University of Windsor
Scholarship at UWindsor

Electronic Theses and Dissertations | Theses, Dissertations, and Major Papers

2-29-2024

# ADVANCED MACHINE LEARNING MODELS FOR ANALYZING SINGLE-CELL RNA-SEQUENCING DATA

AKRAM VASIGHIZAKER
*University of Windsor*

Follow this and additional works at: https://scholar.uwindsor.ca/etd

 Part of the Computer Sciences Commons

## Recommended Citation

VASIGHIZAKER, AKRAM, "ADVANCED MACHINE LEARNING MODELS FOR ANALYZING SINGLE-CELL RNA-SEQUENCING DATA" (2024). *Electronic Theses and Dissertations*. 9438.
https://scholar.uwindsor.ca/etd/9438

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

# ADVANCED MACHINE LEARNING MODELS FOR ANALYZING SINGLE-CELL RNA-SEQUENCING DATA

by
**Akram Vasighizaker**

A Dissertation
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada
2024

© 2024 Akram Vasighizaker

**ADVANCED MACHINE LEARNING MODELS FOR ANALYZING SINGLE-CELL RNA-SEQUENCING DATA**

by

**AKRAM VASIGHIZAKER**

APPROVED BY:

---

Y. Li, External Examiner
Brock University

---

E. Abdel-Raheem
Department of Electrical and Computer Engineering

---

S. Samet
School of Computer Science

---

D. Alhadidi
School of Computer Science

---

L. Rueda, Advisor
School of Computer Science

February 12, 2024

# Declaration of Co-Authorship and Previous Publication

**I. Co-Authorship**

I hereby declare that this Dissertation incorporates the outcome of a joint research undertaken in collaboration under the supervision of professor Luis Rueda.

The collaboration is covered in Chapters 2 to 7 of the Dissertation. In most cases, primary contributions, conceptualization, methodology, investigation, resources, data collection and data curation, writing—original draft preparation, review and editing, visualization, as well as interpretation and conducting the biological analysis of the results, were performed by myself, and the contribution of coauthors was primarily done through the implementation which was also re-implemented by the author. Professor Luis Rueda supervised the projects, contributed with initial thoughts and the main ideas, and assisted in elaborating on the new novel approaches implemented in this work and finalizing the idea. All the authors contributed to writing and follow-up discussions.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my Dissertation, and have obtained written permission from each of the co-author(s) to include the above material(s) in my Dissertation. I certify that, with the above qualification, this Dissertation,

and the research to which it refers, is the product of my own work.

**II. Previous Publications**

This Dissertation includes six original papers that have been previously published/submitted for publication in conferences and peer-reviewed journals, as follows:

| Dissertation chapter | Publication Title | Status |
|---|---|---|
| Chapter 2 | Vasighizaker, A., Danda, S. & Rueda, L. (2022). Discovering Cell Types using Manifold Learning and Enhanced Visualization of Single-cell RNA-seq Data. Scientific Reports 12, 120. | Published |
| Chapter 3 | Vasighizaker, A., Trivedi, Y., & Rueda, L. (2023). Cell Type Annotation Model Selection: General-Purpose vs. Pattern-Aware Feature Gene Selection in Single-Cell RNA-Seq Data. Genes, 14(3), 596. | Published |
| Chapter 4 | Vasighizaker, A., Zhou, L., & Rueda, L. (2021, August). Cell Type Identification via Convolutional Neural Networks and Self-Organizing Maps on Single-cell RNA-seq Data. In Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (pp. 1-6). | Published |
| Chapter 5 | Vasighizaker, A., Hora, S., Trivedi, Y., & Rueda, L. (2022, June). Comparative Analysis of Supervised Cell Type Detection in Single-Cell RNA-seq Data. In International Work-Conference on Bioinformatics and Biomedical Engineering (pp. 333-345). Cham: Springer International Publishing. | Published |
| Chapter 6 | Danda, S., Vasighizaker, A., & Rueda, L. (2020, December). Unsupervised Identification of SARS-CoV-2 Target Cell Groups via Nonlinear Dimensionality Reduction on Single-cell RNA-Seq Data. In 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 2737-2744). IEEE. | Published |
| Chapter 7 | Vasighizaker, A., Hora, S., & Rueda, L. (2023) "SEGCECO: Subgraph Embedding of Gene expression matrix for CEll cell COmmunication prediction", Briefings in Bioinformatics. | Major revision submitted |

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my Dissertation. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my Dissertation does not infringe upon anyone copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my Dissertation, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my Dissertation. I declare that this is a true copy of my Dissertation, including any final revisions, as approved by my Dissertation committee and the Graduate Studies office, and that this Dissertation has not been submitted for a higher degree to any other University or Institution.

# Abstract

The advent of high-throughput scRNA-seq technologies has enabled the study of individual cells and their biological mechanisms. Traditional clustering methods, commonly employed in scRNA-seq data analysis for identifying cell types, face challenges due to the sparsity and high-dimensionality of the data. To overcome these limitations, we propose an integrated approach that combines non-linear dimensionality reduction techniques with clustering algorithms.

Our method involves the use of modified locally linear embedding in conjunction with independent component analysis to identify representative clusters of different cell types. We evaluate the performance of this approach across thirteen publicly available scRNA-seq datasets, encompassing various tissues, sizes, and technologies. Gene set enrichment analysis further confirms the effectiveness of our method, demonstrating superior performance compared to existing unsupervised methods across diverse datasets.

Also, we investigate Neural Network-based methods combined with self-organizing maps, feature selection approaches for informative marker gene selection in sparse datasets, as well as supervised techniques, to overcome the high-dimensionality and sparsity of scRNA-seq datasets in cell type identification.

Building on the foundation of identifying cell types, we extend our investigation to intercellular signaling networks. Recognizing the limitations of existing link prediction

approaches based on graph-structured data, we introduce a novel method named Subgraph Embedding of Gene expression matrix for prediction of CEll-cell COmmunication (SEGCECO). SEGCECO utilizes an attributed graph convolutional neural network to predict cell-cell communication from scRNA-seq data.

Overcoming challenges associated with high-dimensional and sparse scRNA-seq data, we employ SoptSC, a similarity-based optimization method, to construct a cell-cell communication network. Our experiments on six datasets from human and mouse pancreas tissue reveal that SEGCECO outperforms latent feature-based approaches and the state-of-the-art link prediction method, WLNM, achieving a remarkable 0.99 ROC and 99% prediction accuracy.

In summary, our approach, spanning the identification of cell types and the prediction of cell-cell communication, leverages advanced techniques to enhance the analysis of scRNA-seq data. This research contributes to the comprehensive understanding of disease modules and intercellular signaling networks, paving the way for more accurate and insightful investigations in the field of single-cell genomics.

# Dedication

In gratitude to my family, the bedrock of my support, especially my parents, for instilling in me the values of hard work, self-belief, and determination. Their enduring love and prayers fueled my resilience through the challenges of my Ph.D. journey, serving as a guiding force and inspiration. I am profoundly thankful for their belief in my capabilities and the sacrifices made for my success.

# Acknowledgements

I would like to take this opportunity to express my sincere gratitude to Dr. Luis Rueda, my supervisor, for his steady encouragement, patient guidance and enlightening discussions throughout my graduate studies. Without his help, the work presented here could not have been possible.

I also wish to express my appreciation to Dr. Yifeng Li from Brock University, Dr. Esam Abdel-Raheem, Department of Electrical and Computer Engineering, Dr. Saeed Samet and Dr. Dima Alhadidi, School of Computer Science for being in the committee and spending their valuable time.

Special thanks to my friends in the Pattern Recognition and Bioinformatics Lab and my collaborators in my research group, Sheena Hora, Saiteja Danda, Karan Kashyap, and others. I found our collaborative brainstorming sessions and shared projects to be highly beneficial in exchanging our knowledge and experiences.

# Contents

**Declaration of Co-Authorship and Previous Publication** iii

**Abstract** vii

**Dedication** ix

**Acknowledgements** x

**List of Figures** xvii

**List of Tables** xxii

**1 Introduction** 1

1.1 Single-Cell RNA Sequencing . . . . . 2

1.2 Cell Type Identification . . . . . 6

1.3 Computational Approaches for Cell Type Identification . . . . . 7

1.3.1 Unsupervised Approaches . . . . . 7

1.3.2 Supervised Approaches . . . . . 8

1.4 Dimensionality Reduction and Visualization . . . . . 8

1.4.1 t-Distributed Stochastic Neighbor Embedding . . . . . 9

1.4.2 Uniform Manifold Approximation and Projection . . . . . 9

1.5 Prediction of Cell-Cell Interactions . . . . . 10
1.6 Computational Methods for Cell-cell Interaction Prediction . . . . . 11
1.6.1 Heuristic Methods . . . . . 11
1.6.2 Graph-Based Methods . . . . . 12
1.7 Motivation and Objective . . . . . 14
1.8 Contributions . . . . . 14
1.9 Thesis Organization . . . . . 17

**2 Discovering cell types using manifold learning and enhanced visualization of single-cell RNA-Seq data 18**

2.1 Introduction . . . . . 18
2.2 Materials and Methods . . . . . 21
2.2.1 Datasets . . . . . 21
2.2.2 Data Pre-processing and Quality Control . . . . . 23
2.2.3 Dimensionality Reduction . . . . . 26
2.2.4 Clustering . . . . . 32
2.2.5 Cluster Annotation . . . . . 33
2.2.6 Parameter Optimization . . . . . 33
2.2.7 Performance Evaluation . . . . . 34
2.3 Results and Discussion . . . . . 35
2.3.1 Clustering and Cell Type Discovery . . . . . 36
2.3.2 Biological Assessment . . . . . 42
2.4 Conclusion and Future Work . . . . . 46

**3 Cell Type Annotation Model Selection: General-Purpose vs. Pattern-Aware Feature Gene Selection in Single-Cell RNA-Seq Data 49**

3.1 Introduction . . . 49

3.2 Materials and Methods . . . 53

3.2.1 Framework . . . 53

3.2.2 Dataset . . . 54

3.2.3 Data Pre-Processing . . . 55

3.2.4 Hyperparameter Tuning . . . 56

3.2.5 Feature Selection . . . 57

3.2.6 XGBoost . . . 58

3.3 Results and Discussion . . . 59

3.3.1 Classification Results . . . 60

3.3.2 Biological Validation . . . 62

3.4 Conclusions . . . 62

**4 Cell type identification via convolutional neural networks and self-organizing maps on single-cell RNA-seq data 67**

4.1 Introduction . . . 67

4.2 Materials and Methods . . . 69

4.2.1 Dataset . . . 69

4.2.2 Data Pre-processing . . . 70

4.2.3 Feature Selection . . . 72

4.2.4 Creating a Gene Similarity Network via Self-Organizing Maps . . . 73

4.2.5 Cell Type Classification . . . 74

4.3 Results and Discussion . . . 75

4.3.1 Experimental Results . . . . . 75
4.3.2 Biological validation . . . . . 78
4.4 Conclusion and Future Work . . . . . 79

**5 Comparative Analysis of Supervised Cell Type Detection in Single-Cell RNA-seq Data 81**
5.1 Introduction . . . . . 81
5.2 Materials and Methods . . . . . 83
5.2.1 Framework . . . . . 83
5.2.2 Dataset . . . . . 85
5.2.3 Data Pre-processing . . . . . 85
5.2.4 Feature Selection . . . . . 86
5.2.5 Evaluation Metrics . . . . . 88
5.3 Results and Discussion . . . . . 88
5.3.1 Parameter Optimization . . . . . 88
5.3.2 Classification Results . . . . . 91
5.3.3 Biological Validation . . . . . 93
5.4 Conclusion and Future Work . . . . . 94

**6 Unsupervised Identification of SARS-CoV-2 Target Cell Groups via Nonlinear Dimensionality Reduction on Single-cell RNA-Seq Data 98**
6.1 Introduction . . . . . 98
6.2 Materials and Methods . . . . . 101
6.2.1 Dataset . . . . . 102
6.2.2 Data Pre-processing and Quality Control . . . . . 102

6.2.3 Dimensionality Reduction . . . . . 105
6.2.4 Cell Clustering . . . . . 108
6.2.5 Cluster Annotation . . . . . 109
6.2.6 Parameter Optimization . . . . . 109
6.2.7 Performance Evaluation . . . . . 109
6.3 Results and Discussion . . . . . 111
6.3.1 Clustering Results . . . . . 111
6.3.2 Biological Assessment of the Results . . . . . 114
6.4 Conclusion and Future Work . . . . . 117

**7 SEGCECO: Subgraph Embedding of Gene expression matrix for prediction of CEll-cell COmmunication 119**

7.1 Introduction . . . . . 119
7.2 Materials and Methods . . . . . 123
7.2.1 Datasets . . . . . 123
7.2.2 Proposed Method . . . . . 123
7.3 Performance Evaluation . . . . . 131
7.3.1 Latent Feature Methods . . . . . 132
7.3.2 Graph-based Methods . . . . . 134
7.4 Results and Discussion . . . . . 135
7.5 Biological Assessment . . . . . 137
7.6 Conclusion . . . . . 139

**8 Conclusion and Future Work 141**

8.1 Conclusion and Future Work . . . . . 141

**Bibliography** **151**

**Vita Auctoris** **171**

# List of Figures


1.1 Schematic of a typical single-cell RNA-seq analysis workflow. [67] . . . . . 3

1.2 Overview of downstream analysis methods [68]. . . . . 4

1.3 An overview of the experimental steps in an RNA sequencing protocol. [105] 4

1.4 Cellular interaction among cells [4]. . . . . 10

1.5 Cell-cell interaction analysis framework [28]. . . . . 15

2.1 Block diagram of the proposed approach for discovering cell types in scRNA-seq data. . . . . 23

2.2 Investigating the distribution of the data to filtered out weakly expressed genes and low-quality cells from dataset; (a) number of expressed genes, (b) total counts per cell, and (c) the percentage of mitochondrial genes for H1299_scRNAseq. . . . . 24

2.3 Top 20 highly-variable genes before normalization. . . . . 24

2.4 Top 20 highly-variable genes after normalization. . . . . 25

2.5 Dispersion of genes before normalization. . . . . 27

2.6 Dispersion of genes after normalization. . . . . 27

2.7 Comparison of dispersion of normalized and not normalized genes to extract highly variable genes. . . . . 27

2.8 Visualization of t-SNE on Muraro dataset . . . . . 38

2.9 Visualization of PCA on Wang dataset . . . . . 39

2.10 Visualization of Laplacian eigenmaps on H1299_scRNAseq; outliers have been removed to enhance visualization. . . . . 39

2.11 Two-dimensional ICA projection of cells colored by *k*-means clustering applied on high-dimensional original data (H1299_scRNAseq). . . . . 40

2.12 Two-dimensional ICA projection of cells colored by *k*-means clustering applied to the three-dimensional points output by MLLE on the H1299_scRNAseq dataset. . . . . 40

2.13 Cluster annotation for H1299_scRNAseq. . . . . 41

2.14 A set of biological process that are enriched by marker genes in H1299_scRNAseq dataset. The numbers show the clusters and edges shows the link between a cluster and a biological process term. . . . . 45

2.15 Pathway that are enriched by marker genes in H1299_scRNAseq dataset. The numbers show the clusters and edges shows the link between a cluster and a pathway. Node that is highlighted yellow show the SARS-CoV-2 cell-specific pathway. Most of the other green nodes reveal the shared and cluster-specific functional pathways in the immune system. . . . . 46

3.1 Pipeline overview of the experiments. . . . . 55

4.1 Block diagram with the main components of proposed method. Step (1). The relationships among data points can be visualized in a way that similar points be placed in the same group in the resulting graph via Self-Organizing Maps. Step (2). The ”colored template” shows a representation of each cell based on the relationship among their marker genes. In this figure, two sample images are presented. We follow the standard color scheme of images in which the three color channels include red, green, and blue, where each channel is represented via 256 distinct values. In this work, we used gene expression values as color channels to color ”templates”. It is worth mentioning that due to varying gene expression values in different cells, the marker genes are not equally informative for all cells. Step (3). Detecting cell types in the classification step. The CNN uses the transformed images as inputs to classify cells into different cell types. . . . . 69

4.2 Distribution of genes in a read count matrix per total counts of reads. The number of genes expressed in the count matrix is mainly between 500 and 4,000 genes, and the distribution of several expressed genes over total count per cell is dense for less than 4,000 genes. As such, we filtered seven genes, i.e. the points above 4,000, to remove those low-quality cells. . . . . 71

4.3 Image created for one single sample from Class 2. Thirteen marker genes which found by the SOM are arranged in the GSN template after coloring. We follwo the standard color scheme of images in which the three color channels include red, green, and blue, where each channel is represented via 256 distinct values. In this work, we used gene expression values as color channels to color ”templates”. It is worth mentioning that due to varying gene expression values in different cells, the marker genes are not equally informative for all cells. . . . . 76

4.4 The plot of decreasing loss score by increasing the number of epochs ranging from 0 to 300. It can be observed that the model reaches its highest accuracy after 150 epochs. . . . . 77

5.1 Pipeline overview of the experiments. . . . . 84

5.2 Average performance of the SVM classifier combined with three feature selection methods. . . . . 93

5.3 Average performance of the *k*-NN classifier combined with three feature selection methods. . . . . 93

5.4 Average performance of the RF classifier combined with three feature selection methods. . . . . 93

6.1 Block diagram of the proposed approach. . . . . 100

6.2 (a) The number of expressed genes, (b) the total counts per cell, and (c) the percentage of mitochondrial genes. . . . . 102

6.3 Top 20 highly-variable genes before normalization. . . . . 103

6.4 Top 20 highly-variable genes after normalization. . . . . 103

6.5 Dispersion of genes before normalization. . . . . 104

6.6 Dispersion of genes after normalization. . . . . 104

6.7 *k*-means applied on two-dimensional Laplacian eigenmaps; outliers have been removed to enhance visualization. . . . . 112

