*0.339330244*
Fi	*0.001*	*0.061665126*	*0.061031154*	*0.15169793*
Fi	*0.005*	*0.185708427*	*0.183215647*	*0.065754824*
Fi	*0.01*	*0.721740687*	*0.721638474*	*0.172244312*
Fi	*0.1*	*0.996335282*	*0.996328807*	*0.993537274*
Fi	*1*	*0.999079993*	*0.99907693*	*0.998847524*
Gamma	*0.01*	*0.404148719*	*0.402075548*	*0.465113431*
Gamma	*0.1*	*0.020012276*	*0.019725304*	*0.027002903*
Gamma	*1*	*0.273661585*	*0.272609533*	*0.211856888*
Gamma	*10*	*0.505974017*	*0.505992115*	*0.565385718*
Gamma	*100*	*0.859013414*	*0.858343271*	*0.82483211*

APPENDIX E

STATISTICAL TESTS OF DSSC

Table E.1 Statistical Test of the Simulation Results with 10% Permutation

Datasets	Methods1	Methods2	Pval_AC	Pval_NMI	Pval_ARI
151507	*K*-means + PCA	DSSC	4.24633E-13	3.28874E-14	2.0376E-13
151507	SC3	DSSC	5.18832E-14	5.36234E-17	4.54032E-15
151507	Seurat	DSSC	4.66238E-15	5.61941E-13	1.68655E-14
151507	BayesSpace	DSSC	0.082742145	0.119675253	0.120332566
151507	Giotto	DSSC	4.66252E-08	1.77105E-07	3.495E-07
151507	spaGCN	DSSC	0.00126916	0.000167417	0.003788902
151507	stLearn	DSSC	9.9485E-11	4.79135E-12	9.39773E-12
osmFish	*K*-means + PCA	DSSC	1.35883E-10	2.36585E-10	4.95313E-11
osmFish	SC3	DSSC	3.02622E-14	8.97591E-15	9.08673E-16
osmFish	Seurat	DSSC	2.43221E-09	9.52464E-11	7.35759E-10
osmFish	BayesSpace	DSSC	1.60168E-08	3.67412E-08	4.25781E-09
osmFish	Giotto	DSSC	1.12121E-07	9.31088E-07	4.97187E-07
osmFish	spaGCN	DSSC	1.5628E-05	3.05188E-06	2.46248E-05
osmFish	stLearn	DSSC	1.8695E-10	1.86878E-10	5.91744E-10

Table E.2 Statistical Test of the Simulation Results with 15% Permutation

Datasets	Methods	Methods	Pval_AC	Pval_NMI	Pval_ARI
151507	*K*-means + PCA	DSSC	2.43431E-09	1.89334E-12	1.40263E-10
151507	SC3	DSSC	1.77821E-11	2.25215E-17	2.07945E-12
151507	Seurat	DSSC	1.46316E-10	1.51041E-11	7.82944E-11
151507	BayesSpace	DSSC	0.283311661	0.322865463	0.345060618
151507	Giotto	DSSC	8.76033E-05	3.53207E-06	8.19187E-05
151507	spaGCN	DSSC	0.024437147	0.000562175	0.012315758
151507	stLearn	DSSC	7.54573E-08	3.32349E-10	1.06938E-08
osmFish	*K*-means + PCA	DSSC	2.14806E-10	8.87443E-10	5.06215E-10
osmFish	SC3	DSSC	1.1814E-14	7.26523E-17	6.14065E-15
osmFish	Seurat	DSSC	1.89863E-08	3.04302E-10	8.1556E-09
osmFish	BayesSpace	DSSC	7.83113E-08	1.97529E-05	5.59584E-07
osmFish	Giotto	DSSC	1.31499E-09	3.46474E-10	1.24645E-09
osmFish	spaGCN	DSSC	4.02776E-05	1.06192E-05	3.74999E-05
osmFish	stLearn	DSSC	7.13647E-10	3.29909E-10	3.02557E-09

Table E.3 Statistical Test of the Simulation Results with 20% Permutation

Datasets	Methods	Methods	Pval_AC	Pval_NMI	Pval_ARI
151507	*K*-means + PCA	DSSC	6.89168E-14	1.04625E-13	3.67873E-15
151507	SC3	DSSC	1.78392E-15	2.81701E-18	7.01654E-16
151507	Seurat	DSSC	2.62682E-13	1.39264E-11	1.23138E-13
151507	BayesSpace	DSSC	0.10972168	0.277267626	0.214369712
151507	Giotto	DSSC	3.68703E-07	1.20838E-07	9.85883E-07
151507	spaGCN	DSSC	0.004347438	0.003294814	0.008378494
151507	stLearn	DSSC	3.32357E-10	1.7915E-10	6.27605E-11
osmFish	*K*-means + PCA	DSSC	3.99819E-09	2.84669E-08	3.19154E-09
osmFish	SC3	DSSC	9.24393E-13	3.27959E-15	7.3741E-14
osmFish	Seurat	DSSC	4.12662E-08	7.63717E-09	3.81263E-08
osmFish	BayesSpace	DSSC	6.05709E-08	0.000359375	6.56227E-06
osmFish	Giotto	DSSC	1.39367E-07	1.7724E-07	9.13902E-08
osmFish	spaGCN	DSSC	1.19089E-05	1.21578E-05	2.37321E-05
osmFish	stLearn	DSSC	1.26497E-07	5.1673E-08	2.28827E-07

Table E.4 Statistical Test of the SpatialLIBD Data Results

Method1	Method2	Pval_AC	Pval_NMI	Pval_ARI
BayesSpace	DSSC	0.003729	0.116339	0.008479
spaGCN	DSSC	2.29E-05	1.96E-06	0.000432
stlearn	DSSC	2.2E-07	4.66E-07	8.03E-06
Seurat	DSSC	9.73E-08	8.64E-08	7.82E-06
K-means+PCA	DSSC	1.98E-08	2.99E-08	2.69E-06
SC3	DSSC	2.3E-05	7.28E-08	2.43E-06
Giotto	DSSC	3.43E-06	8.69E-08	7.66E-06

Table E.5 Statistical Test of the 10xMBAD Data Results

Methods1	Methods2	Pval-Silhouette
spaGCN	DSSC	0.211242658
BayesSpace	DSSC	0.476883153
spatialPCA	DSSC	0.002673712
stlearn	DSSC	0.000343514
Seurat	DSSC	0.032782218
K-means+PCA	DSSC	0.028644015
SC3	DSSC	0.043526239
Giotto	DSSC	0.003877227

Table E.6 Statistical Test of the K (in *k*NN) Tuning Results

*k*NN1	*k*NN2	Pval_AC	Pval_NMI	Pval_ARI
10	0	0.00036644	7.1717E-07	6.872E-05
20	0	0.00045065	1.5229E-06	0.0002041
40	0	0.00070832	1.444E-06	0.00040561
80	0	0.00032282	3.6124E-06	0.00053503
160	0	0.00083513	7.4927E-06	0.00044418

Table E.7 Statistical test of the Gamma (clustering loss) Tuning Results

Gamma1	Gamma2	Pval_AC	Pval_NMI	Pval_ARI
0.001	0	0.32270604	0.26621917	0.30667892
0.01	0	0.01047752	0.00072134	0.00151559
0.1	0	0.57586793	0.62270734	0.15635105
1	0	0.94119189	0.92822479	0.8616268
10	0	0.99289988	0.99766288	0.98608235

Table E.8 Statistical Test of the Constraint Number Tuning Results

Constraints1	Constraints2	Pval_AC	Pval_NMI	Pval_ARI
2000	0	0.07327009	0.07467783	0.06374234
4000	0	0.01695955	0.00296233	0.02178978
6000	0	0.01707602	0.02126256	0.01863524

APPENDIX F

SELECT MARKER GENES AS CONSTRAINTS FOR DSSC

For the experiments on the SpatialLIBD dataset, we use the marker genes reported from the original paper of this dataset. Users can add other marker genes according to their prior knowledge or the aim of the study. Before using the marker genes, we suggest checking the spatial dependency and the filtered smoothed expression pattern of the genes. The figures below show a good marker and a bad marker, respectively.