6.8 *k*-means applied on two-dimensional ICA. . . . . 113

6.9 *k*-means clustering on three-dimensional Laplacian eigenmaps. . . . . 114

6.10 *k*-means clustering on three-dimensional MLLE. . . . . 115

6.11 Two-dimensional ICA + *k*-means clustering is performed on three-dimensional Laplacian eigenmaps data; outliers have been removed to enhance visualization. . . . . 116

6.12 Two-Dimensional ICA + *k*-means clustering performed on three-dimensional MLLE data; outlliers have been removed to enhance visualization. . . . . 116

# List of Tables


2.1 Datasets used in this work. . . . . 22

2.2 Parameters used for experiments. These are generated considering both dimensionality reduction and clustering together. . . . . 37

2.3 Silhoutte scores comparison of proposed method with other dimensionality reduction techniques. . . . . 38

2.4 Identified cell types for H1299_scRNAseq. . . . . 43

2.5 Identified cell types for Calu3_scRNAseq. . . . . 44

2.6 Identified cell types for Baron_human1 dataset. . . . . 44

3.1 Details of the datasets analyzed in this study. . . . . 55

3.2 Comparison of classification results for Data2. . . . . 61

3.3 Comparison of classification results for Data3. . . . . 62

3.4 List of 17 out of 20 overlapped genes between our top 20 ranked genes and pancreas gene sets (Data2). . . . . 63

3.5 List of 15 out of 20 overlapped genes between our top 20 ranked genes and pancreas gene sets (Data3). . . . . 64

3.6 Comparison of classification results for Data1. . . . . 64

3.7 List of eight gene sets correlated to the Pancreatic cell types of Data1 resulting from the GSEA analysis. . . . . 65

3.8 List of 9 out of 20 overlapped genes between our top 20 ranked genes and pancreas gene sets (Data1). . . . . 65

4.1 Summary of the Pancreas dataset. . . . . 70
4.2 Performance metrics of proposed method. . . . . 76
4.3 Muraro Pancreas Acinar Cell gene set along with gene description. . . . . 78
4.4 Muraro Pancreas Alpha Cell gene set along with gene description. . . . . 79

5.1 Details of the datasets studied in this work. . . . . 85
5.2 The best parameters for each method obtained using Bayesian Optimization for the datasets. . . . . 89
5.3 Classification accuracy obtained by three classification methods combined with feature selection methods through selected features for Data1. . . . . 92
5.4 Muraro Pancreas Endothelial Cell gene set. . . . . 94
5.5 Muraro Pancreas Ductal Cell gene set. . . . . 95
5.6 Travaglini Lung Ereg Dendritic Cell gene set. . . . . 96

6.1 Comparison of *k*-means clustering score using different dimensionality reduction techniques. . . . . 112
6.2 Results of manifold learning techniques followed by ICA and *k*-means clustering. . . . . 113
6.3 Cell types identified by our proposed method. . . . . 114
6.4 Marker genes found in similar diseases. . . . . 116

7.1 Details of the datasets used in this work including tissue, the accession number, the number of cell types, the number of cells, and the number of genes. . . . . 124

7.2 Comparison of our method with latent methods. . . . . 135

7.3 Comparison of our method with other methods. . . . . 136

7.4 Performance metrics of our method for the datasets. . . . . 137

# Chapter 1

# Introduction

Single-cell sequencing has emerged as a powerful technology, allowing researchers to capture cell information at a single-nucleotide resolution and analyze individual cells separately. With the generation of high-dimensional and sparse scRNA-seq datasets for various purposes, the field faces analytical challenges, particularly in addressing the sparsity and curse of dimensionality inherent in scRNA-seq data. While computational methods have been proposed to analyze scRNA-seq data, open problems persist, including the identification of cell sub-types and tissue-specific gene sets as well as interaction prediction among cells.

In a typical scRNA-seq data analysis workflow 1.1, the process begins with data pre-processing, involving quality control, cell filtering, normalization, and gene filtering. After the pre-processing stage, downstream analysis is employed to extract biological insights and understand the underlying biological system using interpretable models. In downstream analysis, dimensionality reduction techniques such as principal component analysis (PCA) and visualization methods such as t-Distributed Stochastic Neighbor Embedding (t-SNE) or Uniform Manifold Approximation and Projection (UMAP) are then applied to identify ma-

jor sources of variation and visualize the high-dimensional data. Also, clustering to group cells with similar expression profiles, and trajectory inference to reveal the developmental paths within dynamic datasets ae another forms of downstream analysis. Differential expression analysis also identifies genes with significant expression changes between clusters or along trajectories, and compositional analysis includes cell type annotation and functional enrichment analysis. Visualization tools aid in exploring gene expression patterns, and interactive platforms facilitate in-depth exploration.

Downstream analysis is categorized into cell- and gene-level approaches, Fig. 1.2, with cell-level analysis concentrating on cell-type clusters and trajectories. Cluster analysis aims to categorize cells into groups, explaining data heterogeneity, while trajectory analysis treats the data as a snapshot of a dynamic process, investigating the underlying process. These cellular structures can in turn be analysed on the cell and the gene level leading to cluster analysis and trajectory analysis methods. On the other hand, in gene-level analysis, we have gene set enrichment analysis (GSEA), gene regulatory network (GRN) analysis, and gene differential expression (DE) analysis.

## 1.1 Single-Cell RNA Sequencing

scRNA-seq is a powerful tool that allows for the detection and quantitative analysis of messenger RNA molecules in individual cells [66]. This technique has revolutionized our understanding of biological phenomena at the cellular scale [81].

An RNA-sequencing (RNA-seq) protocol involves several key experimental steps, from sample preparation to computational analysis; see Fig.1.3. The process begins with the collection of biological samples, such as tissues or isolated cells. Subsequently, tissues are dissociated into individual cells. The cells are then tagged or identified before sequencing

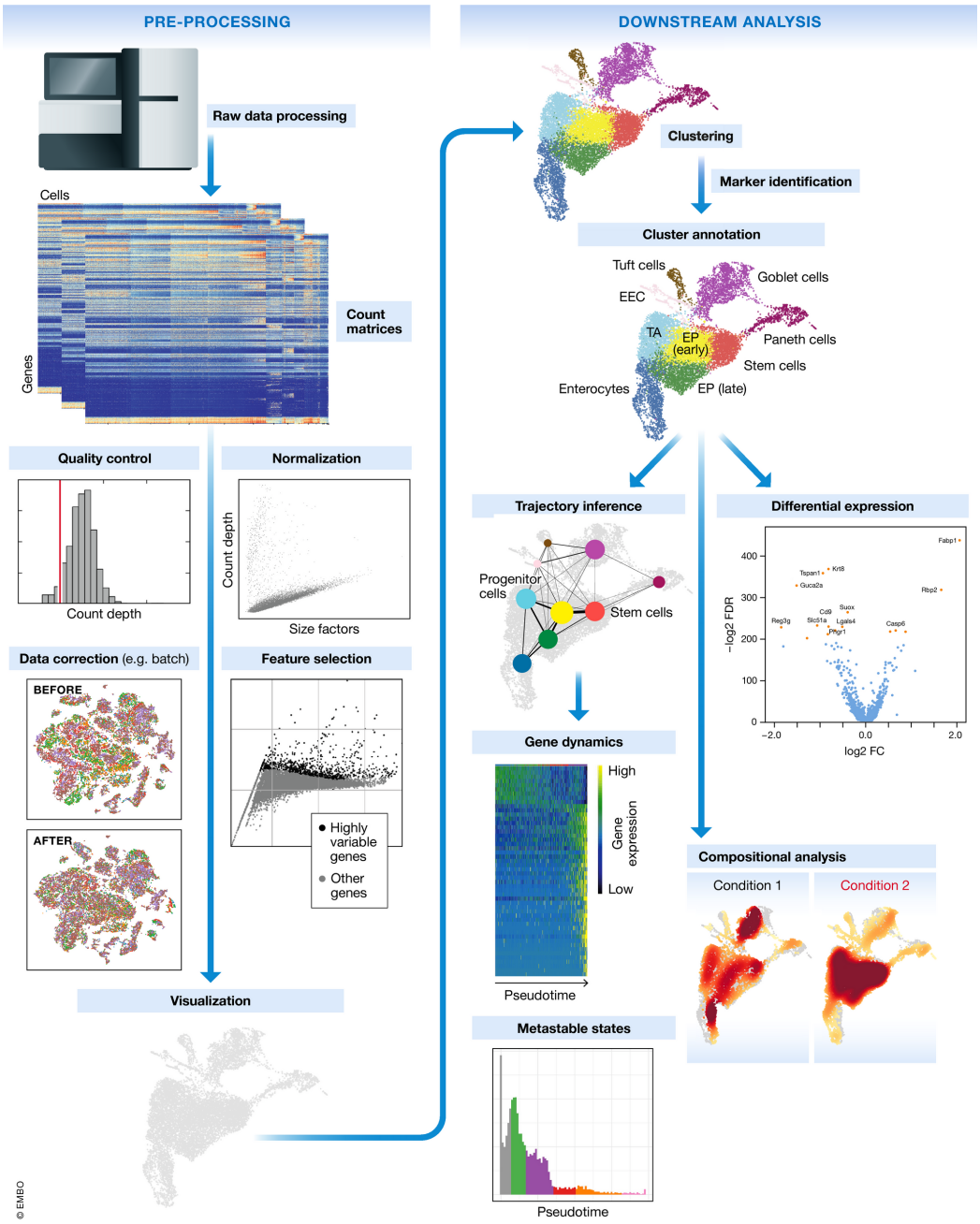


Figure 1.1: Schematic of a typical single-cell RNA-seq analysis workflow. [67]

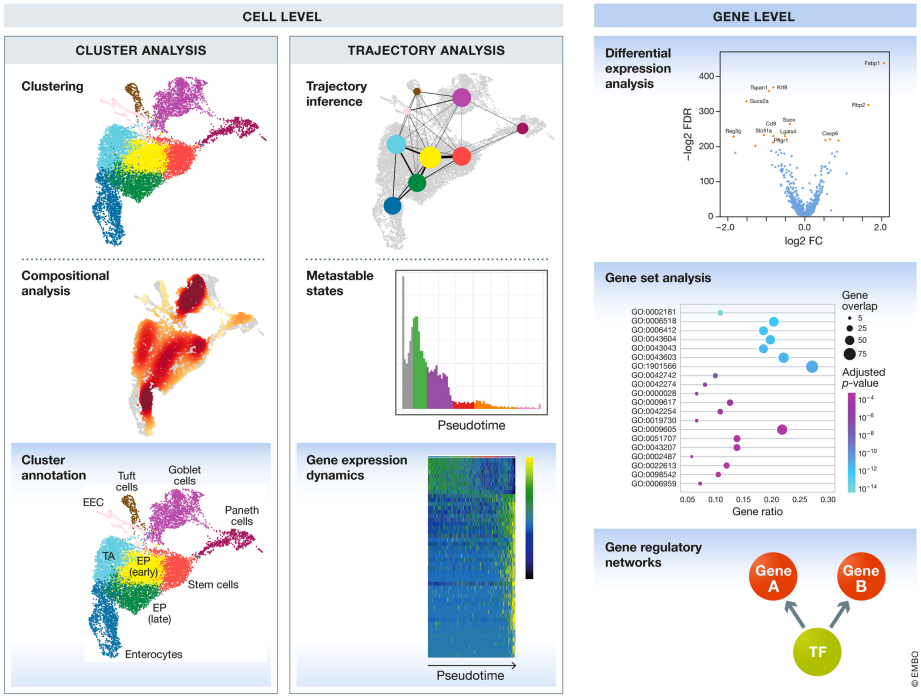


Figure 1.2: Overview of downstream analysis methods [68].

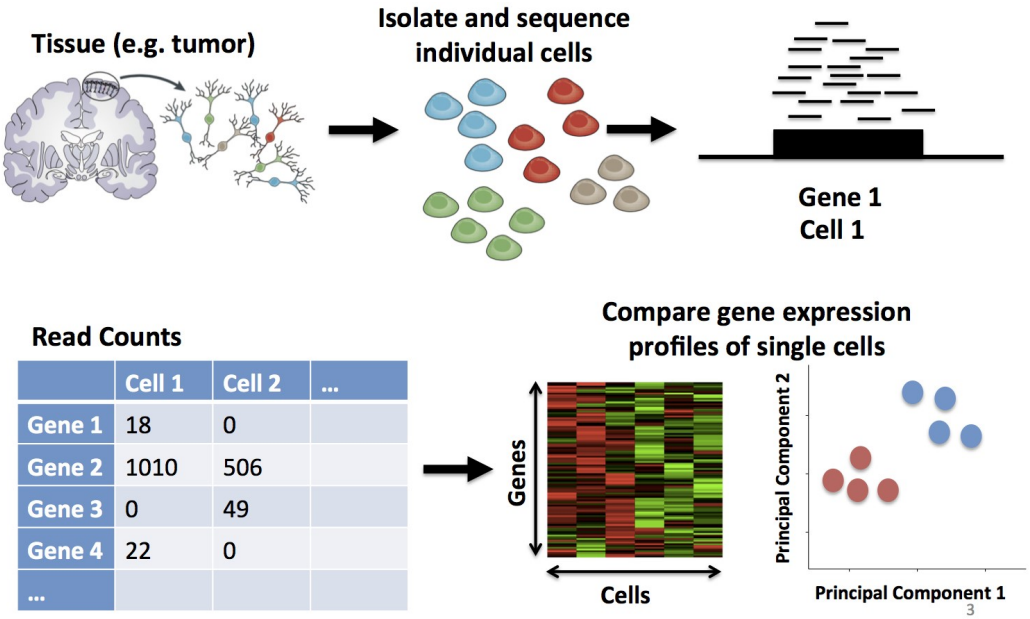


Figure 1.3: An overview of the experimental steps in an RNA sequencing protocol. [105]

takes place.

After isolation, total RNA is extracted from the cells using methods such as TRIzol or commercial kits. The quality and quantity of the extracted RNA are then assessed through techniques such as agarose gel electrophoresis or spectrophotometry. High-quality RNA is essential for reliable downstream analysis.

The next step involves library preparation, where extracted RNA is converted into a cDNA (complementary DNA) library suitable for sequencing. The resulting cDNA library is then amplified using polymerase chain reaction (PCR), and its quality is rigorously checked before sequencing.

Sequencing is performed on platforms such as Illumina Hiseq 2000, generating millions of short sequence reads from the cDNA fragments. These reads are mapped to a reference genome or transcriptome, and gene expression levels are quantified by counting the mapped reads. Various normalization methods are applied to account for differences in library size and sequencing depth.

Finally, functional analysis highlights the biological significance of these genes using analysis tools.

scRNA-seq has shifted the focus from measuring the average expression of tissue to measuring the specific gene expression of individual cells within those tissues. Instead of comparing tissue against tissue, the comparison is now cell against cell. This shift allows for the recognition of cell types and the identification of cells transitioning between states, providing a much clearer view of the dynamics of tissue and organism development [98].

With the increasing availability of scRNA-seq platforms and the rapid maturation of bioinformatics approaches, biomedical researchers or clinicians can use scRNA-seq to make significant discoveries [73]. scRNA-seq data has been widely used in various research

fields such as neuroscience, immunology, and oncology. One of the purposes of scRNA-seq data analysis is to recognize cell types, which is commonly achieved through clustering analysis. However, due to the existance of high noise, high dimensions, and increasing data scale of scRNA-seq data, clustering remains a significant challenge [111]. Moreover, it is important to note that while scRNA-seq provides a wealth of data, interpreting this data can be challenging. Therefore, a variety of computational tools and methods have been developed to assist researchers with tasks such as dimensionality reduction, clustering, and statistical analysis [75].

## 1.2 Cell Type Identification

Identification of cell types is crucial for interpreting scRNA-seq data and making connections between the transcriptome and phenotype [81]. For instance, scRNA-seq has been used to investigate the altered transcriptome of drug-resistant cells in triple-negative breast cancer [81]. It has also been used to identify potential targets for therapy by clarifying the cell type composition in tumors [81].

Cell type identification is a critical step in interpreting scRNA-seq data. With the accumulation of public scRNA-seq data, supervised cell type identification methods have gained increasing popularity due to their better accuracy, robustness, and computational performance [74].

These methods rely heavily on several key factors: feature selection, prediction method, and, most importantly, the choice of the reference dataset [74]. Accurate cell type identification is crucial for deciphering the mechanism behind numerous biological processes, such as the development of an embryo, the differentiation of stem cells, and the metastasis/recurrence/drug resistance of tumors [117]

The main challenges of clustering scRNA-seq data include the "curse" of dimensionality and the computationally intensive nature of geodesic computations in high-dimensional spaces. In addition, sparsity and noise in the data can affect the performance of algorithms [49]. These drawbacks, collectively regarded as nonbiological technical noise, pose a key challenge in scRNA-seq data analysis and interpretation [49].

To circumvent these drawbacks, pre-processing of scRNA-seq data, such as dimension reduction and normalization, is widely used. A noise reduction method, RECODE (resolution of the curse of dimensionality), has been proposed for high-dimensional data with random sampling noise. RECODE consistently resolves COD in relevant scRNA-seq data with unique molecular identifiers [49].

## 1.3 Computational Approaches for Cell Type Identification

### 1.3.1 Unsupervised Approaches

Unsupervised methods are often the first step in scRNA-seq data analysis. These methods aim to reduce the dimensionality of the data and identify clusters of cells without any prior knowledge [61]. A typical scRNA-seq data analysis workflow includes steps for reducing the number of gene features, creating a manifold representation for visualization, and unsupervised clustering to define discrete cell types [61]. Differential expression analysis is then performed to identify expression signatures of cell populations [61].

The primary goal of scRNA-seq is to identify cell types within a sample [61]. This is typically achieved through a process that involves unsupervised clustering, the identification of signature genes in each cluster, and then a manual lookup of these genes in the

literature and databases to assign cell types [61].

### 1.3.2 Supervised Approaches

With the accumulation of public scRNA-seq data, supervised cell type identification methods have gained increasing popularity due to better accuracy, robustness, and computational performance [73]. These methods rely on several key factors: feature selection, prediction method, and, most importantly, the choice of the reference dataset [73]. For instance, one study suggests combining all individuals from available datasets to construct the reference dataset and use a multi-layer perceptron (MLP) as the classifier, along with F-test as the feature selection method [73].

In recent years, machine learning models have been benchmarked for automatic cell identity assignment, evaluated based on classification accuracy and computation time [1]. Artificial intelligence techniques, including neural networks, have provided faster, more accurate, and user-friendly approaches for cell-type identification [75].

It is important to note that both unsupervised and supervised methods have their strengths and weaknesses, and the choice of method often depends on the specific research question and the nature of the data [81].

## 1.4 Dimensionality Reduction and Visualization

Dimensionality reduction and clustering are fundamental techniques in machine learning and data mining. Dimensionality reduction methods typically concern themselves with a reduction in the variable space through either selection of variables or the construction of new variables as combinations of the original ones [58]. Popular dimensionality reduction

techniques include Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), Uniform Manifold Approximation and Projection (UMAP), and Laplacian graph embedding, which are closely related to $k$-means clustering and spectral clustering, respectively [58].

### 1.4.1 t-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor Embedding (t-SNE) is an unsupervised nonlinear dimensionality reduction technique that is widely used for the visualization of scRNA-seq data [133]. It constructs a high-dimensional graph representation of the data, then builds a low-dimensional graph that is as structurally similar as possible [81]. The goal of t-SNE in single-cell studies is to place similar cells together and different cells further apart on a 2D or 3D plot [133]. However, when the results are shared as a report or published in a paper format (a static 2D image), it is only possible to see a snapshot of the analysis corresponding to a single gene and a single set of cell metadata [18].

### 1.4.2 Uniform Manifold Approximation and Projection

Uniform Manifold Approximation and Projection (UMAP) is another algorithm that takes a high-dimensional dataset (such as a single-cell RNA dataset) and reduces it to a low-dimensional plot that retains much of the original information. Similar to t-SNE, UMAP constructs a high-dimensional graph representation of the data, then builds a low-dimensional graph that is as structurally similar as possible. The goal of UMAP in single-cell studies is to place similar cells together and different cells further apart on a 2D or 3D plot. UMAP has rapidly become the most-used method by single-cell data analysts, mainly due to its computational speed and scalability to large datasets.

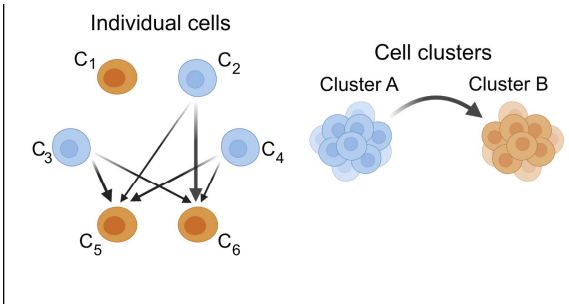


Figure 1.4: Cellular interaction among cells [4].

## 1.5 Prediction of Cell-Cell Interactions

Cell-cell interactions are fundamental for information exchange between different cells, which forms the basis of many biological processes [66]. Recent advances in single-cell RNA sequencing (scRNA-seq) enable the characterization of cell-cell interactions using computational methods [66]. However, evaluating these methods can be challenging since no ground truth is provided [66]. As shown in Fig. 1.4, cellular interaction can be deduced using scRNA-seq at either the individual cell or cell cluster level.

At the individual cell level, cells communicate through various mechanisms such as direct cell-cell contact, paracrine signaling, and the release of signaling molecules. Cell surface receptors and ligands play a key role in mediating these interactions, influencing cellular behavior, differentiation, and response to external stimuli. The communication between individual cells is vital for processes such as immune response, tissue development, and homeostasis.

On a broader scale, cells often organize into functional groups or clusters, each exhibiting distinct gene expression profiles and behaviors. These clusters represent specific cell types or states within a heterogeneous population. Interactions among these cell clusters contribute to tissue-level functions and responses. These interactions may involve coor-

dinated gene expression and physical interactions, influencing the overall behavior of the tissue or organ. Understanding the interplay between individual cells and the communication among cell clusters is fundamental to unraveling complex biological phenomena, ranging from embryonic development to disease progression. Advances in technologies such as single-cell RNA sequencing enable researchers to explore these interactions, providing insights into the cellular dynamics that underlie physiological and pathological processes.

## 1.6 Computational Methods for Cell-cell Interaction Prediction

### 1.6.1 Heuristic Methods

There are a variety of computational tools and resources to predict cell-cell communication using gene expression profiles obtained from scRNA-seq data. Traditional approaches include heuristic methods which use network structure, i.e. network topology information, in the prediction process. Existing algorithms can be classified based on the maximum hop of neighbours required to calculate the probability score of interaction. Also, some supervised approaches are used for connection prediction, including support vector machine (SVM), baggings, and naive Bayes. These are used to model the problem as a binary classification in which edge feature extraction is fundamental. Moreover, recent methods are mostly built on top of node embedding methods, with the edge representation constructed from the interaction between corresponding node embeddings. However, the performance of these methods is dependent on some assumptions according to the type of network and the inherent features of that specific link prediction problem. On the other hand, some graph-based approaches have been introduced to overcome this limitation. These approaches are based

on the local enclosing subgraph for a node pair ($u$, $v$) induced from the network by the union of $u$ and $v$'s neighbours up to $h$ hops or steps. The hop is the maximum distance that nodes can propagate features to their neighbours. These approaches give higher accuracy than heuristic and latent methods but require additional computation time and memory.

### 1.6.2 Graph-Based Methods

Link prediction is a field within network analysis that focuses on predicting missing or potential edges between nodes in a graph. Graph-based methods have emerged as powerful tools for predicting cell-cell interactions from scRNA-seq data. These methods typically involve constructing a graph where nodes represent cells and edges represent potential interactions between cells. The strength of an interaction can be inferred based on the expression levels of known interactions [76].

Some of such methods such as WLNM (Weisfeiler-Lehman Neural Machine), GAE (Graph Autoencoder), and VGAE (Variational Graph Autoencoder) contribute to this task by capturing structural information. WLNM integrates the Weisfeiler-Lehman graph isomorphism test into graph neural networks, refining node representations iteratively. GAE and VGAE leverage autoencoder architectures to learn node representations, with GAE focusing on reconstructing the adjacency matrix and VGAE introducing a variational approach for handling uncertainty in the learned representations. On the other hand, graph embedding methods such as Node2Vec, DeepWalk, and LINE aim to map nodes into continuous vector spaces, capturing local and global structures. These methods employ strategies such as random walks and spectral decomposition to create embeddings that reflect the structural similarities of nodes in the graph. Additionally, graph labeling methods such as Spectral Clustering and GraphSAGE provide an alternative approach by assigning la-

bels to nodes based on their structural properties, offering a richer representation of the graph. Overall, these techniques play a vital role in deciphering complex relationships within large-scale graphs and contribute to effective link prediction, particularly in scenarios where traditional methods may be limited.

An example of the graph-based methods in cell-cell interactions prediction is GNNLink which leverages known gene regulatory networks to deduce potential regulatory interdependencies between genes [76]. It uses a graph convolutional network-based interaction graph encoder to refine gene features by capturing interdependencies between nodes in the network [76].

Another method, DeepLinc, uses a deep generative model of variational graph autoencoder (VGAE) for the de novo reconstruction of cell interaction networks from single-cell spatial transcriptomic data [60].

Moreover, another group of graph-based methods in link prediction are based on the concepts of Graph Convolutional Networks (GCNs). GCNs are a type of neural network designed to work directly on graphs, which are a common form of data structure in many fields, including healthcare, social networks, transportation systems, and biology. GCNs operate by performing a series of graph convolutions, which apply a linear transformation to the feature vectors of each node and its neighbors. The output of each convolution is fed into a non-linear activation function and passed to the next layer [120]. One of such methods is SEAL (Learning from Subgraphs, Embeddings, and Attributes for Link Prediction) [129]. SEAL is also a subgraph method that addresses a number of weaknesses that WLNM has. To begin with, it enables learning not only from subgraph structures but also from latent and explicit node attributes, allowing it to incorporate a variety of information. Secondly, the fully-connected neural network in WLNM is replaced by a GNN that enables graph feature

learning improvement. SEAL derived $\gamma$ decaying theory and proved that a small number of hops is enough to extract high-order heuristics and outperform WLNM.

These graph-based methods provide a robust and scalable framework for predicting cell-cell interactions from scRNA-seq data, thereby enabling a deeper understanding of cellular interaction.

## 1.7 Motivation and Objective

For a precise cell-cell interaction prediction, we need precise cell type identification. The identification of cell types provides the necessary context for understanding cell-cell interaction. By knowing the types of cells involved in interaction, we can gain insights into the roles these cells play in various biological processes. On the other hand, cell-cell interaction can provide valuable clues for identifying cell types. The patterns of interaction between cells can help distinguish different cell types based on their interaction partners and the signals they exchange. Therefore, cell type identification and cell-cell interaction prediction are two intertwined topics, each enriching and informing the other. As shown in Fig. 1.5, clustering cells and identifying the correct groups of cells or cell types, is a crucial step in cell-cell interaction prediction frameworks.

## 1.8 Contributions

Identifying cell sub-types and clustering cells based on gene expression patterns are crucial steps, though traditional dimensionality reduction and clustering algorithms face inefficiencies in high-dimensional spaces. To overcome this problem, hybrid models combining dimensionality reduction and clustering techniques have shown promise. The first part of

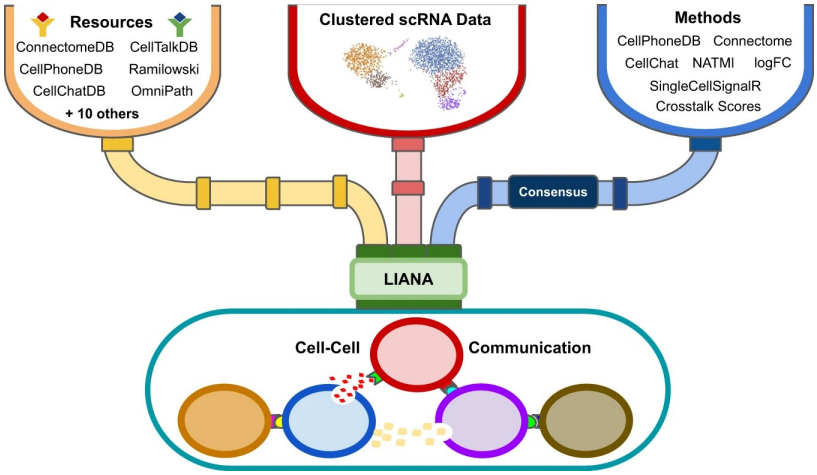


Figure 1.5: Cell-cell interaction analysis framework [28].

this research addresses the challenges of cell-type identification using the integration of unsupervised dimensionality reduction and conventional clustering in large-scale scRNA-seq data. Moving beyond cell clustering, the second part of this study delves into the realm of cell-cell interaction prediction. With the increasing importance of understanding intercellular signalling, graph convolutional neural networks for link prediction in scRNA-seq data are explored in this research. The key contributions of this thesis are as follows:

1. A novel pipeline for identification of cell types based on manifold learning and independent component analysis

   The proposed two-step representation learning approach, combining $k$-means clustering with Modified Locally Linear Embedding (MLLE), proved effective in untangling complex, hidden relationships in high-dimensional scRNA-seq data. MLLE demonstrated superiority over UMAP for dimensionality reduction when combined with clustering techniques, improving the visualization of cell clusters.

2. Leveraging sparsity-aware feature selection in automating cell-type annotation

By contrasting blind feature selection (pure information gain) with sparsity-aware inherent feature selection (GXBoost feature splitting algorithm), the research reveals that considering the data's latent sparsity pattern significantly boosts predictive model accuracy. Particularly impactful in the context of scRNA-seq data, where sparsity arises from technical and biological zeros, this approach enhances precision, speed, and meaningful biomarker selection.

3. A deep learning fusion of Self-Organizing Maps and convolutional neural networks for cell type identification

   Introduction of a deep learning approach using a combination of self-organizing map (SOM) and convolutional neural network (CNN) for simultaneous dimensionality reduction, feature selection, and classification. The proposed approach showcased potential as an unsupervised clustering algorithm, accommodating a large number of unlabeled samples alongside a small number of labeled scRNA-seq data for cell type identification on a larger scale.

4. Identification of SARS-CoV-2 target cell groups via nonlinear dimensionality reduction on single-cell RNA-Seq data

   The proposed two-step clustering method successfully identified SARS-CoV-2 target cell groups with over 90% accuracy, showcasing the potential application of manifold learning and clustering techniques beyond traditional cell type identification.

5. Prediction of cell-cell interaction using Graph Convolutional Networks (GCN)

   Introduction of a pipeline for cell-cell interaction prediction using Graph Convolutional Networks (GCN) demonstrated superior performance compared to previous

state-of-the-art techniques, opening new avenues for research in network-based analyses.

These key findings collectively contribute to a comprehensive understanding of scRNA-seq data analysis, offering methodologies for improved cell type identification, classification, and interaction prediction. The implications of this research extend beyond individual studies, providing a foundation for future investigations in diverse biological contexts.

## 1.9 Thesis Organization

This thesis presents a comprehensive study on cell type identification and cell-cell interaction prediction using scRNA-seq data. In the subsequent chapters, we delve into the methodologies used for cell type identification using clustering techniques and dimensionality reduction techniques, as well as cell-cell interaction prediction using graph neural networks. We also present the results of applying these methodologies to real-world scRNA-seq datasets, demonstrating their effectiveness in uncovering the complex landscape of cell types and interactions in biological systems.

# Chapter 2

# Discovering cell types using manifold learning and enhanced visualization of single-cell RNA-Seq data

## 2.1 Introduction

Single-cell sequencing is an emerging technology used to capture cell information at a single-nucleotide resolution and by which individual cells can be analyzed separately [41]. As of now, single-cell RNA-seq (scRNA-seq) datasets have been generated for different purposes [44]. However, these high-dimensional and sparse data lead to some analytical challenges. While many computational methods have been successfully proposed for analyzing scRNA-seq data, there are still some open problems in this research area. One of the main challenges is sparsity of data and the curse of dimensionality presented in scRNA-seq data. Also, performing well-defined pre-processing steps leads to enhance the quality of data and new biological insights. Analyzing scRNA-seq data can be divided into two main

categories: at the cell level and gene level. Finding cell sub-types or highly differentially expressed tissue-specific gene set is one of the common challenges at the cell level [91]. Arranging cells into clusters to find the data's heterogeneity is arguably the most significant step of any scRNA-seq data downstream analysis. This step could be used to distinguish tissue-specific sub-types based on identified gene sets. Indeed, cell clustering aims to identify cell types based on the patterns embedded in gene expression without prior knowledge at the cell level. Since the number of genes that are profiled in scRNA-seq data is typically large, cells tend to be located close to each other via non-metric distances, but rather complex relationships in high-dimensional spaces [55]. Therefore, traditional dimensionality reduction and clustering algorithms are unsuitable for these scenarios, and hence, they cannot efficiently separate individual cell types. Several algorithms have been proposed to lower the dimension of the data and cluster cells from scRNA-seq profiles to alleviate the problem of curse of dimensionality.

Dimensionality reduction techniques have been widely used in several studies of large-scale scRNA-seq data processing [29]. Most of the previous studies use principal component analysis (PCA). However, one of the main drawbacks of PCA is that it cannot deal with sparse matrices and non-metric relationships among high-dimensional data points. Also, there was no advantage in keeping the clustering performance after the changes in the data in lower dimensions [36]. Other works have also employed PCA as a pre-processing step to remove cell outliers for performing dimensionality reduction and visualization. Other methods proposed nonlinear dimensionality reduction methods such as t- distributed Stochastic Neighborhood Embedding (t-SNE) [106] and UMAP [77]. However, UMAP and t-SNE is not useful for high-dimensional cytometry. Moreover, several studies have used unsupervised clustering models to identify rare novel cell types. For instance, the hierarchical

clustering algorithm divides large clusters into smaller ones or merge each data points into larger clusters progressively. This algorithm has been employed to analyze scRNA-seq data by BackSPIN [125] and pcaReduce [124], through dimension reduction after each division or combination in an iterative manner. *k*-means, which is one of the most common clustering algorithms has been employed in the Monocle, specifically for analyzing scRNA-seq data [86]. Also, the authors of [118] used the Louvain algorithm, which is based on community detection techniques to analyze complex networks [42].

However, to achieve acceptable clustering performance on scRNA-seq data, other comprehensive studies indicated that hybrid models, designed as a combination of clustering and dimensionality reduction techniques, tend to improve the clustering results [36]. They learned 20 different models using four dimensionality reduction methods, including PCA, non-negative matrix factorization (NMF), filter-based feature selection (FBFS), and Independent Component Analysis (ICA). They also used five clustering algorithms as *k*-means, density-based spatial clustering of applications with noise (DBSCAN), fuzzy *c*-means, Louvain, and hierarchical clustering. Their experiments highlighted the positive effect of hybrid models and showed that using feature-extraction methods could be a decent way to improve clustering performance. Their experimental results indicate that Louvain combined with ICA performed well in small feature spaces.

In this paper, we proposed a model to obtain efficient and meaningful clusters of cells from large-scale scRNA-seq data. We focus on the combination of unsupervised dimensionality reduction followed by conventional clustering. We discovered a hybrid model of non-linear dimensionality reduction technique (MLLE) and linear combination method (ICA) for visualization and compared it to PCA, t-SNE, Isomap, regular Locally Linear Embedding (LLE), and Laplacian eigenmaps. ICA is employed to enhance visualization

and clustering of the data. Parameter tuning or choosing the best parameters for dimensionality reduction and clustering has been one of the main challenges in the field that is well addressed in our work. Experimental results on thirteen different benchmark scRNA-seq datasets show the power of modified LLE and ICA on clustering data and representation quality, providing very high accuracy and enhanced visualization. Confirmatory biological annotations were observed in the clusters using corresponding marker genes found by our method.