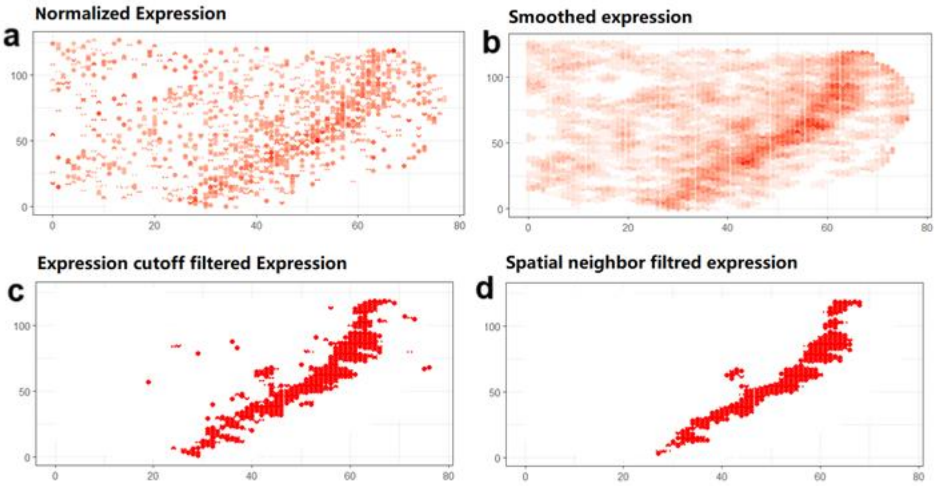


Figure F.1 The expression of a good marker gene in a single region or continuous regions (or only has low expression in a region, such as ENC1 in WM) (a) before smoothed by neighbors, (b) after smoothed by neighbors, (c) after filtered by a cutoff of expression, and (d) after filtered by the expression of neighbors.

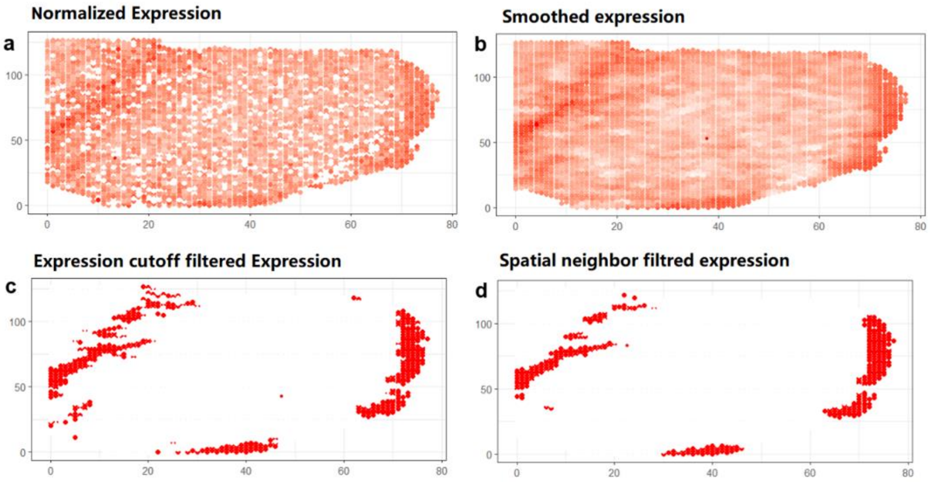


Figure F.2 The expression of a poor marker gene in multiple regions (a) before smoothed by neighbors, (b) after smoothed by neighbors, (c) after filtered by a cutoff of expression, and (d) after filtered by the expression of neighbors.