## 2.2 Materials and Methods

### 2.2.1 Datasets

To evaluate the performance of the proposed method, a total of thirteen benchmark scRNA-seq datasets were used, which include single-cell gene expression profiles. The details of all datasets used in this work are given in Table 5.1. They vary across size, tissue (pancreas, lung, peripheral blood), sequencing protocol (three different protocols), and species (Human and Mouse). Datasets Xin [122], H1299_scRNAseq [121], and Calu3_scRNAseq [121] datasets are unlabeled and do not have any background knowledge of the data. In this case, we analyzed the data and provided useful information about the unknown data. On the other hand, pancreas datasets including Baron [9] , Muraro [78], Segerstolpe [95], Xin [122], and Wang [116]. Moreover, peripheral blood dataset, 3k PBMC from a healthy donor, were downloaded from the 10XGenomics portal [37]. H1299_scRNAseq and Calu3_scRNAseq datasets (GSE148729) were extracted from NCBI's Gene Expression Omnibus [102].

Table 2.1: Datasets used in this work.

| Dataset | # cells | # genes | Accession # | Description | Sequencing technology |
|---|---|---|---|---|---|
| Baron_human1 | 16,381 | 1,937 | GSE84133 | Human pancreas | llumina HiSeq 2500 (inDrop) |
| Baron_human2 | 16,381 | 1,724 | GSE84133 | Human pancreas | llumina HiSeq 2500(inDrop) |
| Baron_human3 | 16,381 | 3,605 | GSE84133 | Human pancreas | llumina HiSeq 2500(inDrop) |
| Baron_human4 | 16,381 | 1,303 | GSE84133 | Human pancreas | llumina HiSeq 2500(inDrop) |
| Baron_mouse1 | 14,878 | 822 | GSE84133 | Mouse pancreas | llumina HiSeq 2500(inDrop) |
| Baron_mouse2 | 14,878 | 1,064 | GSE84133 | Mouse pancreas | llumina HiSeq 2500(inDrop) |
| Muraro | 17,140 | 3,071 | GSE85241 | Human Pancreas | Illumina NextSeq 500 (CEL-Seq2) |
| Segerstolpe | 26,271 | 7,028 | E_MTAB_506 | Human Pancreas | Smart-Seq2 |
| Xin | 39,851 | 1,601 | GSE81608 | Human Pancreas | Illumina HiSeq 2500(SMARTer) |
| Wang | 19,950 | 635 | GSE83139 | Human Pancreas | Illumina HiSeq 2000(SMARTer) |
| H1299_scRNAseq | 48,890 | 27,072 | GSE148729 | Human lung (SARS-CoV-2) | Illumina NextSeq 500 |
| Calu3_scRNAseq | 24,754 | 27,072 | GSE148729 | Human lung (SARS-CoV-2) | Illumina NextSeq 500 |
| PBMC | 32,738 | 2,700 | 10X Genomics (pbmc3k) | 3k PBMCs from a Healthy Donor | Cell Ranger |

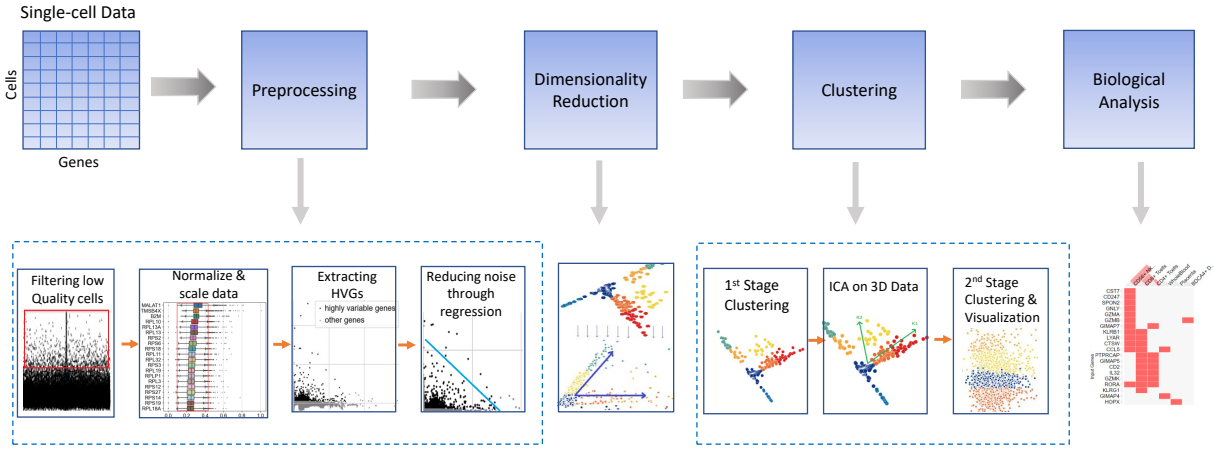


Figure 2.1: Block diagram of the proposed approach for discovering cell types in scRNA-seq data.

### 2.2.2 Data Pre-processing and Quality Control

A common practice for generating RNA-seq raw data is to use next-generation sequencing technologies to create read count matrices. The read count data matrix contains gene names and their expression levels across individual cells. Before analyzing scRNA-seq data, one needs to ensure that gene expressions and cells are of standard quality. We follow a typical scRNA-seq analysis workflow including quality control, as described in [69] [48]. Based on the expression levels, we filtered out weakly expressed genes and low-quality cells in which fewer reads are mapped, as shown in Fig. 6.1, the first step of pre-processing. Low-quality cells that are dyed, degraded, or damaged during sequencing are represented by a low number of expressed genes. Genes expressed in less than three cells and cells with less than 200 expressed genes are removed. This step is performed to remove low quality cells and poorly expressed genes.

We also investigated the distribution of the data (Fig.6.2) as a data-specific quality-control step and filtered out low-quality cells and genes. Also, we remove a percentage of mitochondrial genes that do not contribute significant information to the downstream

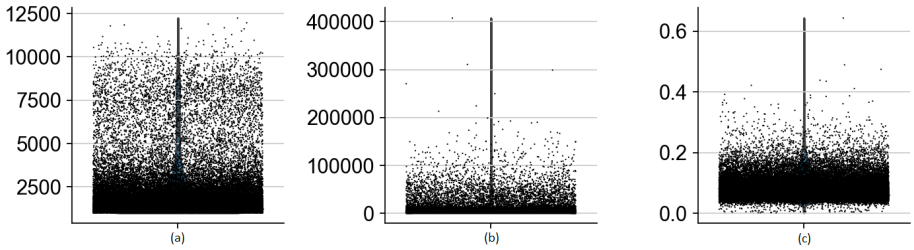


Figure 2.2: Investigating the distribution of the data to filtered out weakly expressed genes and low-quality cells from dataset; (a) number of expressed genes, (b) total counts per cell, and (c) the percentage of mitochondrial genes for H1299_scRNAseq.

analysis [50], [48].

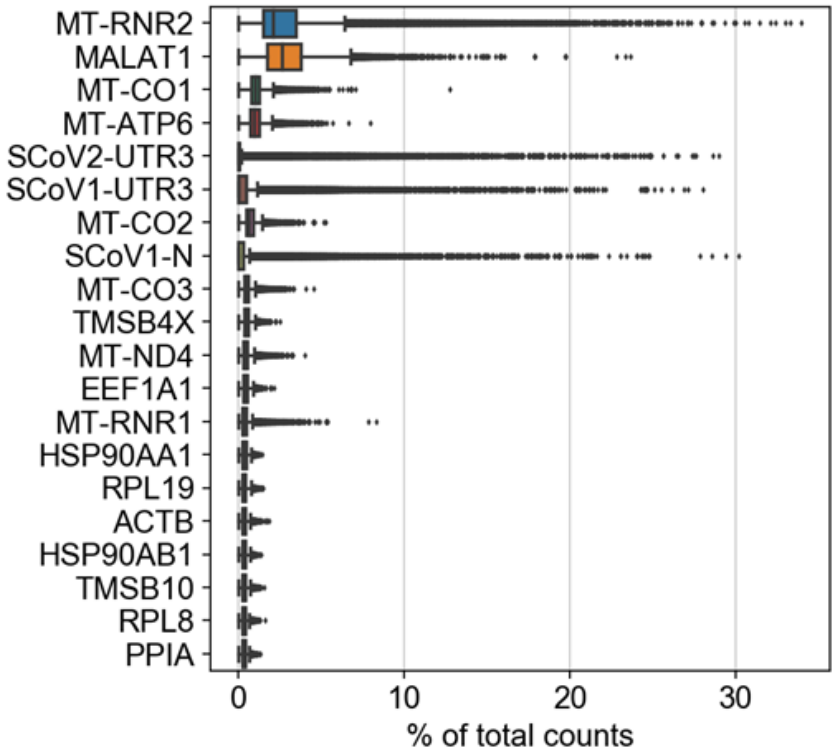


Figure 2.3: Top 20 highly-variable genes before normalization.

Since the scRNA-seq data expressed at different levels, normalization is a must. Normalization is the method of translating numeric columns' values in a dataset to a standard

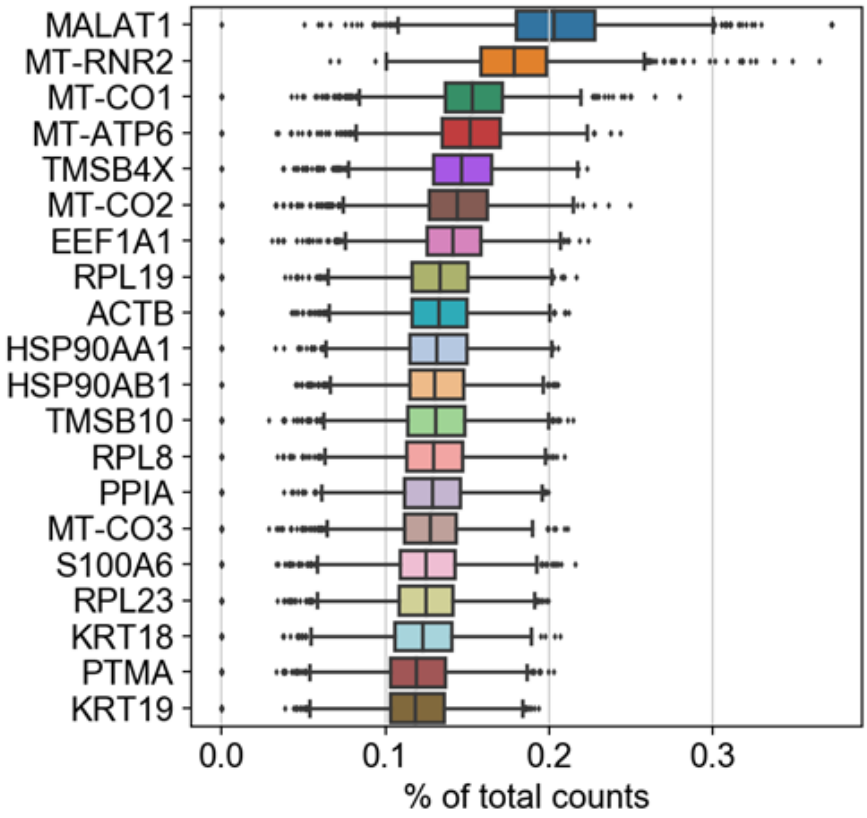


Figure 2.4: Top 20 highly-variable genes after normalization.

scale without distorting the ranges of values. Visualization of top genes in the dataset are shown in Figures 2.3 and 2.4 before and after normalization, respectively. We normalize the data using the Counts Per Million (CPM) normalization combined with logarithmic scaling on the data:

$$CPM = readsMappedToGene \times \frac{1}{totalReads} \times 10^6 \tag{2.1}$$

where $totalReads$ is the total number of mapped reads of a sample, and $readsMappedToGene$ is the number of reads mapped to a selected gene.

At this point, we extracted highly variable genes (HVGs) as a part of the feature selection

step, aiming at minimizing the search space, and only these genes are examined in further evaluation. We then removed any random noise and held genes that highlight relevant biological information. HVGs are those genes that are expressed significantly more or less in some cells compared to other ones. This step in quality control makes sure that the differences occur because of biological differences and not technical noise. The simplest approach to compute such a variation is to quantify the variance of the expression values for each gene across all samples. A good trade-off between mean and variance would help select the subset of genes that keep useful biological knowledge, while removing noise. We use log-normalized data because we want to ensure having the same log-values in the clustering and dimensionality reduction follow a consistent analysis through all steps. There are several widely-used approaches to find the best threshold. The normalized dispersion is obtained by scaling the mean and standard deviation of the dispersion for genes falling into a given bin for the mean expression of genes (Fig. 2.7). This means that for each bin of mean expression, HVGs are selected. A Python package, Scanpy, is used to perform pre-processing and quality control steps.

### 2.2.3 Dimensionality Reduction

The majority of real-life data is multidimensional. Furthermore, the majority of the high-dimensional data is complex and sparse. Most importantly, understanding the data in such dimensions is tricky, and visualization is not possible. Dimensionality reduction is the process of transforming data from a high-dimensional space to a low-dimensional space while retaining some of the original data's meaningful properties, preferably close to its intrinsic dimension. Working in high-dimensional spaces may be inconvenient for various reasons: raw data is often sparse as a result of the curse of dimensionality, and data analysis is

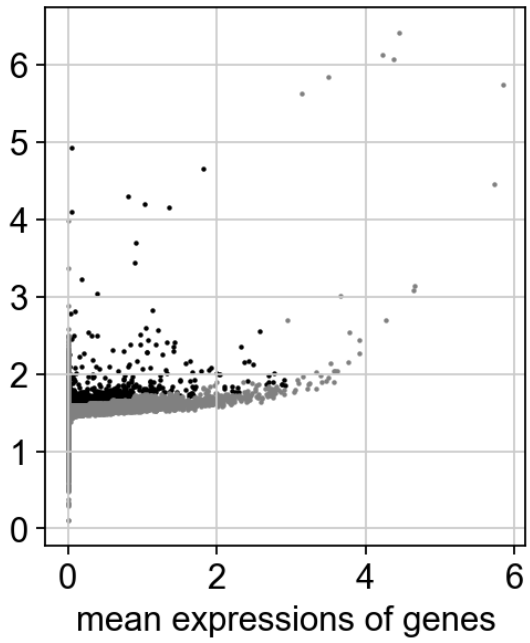


Figure 2.5: Dispersion of genes before normalization.

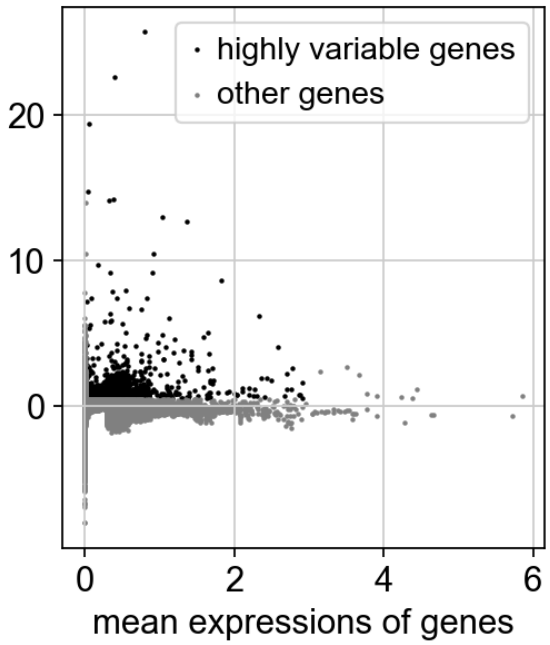


Figure 2.6: Dispersion of genes after normalization.

Figure 2.7: Comparison of dispersion of normalized and not normalized genes to extract highly variable genes.

typically computationally intractable. On the other hand, high-dimensional gene expression data is complex and should be well-explored. Each gene is characterized as a data dimension in a single-cell expression profile in a single-cell expression profile. As such, dimensionality reduction is very productive in summarizing biological attributes in fewer dimensions. Dimensionality reduction is divided into linear and non-linear techniques.

**Modified Locally Linear Embedding**

MLLE is the enhanced version of LLE and hence the authors named it as Modified LLE. To understand the working of MLLE, we need to understand LLE. LLE tries to reveal the manifold's underlying structure based on simple geometric intuitions when used for dimensionality reduction. LLE preserves the data's locality in lower dimensions because it reconstructs each sample point from its neighbors. In the simplest formulation of LLE, one identifies nearest neighbors per data point, as measured by Euclidean distance [94]. One can choose number of neighbors based on some rules or using some metrics or some random number. Consider the sample points $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ in high dimensional space, where $\{\mathbf{x}_j, j \in N\}$ and $\mathbf{W} = \{w_{ij}\}$ is the weight matrix. A directed graph $\mathcal{G} = (\mathbf{X}, \mathbf{E}, \mathbf{W})$ is constructed considering the neighborhood relations of the sample points $\mathbf{X}$, in high dimensional space, and $\mathbf{E} = \{e_{ij}\}$ represents the edges of the graph. Later, weights are assigned to the edge of the graph. To compute the weights $\mathbf{W_{kn}}$, minimize the cost function with respect to two constraints: 1) each data points $\mathbf{x_i}$, is reconstructed only from its neighbors imposing $\mathbf{W_{kn}} = \mathbf{0}$ if $\mathbf{x_i}$ does not belong to that set, 2) sum of the weights matrix rows equal to one, that is $\mathbf{W_{kn}} = 1$. Optimal weights are calculated by solving (2.2) the constrained squared distances problem shown below [94].

$$\min \ \mathbf{x}_i - \sum_{k \in K_n} w_{kn} \mathbf{x}_k \quad \text{s.t.} \sum_{k \in K_n} w_{kn} = 1 \,. \tag{2.2}$$

The computed weights are then allocated to each edge of the graph, with each data point viewed as a small linear patch of the sub-manifold.