REFERENCES

Alexander, S., & Joydeep, G. (2003). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research, 3*, 583-617.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment, 2008*(10), P10008.

Buenrostro, J. D., Wu, B., Litzenburger, U. M., Ruff, D., Gonzales, M. L., Snyder, M. P., Chang, H. Y., & Greenleaf, W. J. (2015). Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature, 523*(7561), 486-490.

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

Caccamo, N., Joosten, S. A., Ottenhoff, T. H., & Dieli, F. (2018). Atypical human effector/memory CD4+ T cells with a naive-like phenotype. *Frontiers in Immunology*, 2832.

Chen, L., Wang, W., Zhai, Y., & 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, S., Lake, B. B., & Zhang, K. (2019). High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nature Biotechnology, 37*(12), 1452-1457.

Cho, S. H., Raybuck, A. L., Blagih, J., Kemboi, E., Haase, V. H., Jones, R. G., & Boothby, M. R. (2019). Hypoxia-inducible factors in CD4+ T cells promote metabolism, switch cytokine secretion, and T cell help in humoral immunity. *Proceedings of the National Academy of Sciences, 116*(18), 8975-8984.

Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Codeluppi, S., Borm, L. E., Zeisel, A., La Manno, G., van Lunteren, J. A., Svensson, C. I., & Linnarsson, S. (2018). Spatial organization of the somatosensory cortex revealed by osmFISH. *Nature Methods, 15*(11), 932-935.

Cusanovich, D. A., Daza, R., Adey, A., Pliner, H. A., Christiansen, L., Gunderson, K. L., Steemers, F. J., Trapnell, C., & Shendure, J. (2015). Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing. *Science, 348*(6237), 910-914.

Dimeloe, S., Mehling, M., Frick, C., Loeliger, J., Bantug, G. R., Sauder, U., Fischer, M., Belle, R. k., Develioglu, L., & Tay, S. (2016). The immune-metabolic basis of effector memory CD4+ T cell function under hypoxic conditions. *The Journal of Immunology, 196*(1), 106-114.

Ding, J., Condon, A., & Shah, S. P. (2018). Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature Communications, 9*(1), 1-13.

Dries, R., Zhu, Q., Dong, R., Eng, C.-H. L., Li, H., Liu, K., Fu, Y., Zhao, T., Sarkar, A., & Bao, F. (2021). Giotto: a toolbox for integrative analysis and visualization of spatial expression data. *Genome Biology, 22*(1), 1-31.

Efremova, M., Vento-Tormo, M., Teichmann, S. A., & Vento-Tormo, R. (2020). CellPhoneDB: inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes. *Nature Protocols, 15*(4), 1484-1506.

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S., & Theis, F. J. (2019). Single-cell RNA-seq denoising using a deep count autoencoder. *Nature Communications, 10*(1), 1-14.

Femino, A. M., Fay, F. S., Fogarty, K., & Singer, R. H. (1998). Visualization of single RNA transcripts in situ. *Science, 280*(5363), 585-590.

Gavin, C., Meinke, S., Heldring, N., Heck, K. A., Achour, A., Iacobaeus, E., Hoglund, P., Le Blanc, K., & Kadri, N. (2019). The complement system is essential for the phagocytosis of mesenchymal stromal cells by monocytes. *Frontiers in Immunology*, 2249.

Gayoso, A., Steier, Z., Lopez, R., Regier, J., Nazor, K. L., Streets, A., & Yosef, N. (2021). Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nature Methods, 18*(3), 272-282.

Gong, B., Zhou, Y., & Purdom, E. (2021). Cobolt: integrative analysis of multimodal single-cell sequencing data. *Genome Biology, 22*(1), 1-21.

Guillozet-Bongaarts, A., Hyde, T., Dalley, R., Hawrylycz, M., Henry, A., Hof, P., Hohmann, J., Jones, A., Kuan, C., & Royall, J. (2014). Altered gene expression in the dorsolateral prefrontal cortex of individuals with schizophrenia. *Molecular Psychiatry, 19*(4), 478-485.