Finally, each high-dimensional input sample $\mathbf{x_i}$ mapped to a low dimensional point set $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n\}$ representing the manifold's global internal coordinates. The reconstruction weights for each data point are calculated independently of the weights for other data points from its local neighborhood. The embedding coordinates are computed by an *NXN* eigen solver, a global operation that combines all data points in connected components of the graph identified by the weight matrix. While reconstructing the structure from the higher dimension to the lower dimension, some information could be lost. This lost information is noted as a reconstruction error and computed using (2.3).

$$\varepsilon_r = \sum_{i=1}^{n} |\mathbf{y}_i - \sum_{k \in K_i} w_{ik} \mathbf{y}_k|^2 \tag{2.3}$$

The regularization problem is a well-known issue with LLE. The matrix representing each local neighborhood is rank-deficient when the number of neighbors exceeds the number of input dimensions. To deal with this, standard LLE uses an arbitrary regularisation parameter in relation to the weight matrix's local trace [132]. This problem manifests itself in embedding which distort the underlying geometry of the manifold. MLLE is one such technique, which overcomes this regularization problem using multiple weights in each neighborhood. MLLE modifies or adjusts the reconstruction weights [34] shown in (2.2)

and this modifies the objective function (2.3).

$$\varepsilon_r = \sum_{i=1}^{n} \sum_{l=1}^{s_i} |\mathbf{y}_i - \sum_{k \in K_i}^{w} {}_{ik}\mathbf{y}_k|^2 \tag{2.4}$$

where, $s_i =$ smallest right singular vectors of $\mathbf{G}$.

MLLE aims to take advantage of the dense relations that exist in the embedding space. It is closely related to the other version of the LLE, that is Local Tangent Space Alignment (LTSA) [112].

### Independent Component Analysis

ICA is an independent and linear dimensionality reduction method. By using simple statistical properties assumptions, ICA learns an efficient linear transformation of the data and attempts to find the underlying components and sources present in the data [46]. Unlike other approaches, the transformation's underlying vectors are presumed to be independent of one another. It employs a non-Gaussian data structure, which is crucial for retrieving the transformed underlying data components. Consider, $\mathbf{r}$ is a random vector whose elements are $\{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n\}$, and similarly, random vector $\mathbf{s}$ with its elements $\{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_n\}$, and $\mathbf{A}$ is the matrix with elements $a_{ij}$. The ICA model is a generative model, and it explains how the observed data are generated (2.5) by mixing the components $s_i$. The independent components are latent variables, which means they are unknown. Also, the mixing matrix is assumed to be unknown.

$$\begin{aligned} \mathbf{r} &= \mathbf{A}\mathbf{s} \\ \mathbf{Y} &= \mathbf{A}\mathbf{X} \end{aligned} \tag{2.5}$$

Rows of these vectors and the matrix are orthogonal to each other. As such, it leads to more informative components than PCA. ICA does not require knowing the system's output to break the data into some measurements. Hence it is referred to as blind source separation [47]. Here, a source means the original data, independent components. Blind means that it knows nothing but very little, if anything, on the mixing matrix and makes modest assumptions on the source data.

**Other Dimensionality Reduction Methods**

We used other dimensionality reduction techniques to compare our proposed method such as Standard LLE, Isomap, Laplacian eigenmap, PCA, and t-SNE. Isomap stands for isometric mapping. Isomap is a non-linear dimensionality reduction method based on the spectral theory that tries to preserve the lower dimension's geodesic distances. Isomap starts by creating a neighborhood network. After that, it uses graph distance to estimate the geodesic distance between all pairs of points. The eigenvalue decomposition of the geodesic distance matrix finds the low-dimensional embedding of the data [38]. The Laplacian eigenmaps is a computationally effective and map nearby input patterns to nearby outputs by computing the low-dimensional representation of a high-dimensional data set that most faithfully preserves proximity relations and it has a natural connection with clustering [13]. PCA is a popular linear technique used for feature extraction or dimensionality reduction. Given a set of data with n dimensions, PCA maps the data linearly to find a subspace in lower-dimensional space so that variance of the data is maximized. It does so by calculating the eigenvectors from the covariance matrix. The principal components (eigenvectors that correspond to the largest eigenvalues) are used to recreate a substantial portion of the original data's variance [39]. t-SNE is a non-linear dimensionality reduction technique. t-SNE is not

used for cluster analysis or outlier detection since it does not preserve the data's distances or densities. But, it is particularly well suited for the visualization of high-dimensional datasets and extensively applied in image processing, Natural language processing, genomic data, and speech processing [106].

### 2.2.4 Clustering

Performing clustering is one of the critical tasks in single-cell analysis. Clusters are formed by grouping cells based on their similarity of the gene expression profiles. Distance metrics are used to describe expression profile similarity, which employs dimensionality-reduced representations as data as input. We used popular clustering technique $k$-means. $k$-means is iterative clustering algorithm groups the data into $n$ separate groups by minimizing the phenomenon within-cluster dispersion. The number of clusters $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n\}$ to be formed from the data needs to be specified as an input to the algorithm.

$$SSE = \sum_{i=1}^{k} \min_{\mu \in C} (|\mathbf{x}_i - \mathbf{x}_j|)^2 \tag{2.6}$$

$k$-means algorithm works in three key steps. The first step is to choose the initial centroids and the simple method is to choose $k$ samples from the dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. Then, each point in the dataset is allocated to its nearest centroid. The next step involves taking the mean value of all of the samples allocated to each previous centroid and creating new centroids. The algorithm calculates the difference between the old and new centroids, then repeats the last two steps until the value falls below a certain threshold. In other words, it keeps repeating until the centroids are converged. The points in the data choose centroids with a high degree of cluster compactness or a minimum sum of squared error ($SSE$) as shown in (2.6) where $n$ is the number of samples in the data, $C$ is the cluster, $\mu$ is the mean

of the samples, and $x$ is the corresponding sample.

### 2.2.5 Cluster Annotation

Gene Set Enrichment Analysis (GSEA) [99] is a computational tool that determines whether a predefined set of genes shows a statistically significant level of expression in a specific cell type, biological process, cellular component, molecular function, or biological pathway. The GSEA uses MSigDB, the Molecular Signature Database, to provides different gene sets for the analysis with the gene set enrichment analysis. To annotate the cell clusters, we first extracted the top 20 differentially expressed genes as markers in each cluster per dataset. Then, we found the corresponding cell types of each group of marker genes in each cluster. Gene ontology (GO) analysis is also used as part of enrichment analysis.

### 2.2.6 Parameter Optimization

With the aim of preserving locality, the number of nearest neighbors ($t$-NN) to construct the neighborhood graph is a crucial parameter in manifold learning techniques. Another critical step in any clustering algorithm is determining the number of clusters, $k$. We used the nearest neighbor check and validity indices check, which runs through different $t$ and a distinct number of clusters to find the best dimensionality reduction and clustering parameters. We further systematically evaluated more appropriate parameters for MLLE after finding the best $t$. The nearest neighbors are examined between the range of 8 and 26. The number of clusters $k$ for each value of $t$ is also assessed, where $k$ ranges from 4 to 15, and the validity of indices are calculated for each cluster. We select a combination of $t$ and the number of clusters with the highest number of clustering scores considering all the three validity of indices explained in the performance evaluation section.

### 2.2.7 Performance Evaluation

Generally speaking, the best clustering is the one that maintains high intra-cluster distance and gives the most compact clusters. In this work, we use the Silhouette coefficient [89], an evaluation metric that measures either the mean distance between a sample point and all other points in the same cluster or all other points in the next nearest neighbor cluster. Consider a set of clusters $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_k\}$, output by a clustering algorithm, k-means in our case. The Silhouette coefficient, $SH$, for the $i^{th}$ sample point in cluster $\mathbf{C}_j$, where $j = 1, ..., k$, can be defined as follows:

$$SH(\mathbf{x}_i) = \frac{b(\mathbf{x}_i) - a(\mathbf{x}_i)}{max(a(\mathbf{x}_i), b(\mathbf{x}_i))}, \tag{2.7}$$

where $a$ is the mean distance between point $\mathbf{x}_i$ and all other points inside the cluster (intra-cluster distance) and $b$ is the minimum mean value of the distance between a sample point $\mathbf{x}_i$ and the nearest neighbor cluster, and are calculated as:

$$\begin{aligned} a(\mathbf{x}_i) &= \frac{1}{|\mathbf{C}_k| - 1} \sum_{\mathbf{x}_j \in \mathbf{C}_k, i \neq j} d(\mathbf{x}_i, \mathbf{x}_j) \\ b(\mathbf{x}_i) &= \min_{k \neq i} \frac{1}{|\mathbf{C}_k|} \sum_{j=1}^{k} d(\mathbf{x}_i, \mathbf{x}_j). \end{aligned} \tag{2.8}$$

We also used Calinski-Harabasz (CH) and Davies-Bouldin (DB) validity of indices to assess the clustering performance. Calinski-Harabasz score [19], is a score used to evaluate the model where a higher score tells better-defined clusters. CH score is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters that is as

follows:

$$CH = \frac{tr(\mathbf{S}_B)}{tr(\mathbf{S}_W)} \times \frac{n-k}{k-1} \tag{2.9}$$

in which $n$ is size of input samples, $tr(\mathbf{S}_B)$ is the trace of the between-group dispersion matrix and $tr(\mathbf{S}_W)$ is the within-cluster dispersion.

Davies-Bouldin (DB) index [24] is another validity index defined as the average of the similarity measure of each cluster. DB is computed as follows:

$$DB = \frac{1}{k} \sum_{i=1}^{k} max_{i \neq j} s_{ij}\,, \tag{2.10}$$

where $s_{ij}$ is the ratio between within-cluster distances and between cluster distances, and is calculated as $s_{ij} = \frac{w_i + w_j}{d_{ij}}$. The smaller DB value the better clustering, and as such, we aim to minimize Equation (6.9). Here, $d_{ij}$ is the Euclidean distance between cluster centroids $\mu_i$ and $\mu_j$, and $w_i$ is the within-cluster distance of cluster $\mathbf{C}_k$.

Overall, we used the Silhouette score to evaluate the clustering performance, whereas CH and DB indices were used to verify and find the optimal parameters, namely the best number of clusters.

## 2.3 Results and Discussion

We developed a well-constructed pipeline that can be applied to single-cell data to discover individual cell types. Considering dimensionality reduction and clustering as two significant steps in the pipeline, we conducted many experiments on different dimensionality reduction techniques and explored many ways of untangling the data in two and three dimensions. We found optimum parameters for both dimensionality reduction and clustering to achieve the best clustering results. To demonstrate the applicability of our pipeline, we

tested it on thirteen datasets of different sizes. We evaluated our method in terms of both computationally and biologically perspectives to achieve the meaningful separation of cell types.

### 2.3.1 Clustering and Cell Type Discovery

To achieve the best results, we experimented with all possible combinations of parameters as discussed in the Material and Methods section. As a result, the best parameters chosen for each dataset are depicted in Table 2.2. In a few datasets, to achieve the best clustering score in the proposed approach, the data is reduced to lower dimensions such as 5, 6, and 7. Then, the data is reduced to three dimensions to visualize and obtain better results. When applying MLLE, a neighborhood graph is created by connecting points that are close to each other. Different measures are used for this purpose, including number of neighbors, distance from each point to its neighbors, and others. A common measure to determine the sparsity of the neighbor graph is the tolerance factor, which makes the graph sparser or denser. In this regard, we tested different tolerance values on each dataset and selected those values that yielded the best validity index scores. The results of $k$-means clustering combined with each dimensionality reduction method using the best parameters are listed in Table 6.1. The last column shows the result after applying ICA on the result of clustering combined with MLLE. The clustering score ranges from 0 to 1. A score close to 1 represents good quality clustering, with 1 being the best, while a score near zero indicates that the clusters are not well defined.

When trying widely-used techniques such as t-SNE and PCA, we noticed that both methods were not as efficient in separating the data into well-defined clusters. To show the clustering results graphically, we visualize the result of PCA and t-SNE for Wang and Mu-

Table 2.2: Parameters used for experiments. These are generated considering both dimensionality reduction and clustering together.

| **Dataset name** | **# Neighbors** | **# Dimensions** | **Tolerance** | **# Clusters** |
|---|---|---|---|---|
| Baron_human1 | 10 | 6 | 1e-12 | 14 |
| | 23 | 3 | 1e-10 | |
| Baron_human2 | 8 | 3 | 1e-12 | 14 |
| Baron_human3 | 16 | 7 | 1e-12 | 14 |
| | 8 | 3 | 1e-8 | |
| Baron_human4 | 9 | 6 | 1e-12 | 14 |
| | 22 | 3 | 1e-12 | |
| Baron_mouse1 | 17 | 3 | 1e-12 | 13 |
| Baron_mouse2 | 11 | 6 | 1e-12 | 13 |
| | 20 | 3 | 1e-8 | |
| Muraro | 10 | 5 | 1e-3 | 6 |
| | 11 | 3 | 1e-7 | |
| Segerstolpe | 10 | 5 | 1e-3 | 6 |
| | 9 | 3 | 1e-8 | |
| Xin | 15 | 6 | 1e-12 | 6 |
| | 25 | 3 | 1e-3 | |
| Wang | 8 | 3 | 1e-12 | 6 |
| H1299_scRNAseq | 11 | 3 | 1e-8 | 7 |
| Calu3_scRNAseq | 12 | 7 | 1e-3 | 7 |
| | 11 | 3 | 1e-5 | |
| PBMC | 8 | 5 | 1e-12 | 8 |
| | 25 | 3 | 1e-12 | |

Table 2.3: Silhoutte scores comparison of proposed method with other dimensionality reduction techniques.

| **Dataset name** | **t-SNE** | **PCA** | **Isomap** | **SLLE** | **Eigenmap** | **MLLE** | **MLLE+ICA** |
|---|---|---|---|---|---|---|---|
| Baron_human1 | 0.244 | 0.364 | 0.498 | 0.524 | 0.839 | **0.908** | 0.904 |
| Baron_human2 | 0.231 | 0.428 | 0.543 | 0.614 | 0.823 | **0.906** | 0.905 |
| Baron_human3 | 0.243 | 0.377 | 0.522 | 0.467 | 0.826 | **0.990** | 0.976 |
| Baron_human4 | 0.239 | 0.424 | 0.614 | 0.538 | 0.896 | 0.910 | **0.912** |
| Baron_mouse1 | 0.231 | 0.400 | 0.422 | 0.448 | 0.472 | 0.881 | **0.917** |
| Baron_mouse2 | 0.221 | 0.414 | 0.530 | 0.684 | 0.779 | 0.941 | **0.943** |
| Muraro | 0.258 | 0.494 | 0.532 | 0.738 | 0.913 | 0.933 | **0.944** |
| Segerstolpe | 0.231 | 0.410 | 0.399 | 0.400 | 0.537 | **0.960** | 0.956 |
| Xin | 0.242 | 0.445 | 0.481 | 0.494 | 0.751 | **0.899** | 0.888 |
| Wang | 0.230 | 0484 | 0.442 | 0.745 | 0.608 | 0.993 | **0.996** |
| H1299_scRNAseq | 0.245 | 0.269 | 0.701 | 0.683 | 0.782 | 0.938 | **0.943** |
| Calu3_scRNAseq | 0.361 | 0.232 | 0.494 | 0.452 | 0.798 | 0.889 | **0.924** |
| PBMC | 0.244 | 0.401 | 0.622 | 0.621 | 0.632 | 0.867 | **0.876** |

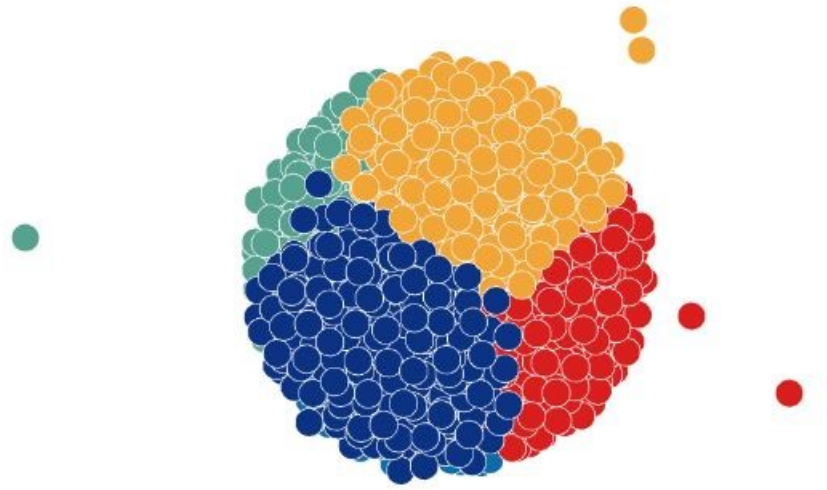

Figure 2.8: Visualization of t-SNE on Muraro dataset

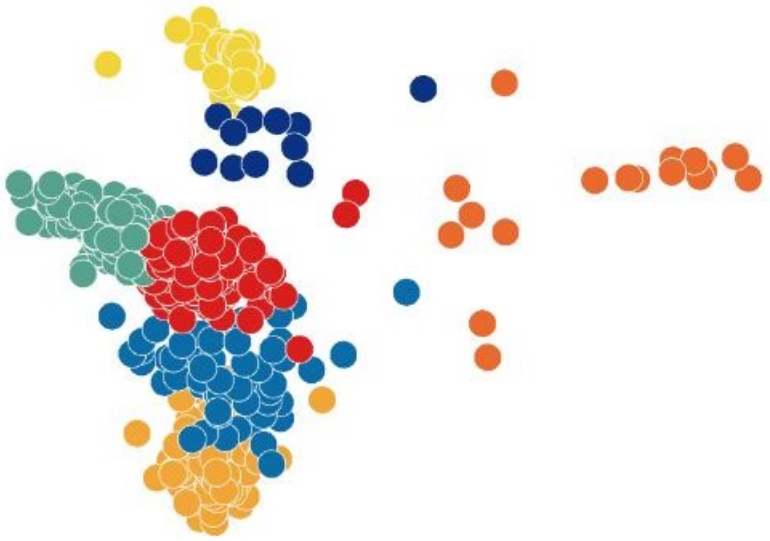

Figure 2.9: Visualization of PCA on Wang dataset

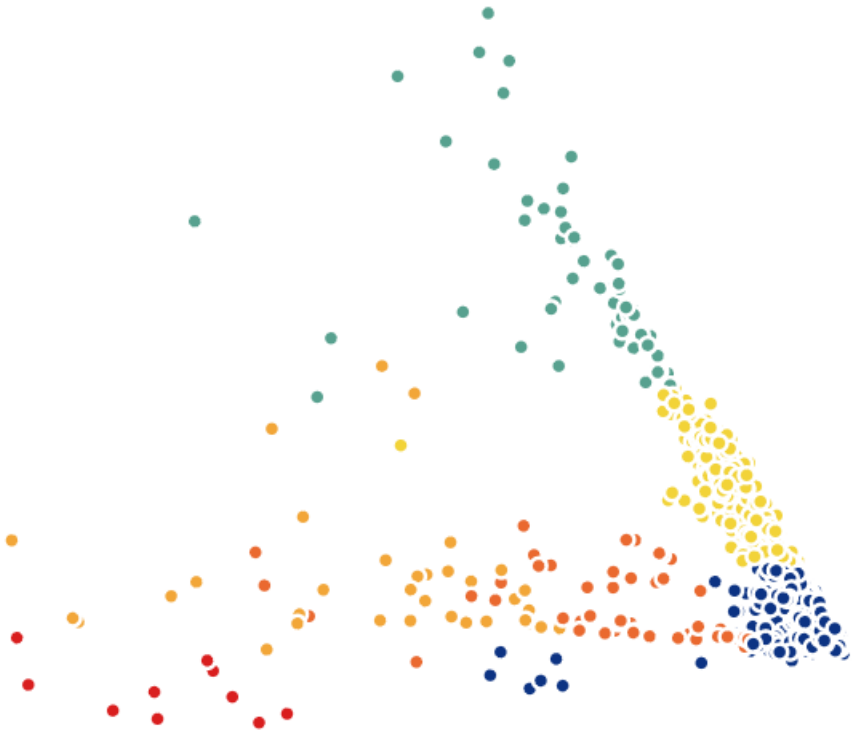

Figure 2.10: Visualization of Laplacian eigenmaps on H1299_scRNAseq; outliers have been removed to enhance visualization.