Haider, S., & Pal, R. (2013). Integrated analysis of transcriptomic and proteomic data. *Current Genomics, 14*(2), 91-110.

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., Zheng, S., Butler, A., Lee, M. J., Wilk, A. J., Darby, C., & Zager, M. (2021). Integrated analysis of multimodal single-cell data. *Cell, 184*(13), 3573-3587. e3529.

Harding, S. D., Sharman, J. L., Faccenda, E., Southan, C., Pawson, A. J., Ireland, S., Gray, A. J., Bruce, L., Alexander, S. P., & Anderton, S. (2018). The IUPHAR/BPS Guide to pharmacology in 2018: updates and expansion to encompass the new guide to immunopharmacology. *Nucleic Acids Research, 46*(D1), D1091-D1106.

Hasan, F., Chiu, Y., Shaw, R. M., Wang, J., & Yee, C. (2021). Hypoxia acts as an environmental cue for the human tissue-resident memory T cell differentiation program. *JCI Insight, 6*(10).

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

Hu, J., Li, X., Coleman, K., Schroeder, A., Ma, N., Irwin, D. J., Lee, E. B., Shinohara, R. T., & Li, M. (2021). SpaGCN: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature Methods, 18*(11), 1342-1351.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*(1), 193-218. doi:10.1007/bf01908075

Ji, Z., & Ji, H. (2016). TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Research, 44*(13), e117-e117.

Jones, D. M., Read, K. A., & Oestreich, K. J. (2020). Dynamic Roles for IL-2—STAT5 Signaling in Effector and Regulatory CD4+ T Cell Populations. *The Journal of Immunology, 205*(7), 1721-1730.

Kaur, G., & Levy, E. (2012). Cystatin C in Alzheimer's disease. *Frontiers in Molecular Neuroscience, 5*, 79.

Kim, H. J., Lin, Y., Geddes, T. A., Yang, J. Y. H., & Yang, P. (2020). CiteFuse enables multi-modal analysis of CITE-seq data. *Bioinformatics, 36*(14), 4137-4143.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., Natarajan, K. N., Reik, W., Barahona, M., & Green, A. R. (2017). SC3: consensus clustering of single-cell RNA-seq data. *Nature Methods, 14*(5), 483-486.

Koch, L. (2018). Altered splicing in Alzheimer transcriptomes. *Nature Reviews Genetics, 19*(12), 738-739.

Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C., & Teichmann, S. A. (2015). The technology and biology of single-cell RNA sequencing. *Molecular Cell, 58*(4), 610-620.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly, 2*(1 - 2), 83-97.

Kumar, S., & Reddy, P. H. (2020). The role of synaptic microRNAs in Alzheimer's disease. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease, 1866*(12), 165937.

Kwon, S. (2013). Single-molecule fluorescence in situ hybridization: quantitative imaging of single RNA molecules. *BMB Reports, 46*(2), 65.

Larsson, L., Frisén, J., & Lundeberg, J. (2021). Spatially resolved transcriptomics adds a new dimension to genomics. *Nature Methods, 18*(1), 15-18.

Liao, J., Lu, X., Shao, X., Zhu, L., & Fan, X. (2021). Uncovering an organ’s molecular architecture at single-cell resolution by spatially resolved transcriptomics. *Trends in Biotechnology, 39*(1), 43-58.

Lipiec, M. A., Bem, J., Kozinski, K., Chakraborty, C., Urban-Ciecko, J., Zajkowski, T., Dabrowski, M., Szewczyk, L. M., Toval, A., & Ferran, J. L. (2020). TCF7L2 regulates postmitotic differentiation programmes and excitability patterns in the thalamus. *Development, 147*(16), dev190181.

Liu, Q., Chen, S., Jiang, R., & Wong, W. H. (2021). Simultaneous deep generative modelling and clustering of single-cell genomic data. *Nature Machine Intelligence, 3*(6), 536-544.

Liu, Q., Xu, J., Jiang, R., & Wong, W. H. (2021). Density estimation using deep generative neural networks. *Proceedings of the National Academy of Sciences, 118*(15), e2101344118.

Longo, S. K., Guo, M. G., Ji, A. L., & Khavari, P. A. (2021). Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics. *Nature Reviews Genetics*, 1-18.

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

Love, M. I., Huber, W., & Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology, 15*(12), 1-21.

Lu, Y. Y., Yu, T. C., Bonora, G., & Noble, W. S. (2021). *ACE: Explaining cluster from an adversarial perspective*. Paper presented at the Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research. https://proceedings.mlr.press/v139/lu21e.html

Lubeck, E., & Cai, L. (2012). Single-cell systems biology by super-resolution imaging and combinatorial labeling. *Nature Methods, 9*(7), 743-748.

Ma, A., McDermaid, A., Xu, J., Chang, Y., & Ma, Q. (2020). Integrative methods and practical challenges for single-cell multi-omics. *Trends in Biotechnology*.

Ma, S., Zhang, B., LaFave, L. M., Earl, A. S., Chiang, Z., Hu, Y., Ding, J., Brack, A., Kartha, V. K., & Tay, T. (2020). Chromatin potential identified by shared single-cell profiling of RNA and chromatin. *Cell, 183*(4), 1103-1116. e1120.

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*(Nov), 2579-2605.

Marchingo, J. M., Sinclair, L. V., Howden, A. J., & Cantrell, D. A. (2020). Quantitative analysis of how Myc controls T cell proteomes and metabolic pathways during T cell activation. *Elife, 9*, e53725.

Maynard, K. R., Collado-Torres, L., Weber, L. M., Uytingco, C., Barry, B. K., Williams, S. R., Catallini, J. L., Tran, M. N., Besich, Z., & Tippani, M. (2021). Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature Neuroscience, 24*(3), 425-436.

McCarthy, D. J., Campbell, K. R., Lun, A. T., & Wills, Q. F. (2017). Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics, 33*(8), 1179-1186. doi:10.1093/bioinformatics/btw777

Miller, B. F., Bambah-Mukku, D., Dulac, C., Zhuang, X., & Fan, J. (2021). Characterizing spatial gene expression heterogeneity in spatially resolved single-cell transcriptomics data with nonuniform cellular densities. *Genome Research*, gr. 271288.271120.

Mimitou, E. P., Cheng, A., Montalbano, A., Hao, S., Stoeckius, M., Legut, M., Roush, T., Herrera, A., Papalexi, E., & Ouyang, Z. (2019). Multiplexed detection of proteins, transcriptomes, clonotypes and CRISPR perturbations in single cells. *Nature Methods, 16*(5), 409-412.

Minoura, K., Abe, K., Nam, H., Nishikawa, H., & Shimamura, T. (2021). A mixture-of-experts deep generative model for integrated analysis of single-cell multiomics data. *Cell Reports Methods, 1*(5), 100071.

Moran, P. A. (1950). A test for the serial independence of residuals. *Biometrika, 37*(1/2), 178-181.

Nair, V., & Hinton, G. E. (2010). *Rectified linear units improve restricted boltzmann machines.* Paper presented at the ICML.

Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems, 14*, 849-856.

Pardilla-Delgado, E., Torrico-Teave, H., Sanchez, J. S., Ramirez-Gomez, L. A., Baena, A., Bocanegra, Y., Vila-Castelar, C., Fox-Fuller, J. T., Guzman-Velez, E., & Martinez, J. (2021). Associations between subregional thalamic volume and brain pathology in autosomal dominant Alzheimer's disease. *Brain Communications, 3*(2), fcab101.

Pardo, B., Spangler, A., Weber, L. M., Page, S. C., Hicks, S. C., Jaffe, A. E., Martinowich, K., Maynard, K. R., & Collado-Torres, L. (2022). spatialLIBD: an R/Bioconductor package to visualize spatially-resolved transcriptomics data. *BMC Genomics, 23*(1), 1-5.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). *Automatic differentiation in pytorch*. Paper presented at the Neural Information Processing Systems.