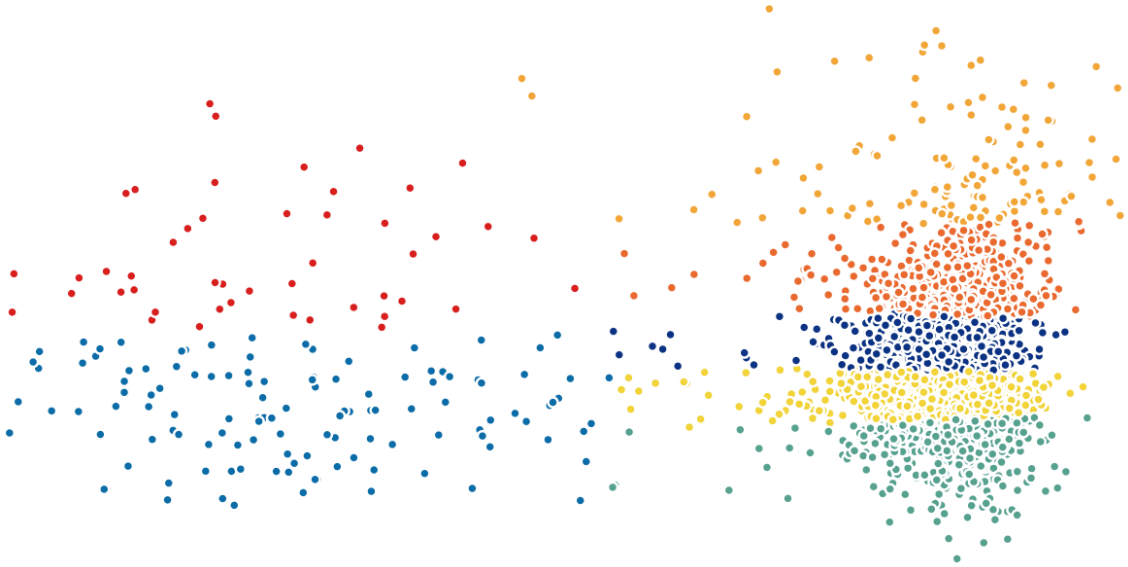

Figure 2.11: Two-dimensional ICA projection of cells colored by $k$-means clustering applied on high-dimensional original data (H1299_scRNAseq).

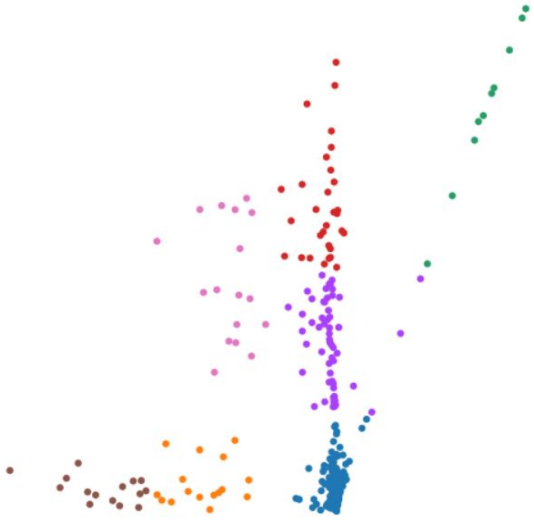

Figure 2.12: Two-dimensional ICA projection of cells colored by $k$-means clustering applied to the three-dimensional points output by MLLE on the H1299_scRNAseq dataset.

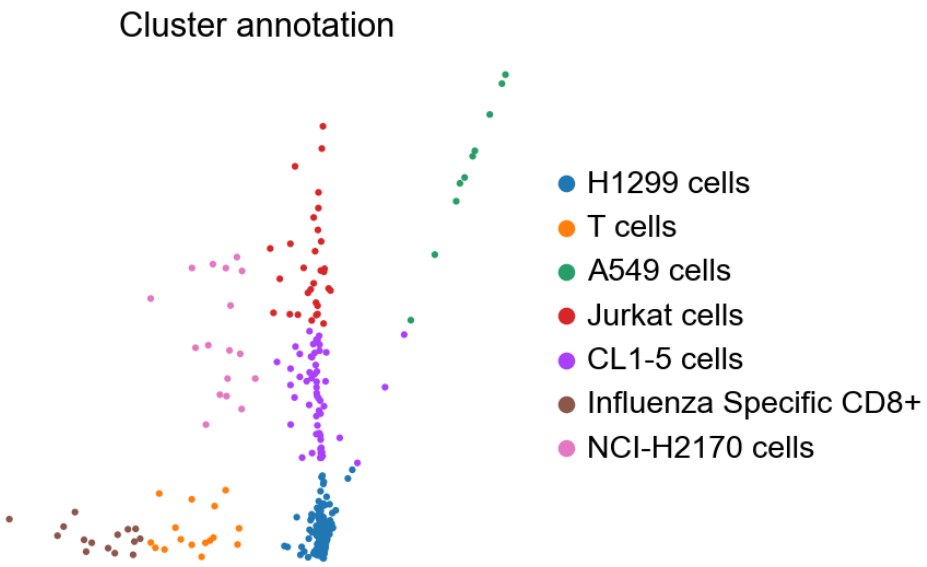


Figure 2.13: Cluster annotation for H1299_scRNAseq.

raro datasets, respectively, in Figures 2.8 and 2.9. On the other hand, the results of Isomap and Standard LLE show slightly better performance comparatively. Moreover, Laplacian Eigenmaps performed better than these two methods, though they could not accomplish competitive clustering. As a good example, we visualize samples from H1299_scRNAseq using Laplacian Eigenmaps (Fig.2.10) in which different clusters are overlapping. Finally, we investigated MLLE and found the most insightful cluster separation in most of the datasets. This outcome demonstrates the power of MLLE in exploring the data's dense and complex relations, creating better lower embeddings. We performed an additional dimensionality reduction step that uses ICA to enhance the visualization of clusters. The last column of Table 6.1 represent that MLLE combined with ICA improves the overall results except for some datasets that we can not see much difference; very negligible difference of 0.004 (Baron_human1), 0.001 (Baron_human2), 0.014 (Baron_human3), 0.004 (Segerstolpe), and 0.011 (Xin) can ignore them. To achieve a better view of the impact of

ICA on the MLLE transformation, we show a visual comparison of clusters in Figures 2.11 and 2.12. Two-dimensional ICA projection of the cells applied to the three-dimensional MLLE data shows the best visualization and clustering scores (Fig. 2.12). When applied alone, ICA performed very poorly with significantly inseparable clusters (Fig. 2.11). This is because ICA is limited to linear transformations.

On the other hand, manifold learning techniques consider data locally. As such, it can reveal complex relationships among the data points in higher-dimensional spaces. We instead applied ICA on the lower-dimensional data because we observed well-marked ”lines” or ”axes” in the three-dimensional data, which led us to think that we could apply ICA to learn the linearly independent components, not necessarily orthogonal. Applying ICA reveals some hidden, complex relationships among the cells in the clusters, which are not noticeable in three dimensions.

### 2.3.2 Biological Assessment

To validate the obtained clusters, we first identified the top 20 genes in each cluster based on the Wilcoxon test. Starting from these top 20 genes, we retrieved a subset of genes from the largest number of overlapping genes across the different clusters. Marker genes are up- or down-regulated in different individual cells, pathways or GO terms. We used GSEA and ToppCluster multi-gene list functional enrichment analysis online tools to identify GO terms and pathways associated with the top 20 gene lists extracted from each cluster. Pathways were extracted from the MSigDB C2 BIOCARTA (V7.3) database [63]. Cytoscape [97] was used to visualize the networks.We decreased the minimum number of genes present in annotations to achieve a better visualization.

As presented in Table 2.6, some of the pancreatic cell types are found for pancreas

Table 2.4: Identified cell types for H1299_scRNAseq.

| **Cell Types** | **Cluster Number** |
|---|---|
| H1299 cells | 0 |
| T cells | 1 |
| A549 cells | 2 |
| Jurkat cells | 3 |
| CL1-5 cells | 4 |
| Influenza-specific CD8+ | 5 |
| NCI-H2170 cells | 6 |

datasets such as the Baron human dataset within well-defined gene sets in MSigDB namely 'MURARO PANCREAS ALPHA CELL', 'MURARO PANCREAS ENDOTHELIAL CELL', 'MURARO PANCREAS MESENCHYMAL STROMAL CELL', 'MURARO PANCREAS DUCTAL CELL', and 'MURARO PANCREAS ACINAR CELL'. Other cell types including CD34, Jurkat, and macrophage are cell subtypes of T-Cells. HB2 is also a cell line originated by epithelial cells. Regarding H1299_scRNAseq and Calu3_scRNAseq datasets, Tables 2.4 and 2.5 list associated cell types mostly involved in the immune system. It is well-known that one of the main SARS-CoV-2 targets is the immune system function. We observed co-expressed gene sets down- or up-regulated in the lung and immune systems specific cell (sub)types. T-cell is a type of immune cell that is found in blood. Jurkat cells are a line of human T cells that are used to study the expression of various chemokine receptors susceptible to viral entry, particularly HIV. CD8+ T cells are found on the surface of immune cells and are key cells in response to viral infection [21]. Moreover, H1299 cells, NCI-H2170 cells, A549 cells, and CL1-5 cells are human lung associated cell lines. These findings show the effectiveness of the proposed method to identify associated cell types using cell type specific marker genes. A projection of the identified cells in H1299_scRNAseq colored by clusters is shown in Fig. 2.13.

Additionally, visualization of GO terms and pathways associated with the correspond-

Table 2.5: Identified cell types for Calu3_scRNAseq.

| Cell Types | Cluster |
|---|---|
| H1299 cells | 0 |
| 293 cells (embryonic kidney) | 1 |
| MCF7 cells | 2 |
| ANBL-6 cell | 3 |
| T-ALL | 4 |
| H460 cells | 5 |
| H1975 cells | 6 |

Table 2.6: Identified cell types for Baron_human1 dataset.

| Cell Types | Cluster |
|---|---|
| Alpha | 0 |
| CD34 | 1 |
| Mesenchyme stem cells | 2 |
| Jurkat cells (T lymphocyte) | 3 |
| Endothelial | 4 |
| Mesenchyme stromal cells | 5 |
| Ductal | 6 |
| Endothelial | 7 |
| Acinar | 8 |
| Myeloid cells | 9 |
| Intestine cells | 10 |
| Macrophage | 11 |
| HB2 cells | 12 |
| T-cells | 13 |

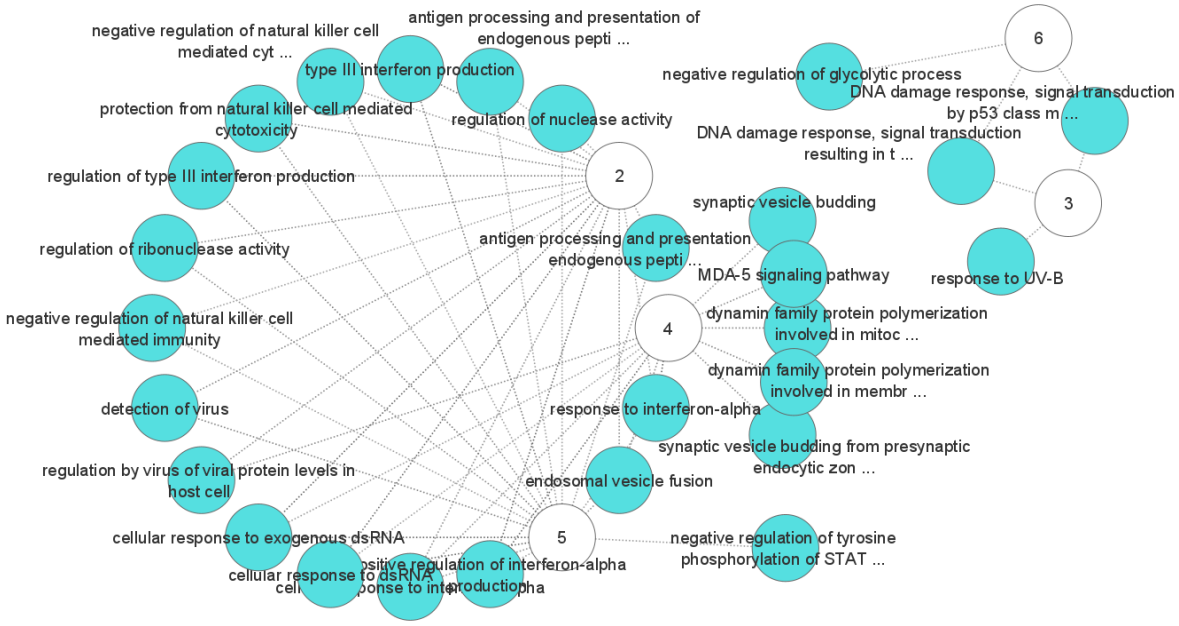


Figure 2.14: A set of biological process that are enriched by marker genes in H1299_scRNAseq dataset. The numbers show the clusters and edges shows the link between a cluster and a biological process term.

ing marker genes are depicted in Figs. 2.14 and 2.15, respectively. For each cluster, we identified a set of biological process or pathway terms. Each edge in the plot shows a link between a cluster and a term that is significantly associated with the 20 top gene list in that cluster.By observing Fig. 2.15, some significant pathways are found to be enriched in immunity functions, and signaling identified, including SARS-CoV-2 innate Immunity Evasion, Host-pathogen interaction of human corona viruses, SARS coronavirus and innate immunity, Type II interferon signaling (IFNG), and the human immune response to tuberculosis. Also, the gene set enrichment of Fig. 2.14 shows that most biological processes are associated with immunity functions, including response to interferon-alpha, protection from a natural killer cell, type III interferon production, regulation by virus of viral protein levels in a host cell, and detection of virus, among others. In addition, we obtained a list

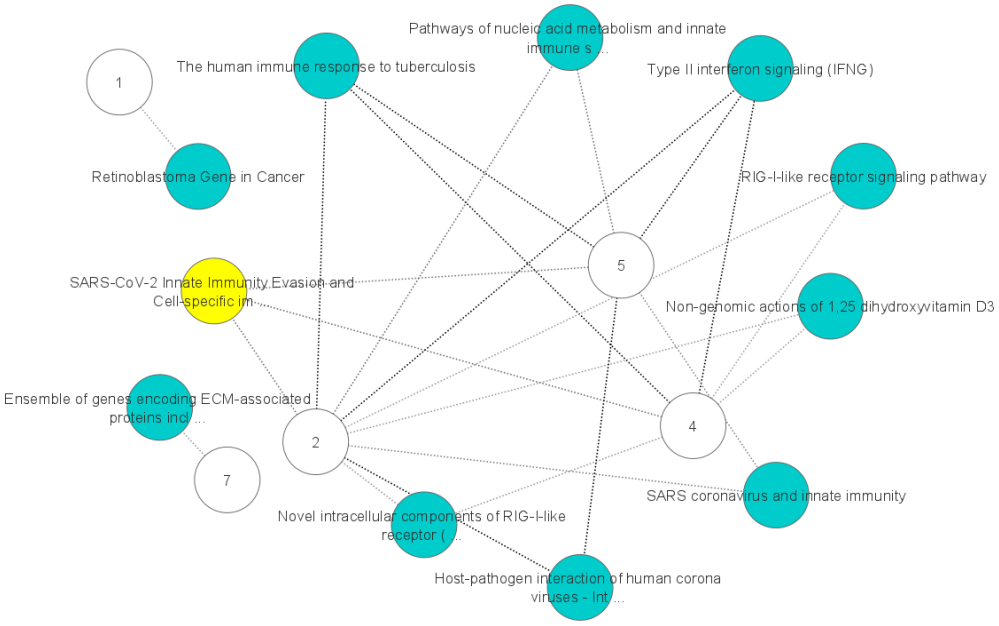


Figure 2.15: Pathway that are enriched by marker genes in H1299_scRNAseq dataset. The numbers show the clusters and edges shows the link between a cluster and a pathway. Node that is highlighted yellow show the SARS-CoV-2 cell-specific pathway. Most of the other green nodes reveal the shared and cluster-specific functional pathways in the immune system.

of overlapping marker genes that are involved in Herpes simplex virus 1 (HSV-1) infection and the Influenza A pathway. These findings suggest potential markers for subsequent medical treatment or drug discovery by comparing to similar diseases in terms of functionality. Moreover, although numerous findings suggest potential links between HSV-1 and Alzheimer's disease, a causal relationship has not been demonstrated yet [26].