Peterson, V. M., Zhang, K. X., Kumar, N., Wong, J., Li, L., Wilson, D. C., Moore, R., McClanahan, T. K., Sadekova, S., & Klappenbach, J. A. (2017). Multiplexed quantification of proteins and transcripts in single cells. *Nature Biotechnology, 35*(10), 936-939.

Pham, D., Tan, X., Xu, J., Grice, L. F., Lam, P. Y., Raghubar, A., Vukovic, J., Ruitenberg, M. J., & Nguyen, Q. (2020). stLearn: integrating spatial location, tissue morphology and gene expression to find cell types, cell-cell interactions and spatial trajectories within undissociated tissues. *BioRxiv*.

Reddi, S., Kale, S., & Kumar, S. (2018). *On the convergence of adam and be-365 yond.* Paper presented at the International Conference on Learning Representations.

Ringeling, F. R., & Canzar, S. (2021). Linear-time cluster ensembles of large-scale single-cell RNA-seq and multimodal data. *Genome Research, 31*(4), 677-688.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., & Vert, J.-P. (2018). A general and flexible method for signal extraction from single-cell RNA-seq data. *Nature Communications, 9*(1), 1-17.

Ross, S. H., & Cantrell, D. A. (2018). Signaling and function of interleukin-2 in T lymphocytes. *Annual Review of Immunology, 36*, 411.

Schlachetzki, J., Prots, I., Tao, J., Chun, H. B., Saijo, K., Gosselin, D., Winner, B., Glass, C. K., & Winkler, J. (2018). A monocyte gene expression signature in the early clinical course of Parkinson's disease. *Scientific Reports, 8*(1), 1-13.

Shah, S., Lubeck, E., Zhou, W., & Cai, L. (2016). In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. *Neuron, 92*(2), 342-357.

Shapiro, E., Biezuner, T., & Linnarsson, S. (2013). Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nature Reviews Genetics, 14*(9), 618-630.

Simidjievski, N., Bodnar, C., Tariq, I., Scherer, P., Andres Terre, H., Shams, Z., Jamnik, M., & Liò, P. (2019). Variational autoencoders for cancer data integration: design principles and computational practice. *Frontiers in Genetics, 10*, 1205.

Skelly, D. A., Squiers, G. T., McLellan, M. A., Bolisetty, M. T., Robson, P., Rosenthal, N. A., & Pinto, A. R. (2018). Single-cell transcriptional profiling reveals cellular diversity and intercommunication in the mouse heart. *Cell Reports, 22*(3), 600-610.

Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P. K., Swerdlow, H., Satija, R., & Smibert, P. (2017). Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods, 14*(9), 865-868.

Stoltzfus, C. R., Filipek, J., Gern, B. H., Olin, B. E., Leal, J. M., Wu, Y., Lyons-Cohen, M. R., Huang, J. Y., Paz-Stoltzfus, C. L., & Plumlee, C. R. (2020). CytoMAP: a spatial analysis toolbox reveals features of myeloid cell organization in lymphoid tissues. *Cell Reports, 31*(3), 107523.

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck III, W. M., Hao, Y., Stoeckius, M., Smibert, P., & Satija, R. (2019). Comprehensive integration of single-cell data. *Cell, 177*(7), 1888-1902. e1821.

Stuart, T., Srivastava, A., Lareau, C., & Satija, R. (2020). Multimodal single-cell chromatin analysis with Signac. *BioRxiv*.

Svensson, V., Teichmann, S. A., & Stegle, O. (2018). SpatialDE: identification of spatially variable genes. *Nature Methods, 15*(5), 343-346.

Takahama, S., Nakaya, N., & Tomarev, S. I. (2014). Olfactomedin 1 may suppress APP cleavage through its interaction with BACE1. *Investigative Ophthalmology and Visual Science, 55*(13), 2959-2959.