## 2.4 Conclusion and Future Work

This work focuses on the identification of different cell types using manifold learning combined with clustering techniques on scRNA-seq data. Identifying similarities that re-

sult from structural, functional, or evolutionary relationships among the genes is the primary goal of clustering the cells. Our proposed two-step representation learning approach demonstrated that *k*-means clustering technique combined with Modified LLE leads to improved clustering output and meaningful organization of cell clusters by ”untangling” the complex, hidden relationship in a higher-dimensional space.

Non-linear dimensionality reduction methods have been shown to be very powerful as they preserve the locality of the data from higher to lower dimensions. UMAP is one of the most commonly-used non-linear dimensionality reduction technique, and has been shown to perform well on large-scale scRNA-seq data. However, for dimensionality reduction, UMAP is not as efficient as MLLE on high-dimensional cytometry, especially when combined with clustering to enhancing the visualization of the clustering results. This behavior of MLLE has been observed in our experiments. A comparative analysis with UMAP in the Supplementary Material, Figure S4, confirms this observation.

Moreover, performing ICA on transformed data after applying manifold learning techniques provides enhanced view of the data in a reduced space. Evaluating the incidence of ICA as a visualization scheme and further reduction step, after applying MLLE, shows better clustering and enhanced visualization simultaneously. This trend leads to a research avenue that involves a combination of non-linear manifold learning techniques followed by linear methods, which has shown to be more powerful than conventional methods such as PCA or ICA applied alone.

Using multiple benchmark datasets shows the effectiveness of our proposed method. Performing gene set enrichment analysis to annotate a set of HVGs obtained from each cluster reveals biomarker genes involved in different gene ontology terms.

There are some other potential applications for investigating scRNA-seq data, even beyond cell type identification. Using an extension of the proposed method by employing other manifold or deep learning techniques on the other epigenetic challenges in scRNA-seq data analysis, such as trajectory analysis, is our next step.

# Chapter 3

# Cell Type Annotation Model Selection: General-Purpose vs. Pattern-Aware Feature Gene Selection in Single-Cell RNA-Seq Data

## 3.1 Introduction

In living organisms, there are a great variety of cells that can be distinguished with the help of single-cell RNA sequencing (sc-RNA sequencing) technology. Single-cell RNA sequencing (scRNA-seq) is a novel sequencing technology that involves individual cell information and can be used in cell heterogeneity studies. Studying different types of cancer, detecting unknown tumours and tumour heterogeneity, drug discovery, diagnosis, and prognosis are a few numbers of the new opportunities for research in this scope.

Identifying cell type heterogeneity is one of the first fundamental steps in an in-depth analysis of single-cell RNA sequencing data. Hidden diversity and characteristics of a particular cell type can be found via deferentially expressed genes (DEGs). Machine learning approaches can be effectively used to identify hidden differentiation in the expression profiles of the genes with high probability. scRNA-seq data comes with a variety of limitations. The highlighted one is the lack of annotation for most of the data which are publicly available. In a general single-cell RNA-seq downstream analysis, clustering techniques are widely used to reveal groups of cells and cell types. However, setting up the parameters, including the number of clusters, is a challenging point [108]. For instance, several methods are compared in [31]. Among them, SC3 [56], CIDR [65], Ascend [96], SAFE-clustering [123], TSCAN [51], and [107] all possess built-in methods for estimating the optimal number of clusters. Although Ascend and CIDR underestimated the number of clusters, SC3 and TSCAN tend to overestimate. In addition, the group of cells identified by the clustering methods requires an additional annotation step with the corresponding cell types using canonical marker genes and reference databases. Hence, the conventional workflow based on clustering and marker genes is not scalable due to manual annotation. The lack of ground-truth information and tool benchmarking makes it more complex to evaluate the model. Therefore, manually annotating the output is a time-consuming and non-reproducible procedure in clustering methods. The other limitation of scRNA-seq data is caused by biological effects during sequencing. This leads to a zero-inflated read counts matrix with thousands of zeros in expression values, which may mislead downstream analyses. Cell types are often distinguished by calculating the differentiation of expression levels of only the most informative genes. Hence, finding known marker genes among thousands of genes with almost zero information is essential in scRNA-seq data analyses.

This is either called dimensionality reduction or feature selection and affects the final result directly. Although the current unsupervised methods show superiority in the performance when combined with feature selection methods, the biological significance of the results is still important for the understanding of the underlying biological information and requires further manual gene set enrichment analysis [15, 108]. Since feature selection plays a significant role in domain-dependent problems, a wide range of supervised techniques shows superiority in the performance utilizing feature selection methods. Supervised techniques have increasingly developed for the automatic identification and annotation of cell types. Moreover, using annotated data, we can evaluate and compare the model by systematically estimating the performance metrics. A comparative study in [1] reviewed 22 supervised techniques, including random forest (RF), which is based on decision tree rules. To assess the probability of a correct label, decision trees inherently select informative features and estimate the minimum number of features needed to create a model. Among the other choices in decision tree categories, XGBoost shows its capability in all scenarios [22]. According to this, CaSTLe [64] was proposed based on an XGBoost model under transfer learning workflow and showed satisfactory classification accuracy compared to two linear models. The idea behind CaSTLE is to use a robust univariate feature engineering workflow followed by the application of a pre-tunedXGBoost model. In the feature selection workflow first, genes with the top mean level of expressions and mutual information were selected, and correlated genes were removed; then, considering pre-defined ranges, genes were categorized. Transfer learning uses information from one scRNA-seq dataset to annotate another one.

Additionally, the ensemble learning schema combines weak learners' voting for an accurate final vote on similarity search space. For instance, EDGE [100] has utilized this

approach on simulated data and learned an ensemble version of similarity matrices into a single embedding space of data, as well as optimizing through stochastic gradient descent. In EDGE, dimensionality reduction and feature gene extraction were used in an ensemble approach in such a way that the problem of finding similarity among features, was broken down into small weak learners. The final similarity matrix achieved shows a common similarity space among all learners. SMaSH [79], on the other hand, is designed explicitly for gene ranking and calculating the significance score of marker genes from scRNA-seq data. Focuses on marker genes ranking, SMaSH compares tree-based and neural network-based approaches. It uses predefined cell types (labels) to categorize cell-specific genes before feeding them into several weak classifiers. In a benchmarking experiment, SMaSH compared the ensemble mode with the network mode. Its performance was evaluated using tree-based models, including XGBoost. Compared to the other two ensemble models and deep neural networks, XGBoost shows excellent performance in most scenarios. Although according to the observations in [72], XGboost failed to detect small changes in expression levels and consequently distinguish cell sub-types. It is a much faster and simpler approach compared to the neural network model. XGBoost is well-suited to large datasets by performing in parallel. Moreover, in our recent comparative study, it has been shown that the support vector machine (SVM), with the help of information gain (IG), as a feature selection method, outperformed the other approaches [109]. The study was performed on nine different experiments composed of three different state-of-the-art popular classifiers combined with three general-purpose feature selection methods. Classifiers, including random forest, $K$-NN, and SVM and feature selection methods, including Analysis of Variance (ANOVA) F-value, Information Gain, and Chi-squared considered complementary of the classifiers. One of the challenges covered in this study was selecting cell-specific genes

in the feature selection step. A benchmark study was performed based on the number of selected features. However, it remains for a exploration of general-purpose feature engineering techniques against domain-specific ones, and in particular pattern-aware techniques to be done. When reviewing different approaches, including supervised and unsupervised, for cell type annotation based on scRNA-seq data, there is no comparison with the XGBoost method. Precisely, the power of XGBoost and SVM was proven in the previous studies and XGBoost performs in a faster and simpler way. Moreover, SVM together with a general-purpose feature selection had been shown as a high-performance method in the supervised cell-type annotation. In this study, inspired by the recent works completed in cell-type classification, we compared two forefront approaches; the general-purpose model, a SVM classifier with information gain feature selection method and XGBoost tree with its inherent feature selection strategy. This paper guides users and practitioners to select the most proper model based on the inherent features of their datasets.

## 3.2 Materials and Methods

### 3.2.1 Framework

The schematic view of Figure 3.1 depicts the pipeline in a cell-type annotation process. First, the raw read count matrix is generated using high-throughput sequencing technologies (Figure 3.1, step 1). These raw data includes expression profiles of thousands of cells separately (Figure 3.1, step 2). Performing pre-processing, including filtering, normalization, and scaling, gives us ready-to-use data for the computational step (Figure 3.1, step 3). Then, the most informative features are extracted in the feature selection (Figure 3.1, step 4) to be used by classification models. Finally, cell types are predicted and annotated by

the method with higher accuracy (Figure 3.1, step 5). As a demonstration of the high performance, a gene set enrichment analysis on the selected features was performed, and the results highlight the power of the model in annotating cell types (Figure 3.1, step 6). The last step is not necessary for supervised approaches. However, it could play a verification phase in the biological context.

To validate our models, we considered the most commonly used evaluation metrics, namely accuracy, precision, and recall, to systematically estimate and compare the performance of our models. To this end, we used 10-fold cross-validation to test and train the model. Additionally, we tuned XGBoost parameters as follows: (1) the regularization parameter value to create a new split in trees, *gamma* is set to 0.2, 0.1, and 0 for Data1 to Data3, respectively. (2) ($Max-depth$ and $min-child-weight$) of the tree, which typically control overfitting, were fine-tuned to (10, 3), (5, 3), and (10, 1) for Data1 to Data3, respectively.

(3) $colsample-bytree$, which determines what portion of features will be used, was set to 0.5, 0.4, and 0.3 for Data1 to Data3, respectively.

We used Scikit-learn [83] in Python version 3.7 to perform computational algorithms, and GSEA [99] for biological validation.

### 3.2.2 Dataset

To evaluate the performance of the model, we used public, annotated scRNA-seq datasets with accession numbers GSM2230757, GSM2230758, and GSM2230762 under series GSE84133 [10] extracted from NCBI's Gene Expression Omnibus [32]. These datasets include transcripts of pancreatic from human and mouse donors. Pancreatic cells are divided into 14 groups of previously characterized cell types, mainly including alpha, beta, acinar, delta,

quiescent, activated pancreatic stellate, endothelial, and ductal cells. The existence of these cell types is validated with immuno-histochemistry stains [10] so that it can be a good resource for discovering cell types. The details of the datasets used in this study are listed in Table 3.1.

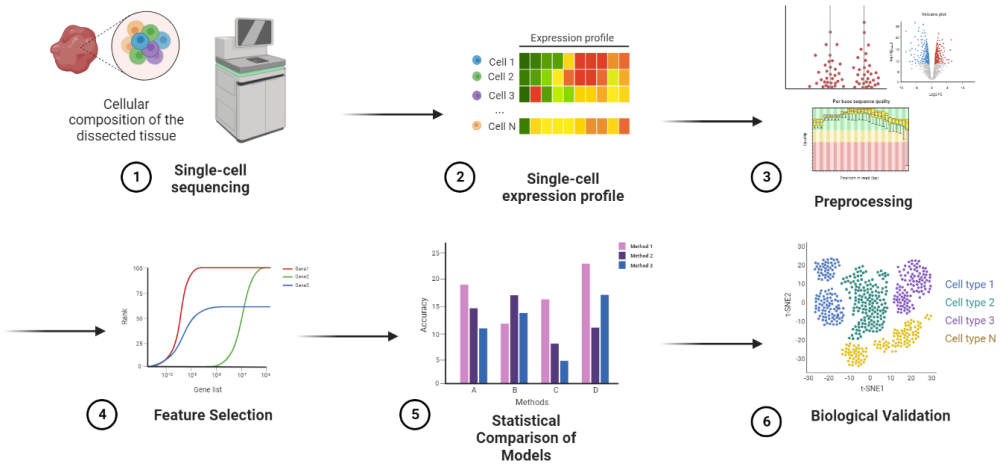


Figure 3.1: Pipeline overview of the experiments.

Table 3.1: Details of the datasets analyzed in this study.

| Dataset | Accession # | # Cell Types | # Cells | # Genes |
|---|---|---|---|---|
| Human Pancreatic Islets, Sample 1 (Data1) | GSM2230757 | 8 | 1937 | 20,125 |
| Human Pancreatic Islets, Sample 2 (Data2) | GSM2230758 | 8 | 1724 | 20,125 |
| Mouse Pancreatic Islets, Sample 2 (Data3) | GSM2230762 | 8 | 1064 | 14,878 |

### 3.2.3 Data Pre-Processing

Raw read count matrices contain low-quality RNA sequencing information based on differential expression levels. Data pre-processing is performed to ensure removing any weakly

expressed genes or low-quality cells, including damaged, dead, or degraded during sequencing, and are represented by a low number of expressed genes in the read count matrices. We followed the standard pre-processing pipeline in scRNA-seq data analysis [68]. Based on this pipeline, cells with less than 200 expressed genes and genes expressed in less than three cells are filtered out. In Data1, for example, we first filtered out 5387 low-expression genes that were detected in less than three cells and kept 14,739 genes. Further analysis of the data distribution showed low-quality cells and led to removing seven cells. After per-gene quantification, we selected a subset of highly variable genes to use in downstream analyses. To this end, we defined the set of highly variable genes given a normalized dispersion higher than 0.5 after normalization and obtained 2546 genes at the end. We used Scanpy [118], a specifically designed package to work with scRNA-seq datasets, for pre-processing steps.

### 3.2.4 Hyperparameter Tuning

Hyperparameter tuning, also known as hyperparameter optimization or model selection, is the process of systematically searching for the best combination of hyperparameters to optimize the performance of a machine learning model. Hyperparameters are parameters that are not learned from data but are set before training. Examples of hyperparameters for XGBoost include the learning rate, max depth, gamma, minimum child weight, and column sample by the tree. In this research, the process of tuning these parameters has been completed automatically using Bayesian optimization. Bayesian optimization is a method for efficiently searching for the best set of hyperparameters of a model. The basic idea is to use a probabilistic model, such as a Gaussian process, to model the function that maps from hyperparameters to the performance of the model on a given task. Mathematically,

Bayesian optimization can be formulated as an optimization problem in which we want to find a set of hyperparameters that maximize the expected performance of the model, given the current state of the probabilistic model. The expected performance is given by the mean of the model, and the uncertainty in the model is represented by the variance. The acquisition function, such as the expected improvement, is used to balance the exploration and exploitation of the search space. To optimize the acquisition function, we optimize the hyperparameters of the probabilistic model to maximize the expected improvement. This is done by using gradient-based optimization algorithms, such as L-BFGS, or using more sophisticated methods, such as Hamiltonian Monte Carlo. In summary, Bayesian Optimization is a powerful method for tuning a machine learning model's hyperparameters by using a probabilistic model to guide the search for the best set of hyperparameters and balancing exploration and exploitation using an acquisition function.

### 3.2.5 Feature Selection

Feature selection is a non-separable part of any algorithms that work with large-scale data due to the curse of dimensionality. The existing thousand genes expressed in each individual cell in the scRNA-seq dataset make it high-dimensional, which required a reduction in the number of genes. The idea behind gene selection in cell type identification is motivated by the fact that cell types are often distinguished by only a few essential genes known as biomarkers. The effectiveness of three general-purpose feature selection methods was explored in cell-type classification problems in [109], including Analysis of Variance (ANOVA) F-value, Chi-squared, and information gain. The findings show that information gain yields the best biomarkers among all other models. Information gain is defined based on impurity and entropy. The group with the higher information gain possesses less uncer-

tainty. The importance of a feature is estimated by considering the information gained from each feature. It is defined as the difference between before and after considering feature $X$ in the classification process, as shown in Equation (7.2) [88].

$$IG(X) = \sum_i U(P(C_i)) - E\left[\sum_i U(P(C_i|X))\right] \tag{3.1}$$

where $IG(X)$ represents the information gain from feature $X$. $U$ represents uncertainty function, $P(C_i)$ represents the probability of class $C_i$ before considering feature $X$, and $P(C_i|X)$ represents the posterior probability of class $C_i$ after considering feature $X$.

On the other hand, the feature selection algorithm in the XGBoost considers sparsity in the data and defines a default direction for missing values. Hence, it simplifies the classification process by utilizing inherent sparsity patterns in the data. Therefore, it divides data into two supergroup samples with missing and present values. XGBoost exploits the sparsity to make the computational complexity linear proportional to the number of existing values in the input matrix [22].

### 3.2.6 XGBoost

Extreme Gradient Boosting, XGBoost, is a scalable and widely-used decision tree gradient-boosted algorithm that offers state-of-the-art results on many machine learning problems. It provides a statistical model that captures the dependency of large datasets considering the sparsity of the data and has been shown in a wide range of standard classification applications [59]. XGBoost is reported as the top first-ranked method among the most popular ones outperforming the other popular solutions. The second-ranked method, deep neural nets, also obtains better results when combined with XGBoost [22]. Similar to the random

forest, a Gradient boosting decision tree follows an ensemble learning algorithm and is under a gradient tree-boosting framework. Ensemble learning algorithms combine multiple models to obtain an average of all models.

The idea follows from the existing Gradient boosting algorithms with minor improvements in the regularized objective. Unlike decision trees, regression trees include a continuous weight on each leaf. For a given data, the regression tree uses the decision rules to classify it into different groups in the leaves. It calculates the overall prediction score by summing up the score weights in the leaves. The regularization objective function has to be minimized as follows:

$$\iota(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{3.2}$$

$$where \quad \Omega(f) = \gamma T + \frac{1}{2}\lambda\omega^2$$

Here, $\iota$ is the loss function to calculate the difference between the predicted and actual class, $\hat{y}_i$ and $y_i$, respectively. The term $\Omega$ controls the complexity of the model, i.e., the regression tree functions. The term $\gamma$ helps avoid over-fitting utilizing the final weights. The regularized greedy forest (RGF) model [53] uses a similar regularization method, but it is more complex. Parallelization is another positive point of XGBoost.