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

Tian, T., Zhang, J., Lin, X., Wei, Z., & Hakonarson, H. (2021a). Model-based deep embedding for constrained clustering analysis of single cell RNA-seq data. *Nature Communications, 12*(1), 1-12.

Tian, T., Zhang, J., Lin, X., Wei, Z., & Hakonarson, H. (2021b). Model-based deep embedding for constrained clustering analysis of single cell RNA-seq data. *Nature Communications, 12*(1). doi:10.1038/s41467-021-22008-3

Van De Mortel, L. A., Thomas, R. M., Van Wingen, G. A., & Initiative, A. s. D. N. (2021). Grey matter loss at different stages of cognitive decline: a role for the thalamus in developing Alzheimer's disease. *Journal of Alzheimer's Disease, 83*(2), 705-720.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). *Extracting and composing robust features with denoising autoencoders.* Paper presented at the Proceedings of the 25th International Conference on Machine Learning.

Vinh, N. X., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research, 11*, 2837-2854.

Wang, B., Mezlini, A. M., Demir, F., Fiume, M., Tu, Z., Brudno, M., Haibe-Kains, B., & Goldenberg, A. (2014). Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods, 11*(3), 333.

Wang, S., Karikomi, M., MacLean, A. L., & Nie, Q. (2019). Cell lineage and communication network inference via optimization for single-cell transcriptomics. *Nucleic Acids Research, 47*(11), e66-e66.

Wang, X., Sun, Z., Zhang, Y., Xu, Z., Xin, H., Huang, H., Duerr, R. H., Chen, K., Ding, Y., & Chen, W. (2020). BREM-SC: a bayesian random effects mixture model for joint clustering single cell multi-omics data. *Nucleic Acids Research, 48*(11), 5814-5824.

Wolf, F. A., Angerer, P., & Theis, F. J. (2018). SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology, 19*(1), 15.

Xia, C., Fan, J., Emanuel, G., Hao, J., & Zhuang, X. (2019). Spatial transcriptome profiling by MERFISH reveals subcellular RNA compartmentalization and cell cycle-dependent gene expression. *Proceedings of the National Academy of Sciences, 116*(39), 19490-19499.

Xie, J., Girshick, R., & Farhadi, A. (2016). *Unsupervised deep embedding for clustering analysis.* Paper presented at the International Conference on Machine Learning.

Zappia, L., Phipson, B., & Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. *Genome Biology, 18*(1), 1-15.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zeisel, A., Munoz-Manchado, A. B., Codeluppi, S., Lonnerberg, P., La Manno, G., Jureus, A., Marques, S., Munguba, H., He, L., & Betsholtz, C. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science, 347*(6226), 1138-1142.

Zhang, X., Xu, C., & Yosef, N. (2019). Simulating multiple faceted variability in single cell RNA sequencing. *Nature Communications, 10*(1), 1-16.

Zhao, E., Stone, M. R., Ren, X., Guenthoer, J., Smythe, K. S., Pulliam, T., Williams, S. R., Uytingco, C. R., Taylor, S. E., & Nghiem, P. (2021). Spatial transcriptomics at subspot resolution with BayesSpace. *Nature Biotechnology*, 1-10.

Zheng, G. X., 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. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature Communications, 8*, 14049. doi:10.1038/ncomms14049

Zhu, Q., Shah, S., Dries, R., Cai, L., & Yuan, G.-C. (2018). Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence in situ hybridization data. *Nature Biotechnology, 36*(12), 1183-1190.

Zhuang, X. (2021). Spatially resolved single-cell genomics and transcriptomics by imaging. *Nature Methods, 18*(1), 18-22.

Zou, Y.-m., Lu, D., Liu, L.-p., Zhang, H.-h., & Zhou, Y.-y. (2016). Olfactory dysfunction in Alzheimer's disease. *Neuropsychiatric Disease and Treatment, 12*, 869.