## 3.3 Results and Discussion

The first objective of this work was to evaluate the accuracy of the main classifiers (i.e., SVM, *k*NN, and RF) with a group of genes extracted from a pioneer general-purpose feature selection method, information gain. The second scenario was defined with genes obtained using the inherent approach in the XGBoost tree, which uses the latent pattern in

scRNA-seq data. We calculated the average of all measurements when comparing the results. The number of features is determined based on the one with the highest final predictive accuracy. Results are presented in Tables 3.6, 3.2 and 3.3. Overall, our findings indicate that XGboost obtained the first rank among other methods in terms of accuracy and recall. On the other hand, when looking at precision, SVM with information gain feature selection is a top-ranked method. These results highlight three facts: (1) XGboost is the best model when it comes to finding cell types in general (higher average accuracy). Since accuracy represents the overall correctness of the model and precision shows how good a model is at predicting a specific cell type, it is more probable to fail in finding specific cell types (less precision). More precisely, finding rare cell types with a few number of cell-specific genes is more effective in exploiting SVM and information gain.

(2) Our observations confirm that XGBoost is faster and more scalable in the case of large-scale datasets, mainly because it uses its inherent feature selection simultaneously with the classification and optimization phase. (3) Compared to a tree-based model without an ensemble approach, i.e. random forest, XGBoost highlights the power of boosting strategy, either in the classification phase or feature selection phase of cell-type annotation.

In addition, an extra validation step was performed to confirm the achievements in the training phase in a more biologically meaningful scheme. The following subsections describe more details of our findings.

### 3.3.1 Classification Results

To explore the effect of the selected feature genes as a form of prior knowledge, we evaluated the classifiers' performance based on the different numbers of selected features. The

Table 3.2: Comparison of classification results for Data2.

| **Method** | **# Features** | **Accuracy %** | **Precision %** | **Recall %** |
|---|---|---|---|---|
| SVM + IG | 400 | 98.09 | 79.04 | 98.08 |
| RF + IG | 400 | 96.06 | 65.86 | 94.76 |
| *k*NN + IG | 200 | 94.66 | 65.68 | 96.63 |
| XGBoost + IG | 400 | 99.67 | 75.63 | 96.62 |
| XGBoost | 400 | 99.78 | 76.83 | 99.22 |

optimal value of features, $k$, where $k = 100, 200, 300$, and 400 was determined by exploiting a greedy approach. Observing the results of the classification methods for Data1 shown in Table 3.6, all models reveal less misclassification rate with 400 features. In particular, SVM combined with IG gives an accuracy of 98.08%, and Precision and Recall of 87.98% and 96.76%, respectively. Additionally, $k$-NN presents a high accuracy of 96.11% when using the IG feature selection method. Moreover, random forest combined with IG delivers a high accuracy of 97.05%. XGBoost with and without IG obtains an average accuracy of 99.51% and 99.63%, respectively, which outperformed other methods. However, regarding precision, XGBoost, with only its inherent feature splitting algorithm, is the best one in the list and shows lower precision compared to the SVM with its external general-purpose feature selection method. These two methods achieved recall values with a low difference of close to zero. Additionally, the results of Data2 show almost an equivalent accuracy for SVM and XGboost methods (Table 3.2). $k$-NN classification method achieves high accuracy (94.66%) with 200 features selected from IG, RF, and IG combined, achieving high accuracy (96.06%) with 400 features. Lastly, SVM achieves high accuracy with 400 features (98.09%) selected from the IG feature selection method. XGBoost coupled with IG provides the best performance, with 99.67% accuracy.

For Data3, SVM outperformed the other two classification methods and achieved the highest precision of 84.91% with 300 features selected from the IG feature selection method. Regarding misclassification rate and recall, the results are very close to XGBoost. In general, Data3 has fewer features, comparatively speaking. Hence, as mentioned earlier, XG-

Table 3.3: Comparison of classification results for Data3.

| Method | # Features | Accuracy % | Precision % | Recall % |
|---|---|---|---|---|
| SVM + IG | 300 | 99.23 | 91.95 | 93.29 |
| RF + IG | 400 | 99.03 | 67.26 | 88.68 |
| *k*NN + IG | 400 | 98.60 | 73.63 | 88.01 |
| XGBoost + IG | 400 | 99.42 | 80.72 | 90.58 |
| XGBoost | 400 | 99.18 | 78.91 | 93.17 |

Boost is less effective when it comes to small-scale datasets.

### 3.3.2 Biological Validation

We performed an extra step of biological evaluation for detecting cell types using highly-ranked features identified in the feature selection phase. Among a wide range of gene set enrichment analysis (GSEA) databases, we chose the C8 collection of MSigDB, which includes cell type signature's gene sets [99]. We separated each class's top 20 differentially expressed genes for enrichment analysis. Table 3.7 shows the list of six pancreatic cell type-specific gene sets identified by the list of marker genes extracted from the feature selection phase on Data1. Additionally, as shown in Table 3.8, a maximum of 9 out of 20 overlapped genes between our top 20 ranked genes and pancreas gene sets were highlighted in the list. The enrichment analysis results of two other datasets are shown in Tables 3.4 and 3.5.

## 3.4 Conclusions

This study compares two recently reported pioneer classification models, XGBoost and SVM, for discovering cell types using a list of marker genes. One with a blind feature selection method, pure information gain, and the other one with data sparsity-aware inherent feature selection, GXBoost feature splitting algorithm. It is shown that considering

Table 3.4: List of 17 out of 20 overlapped genes between our top 20 ranked genes and pancreas gene sets (Data2).

| **Gene Symbol** | **Description of Functionality** |
|---|---|
| PMEPA1 | prostate transmembrane protein, androgen induced 1 [Source:HGNC Symbol;Acc:HGNC:14107] |
| TACSTD2 | tumor associated calcium signal transducer 2 [Source:HGNC Symbol;Acc:HGNC:11530] |
| KRT7 | keratin 7 [Source:HGNC Symbol;Acc:HGNC:6445] |
| SDC4 | syndecan 4 [Source:HGNC Symbol;Acc:HGNC:10661] |
| SOX4 | SRY-box transcription factor 4 [Source:HGNC Symbol;Acc:HGNC:11200] |
| KRT19 | keratin 19 [Source:HGNC Symbol;Acc:HGNC:6436] |
| FLNA | filamin A [Source:HGNC Symbol;Acc:HGNC:3754] |
| FXYD3 | FXYD domain containing ion transport regulator 3 [Source:HGNC Symbol;Acc:HGNC:4027] |
| IFITM3 | interferon induced transmembrane protein 3 [Source:HGNC Symbol;Acc:HGNC:5414] |
| SERPING1 | serpin family G member 1 [Source:HGNC Symbol;Acc:HGNC:1228] |
| COL18A1 | collagen type XVIII alpha 1 chain [Source:HGNC Symbol;Acc:HGNC:2195] |
| PCSK1 | proprotein convertase subtilisin/kexin type 1 [Source:HGNC Symbol;Acc:HGNC:8743] |
| HADH | hydroxyacyl-CoA dehydrogenase [Source:HGNC Symbol;Acc:HGNC:4799] |
| MAFA | MAF bZIP transcription factor A [Source:HGNC Symbol;Acc:HGNC:23145] |
| ARX | aristaless related homeobox [Source:HGNC Symbol;Acc:HGNC:18060] |
| IRX2 | iroquois homeobox 2 [Source:HGNC Symbol;Acc:HGNC:14359] |
| GC | GC vitamin D binding protein [Source:HGNC Symbol;Acc:HGNC:4187] |

Table 3.5: List of 15 out of 20 overlapped genes between our top 20 ranked genes and pancreas gene sets (Data3).

| **Gene Symbol** | **Description of Functionality** |
|---|---|
| SPARC | secreted protein acidic and cysteine rich [Source:HGNC Symbol;Acc:HGNC:11219] |
| COL4A1 | collagen type IV alpha 1 chain [Source:HGNC Symbol;Acc:HGNC:2202] |
| FLT1 | fms related receptor tyrosine kinase 1 [Source:HGNC Symbol;Acc:HGNC:3763] |
| PECAM1 | platelet and endothelial cell adhesion molecule 1 [Source:HGNC Symbol;Acc:HGNC:8823] |
| SERPINH1 | serpin family H member 1 [Source:HGNC Symbol;Acc:HGNC:1546] |
| COL4A2 | collagen type IV alpha 2 chain [Source:HGNC Symbol;Acc:HGNC:2203] |
| IGFBP7 | insulin like growth factor binding protein 7 [Source:HGNC Symbol;Acc:HGNC:5476] |
| CDH5 | cadherin 5 [Source:HGNC Symbol;Acc:HGNC:1764] |
| VIM | vimentin [Source:HGNC Symbol;Acc:HGNC:12692] |
| PMEPA1 | prostate transmembrane protein, androgen induced 1 [Source:HGNC Symbol;Acc:HGNC:14107] |
| MSN | moesin [Source:HGNC Symbol;Acc:HGNC:7373] |
| S100A16 | S100 calcium binding protein A16 [Source:HGNC Symbol;Acc:HGNC:20441] |
| ANXA2 | annexin A2 [Source:HGNC Symbol;Acc:HGNC:537] |
| CD24 | CD24 molecule [Source:HGNC Symbol;Acc:HGNC:1645] |
| NFIB | nuclear factor I B [Source:HGNC Symbol;Acc:HGNC:7785] |

Table 3.6: Comparison of classification results for Data1.

| **Method** | **# Features** | **Accuracy %** | **Precision %** | **Recall %** |
|---|---|---|---|---|
| SVM + IG | 400 | 98.08 | 87.98 | 96.76 |
| RF + IG | 400 | 97.05 | 77.48 | 96.52 |
| *k*NN + IG | 400 | 96.11 | 77.53 | 96.51 |
| XGBoost + IG | 400 | 99.51 | 80.45 | 91.68 |
| XGBoost | 400 | 99.63 | 88.41 | 96.38 |

Table 3.7: List of eight gene sets correlated to the Pancreatic cell types of Data1 resulting from the GSEA analysis.

| **Pancreas Gene Set Name** | **Dataset** |
|---|---|
| Muraro pancreas endothelial cell [362] | Data1, Data2, Data3 |
| Muraro pancreas mesenchymal stromal ce cell [681] | Data1, Data2, Data3 |
| Muraro pancreas acinar cell [732] | Data1, Data2, Data3 |
| Muraro pancreas ductal cell [1276] | Data1, Data2, Data3 |
| Muraro pancreas alpha cell [568] | Data1, Data2 |
| Descartes fetal pancreas islet endocricrine cells [170] | Data1, Data2 |
| Muraro Pancreas Epsilon Cell [44] | Data2 |
| Muraro Pancreas Delta Cell [250] | Data2 |

Table 3.8: List of 9 out of 20 overlapped genes between our top 20 ranked genes and pancreas gene sets (Data1).

| **Gene Symbol** | **Description of Functionality** |
|---|---|
| IFITM3 | interferon induced transmembrane protein 3 [Source:HGNC Symbol;Acc:HGNC:5414] |
| IGFBP4 | insulin like growth factor binding protein 4 [Source:HGNC Symbol;Acc:HGNC:5473] |
| IFITM2 | interferon induced transmembrane protein 2 [Source:HGNC Symbol;Acc:HGNC:5413] |
| COL4A1 | collagen type IV alpha 1 chain [Source:HGNC Symbol;Acc:HGNC:2202] |
| SPARC | secreted protein acidic and cysteine rich [Source:HGNC Symbol;Acc:HGNC:11219] |
| IGFBP7 | insulin like growth factor binding protein 7 [Source:HGNC Symbol;Acc:HGNC:5476] |
| VIM | vimentin [Source:HGNC Symbol;Acc:HGNC:12692] |
| TM4SF1 | transmembrane 4 L six family member 1 [Source:HGNC Symbol;Acc:HGNC:11853] |
| HLA-B | ”major histocompatibility complex, class I, B [Source:HGNC Symbol;Acc:HGNC:4932]” |

the data with its latent sparsity pattern significantly enhances the overall accuracy of the predictive models. Since the high degree of sparsity in scRNA-seq data arises from false technical zeros and true biological zeros, exploiting the patterns of existing and non-existing values for selecting biomarkers makes it more precise, faster, and more meaningful. Our study particularly demonstrates the effectiveness of ensemble tree models with an inherent sparsity-awareness feature selection approach in the cell-type automatic annotation problem. Biological validation of the results confirmed the overall accuracy of the prediction.

Moreover, the lack of canonical biomarkers for certain cell types makes it more complicated to find rare cell types using the existing genes in the list of top-ranked ones. In this case, following a manual lookup in the gene set repositories of related genes in gene sets could support the study and the results. Biologically speaking, the relation among genes is defined by structural, functional or evolutionary information. This work provides a guideline for researchers to select and apply the well-suited tool in annotating cell types using associated genes or uncovering homogeneous markers.

# Chapter 4

# Cell type identification via convolutional neural networks and self-organizing maps on single-cell RNA-seq data

## 4.1 Introduction

Single-cell RNA-seq (scRNA-seq) profiles the unique gene expression of individual cells. Differentially expressed genes lead to cell heterogeneity in different tissues, and as such, tumour heterogeneity is a common phenomenon [30]. In this regard, scRNA-seq can be helpful in detecting unknown tumours, and consequently, improving therapies and drug discovery. One of the main steps to perform an in-depth analysis of scRNA-seq data is cell-type classification and identification. Employing supervised learning techniques on scRNA-seq data is an effective approach to this end. First, by incorporating cell types in annotated datasets, manual annotation of cell groups is not required and identifying cell types can be done automatically. Secondly, supervised learning techniques can take advantage of feature

selection in a grid search approach to select the most relevant features with high accuracy. Finally, previous studies revealed that handling the batch effect problem is less challenging in supervised methods.

In this study, we propose a deep learning approach that combines representation learning via self-organizing maps and deep learning classification using a convolutional neural network to identify human pancreas cell types on annotated scRNA-seq datasets. These techniques have been successfully applied to multi-omics cancer data ( [35]). In this work, they have been found suitable and very efficient for sc-RNA-seq data analysis as well. Fig. 4.1 represents the block diagram of the main components of the proposed method. After performing standard pre-processing steps, discussed in [68], including basic filtering, normalization, log transformation, and scaling, we followed the feature selection step to find the most informative genes. In this study, we used the Correlation-based Feature Selection (CFS) method, and thirteen most relevant genes were selected by maintaining high accuracy. Then, using the selected genes and Self-Organizing Maps (SOMs) algorithm, we create a ”template” to generate gene similarity networks (GSNs) of input cells. A SOM is a data structure that allows us to investigate the intrinsic relationships among samples of a dataset. As such, the relationships among data points can be visualized in a way that similar points be placed in the same group in the resulting graph (Fig. 4.1 step 1, ”Create SOM template”). We then create GSN for each cell by coloring template using corresponding gene expression values. Ideally, the GSN reveals the closeness of the most informative genes in terms of biological pathways or functionality. Therefore, the ”colored template” shows a representation of each cell based on the relationship among their marker genes. This step is shown in Fig. 4.1, step 2, ”Create images”. Finally, we detected cell types in the classification step using created images (the final step in Fig. 4.1).

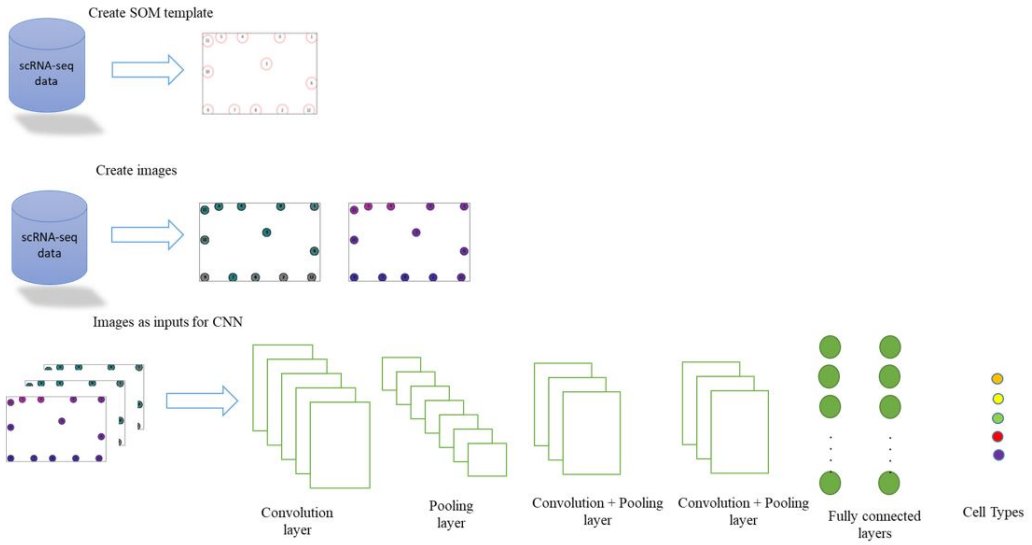


Figure 4.1: Block diagram with the main components of proposed method. Step (1). The relationships among data points can be visualized in a way that similar points be placed in the same group in the resulting graph via Self-Organizing Maps. Step (2). The ”colored template” shows a representation of each cell based on the relationship among their marker genes. In this figure, two sample images are presented. We follow the standard color scheme of images in which the three color channels include red, green, and blue, where each channel is represented via 256 distinct values. In this work, we used gene expression values as color channels to color ”templates”. It is worth mentioning that due to varying gene expression values in different cells, the marker genes are not equally informative for all cells. Step (3). Detecting cell types in the classification step. The CNN uses the transformed images as inputs to classify cells into different cell types.

## 4.2 Materials and Methods

### 4.2.1 Dataset

A public annotated scRNA-seq dataset of the human pancreas extracted from NCBI’s Gene Expression Omnibus [32] (GSE84133) [11], is used to evaluate the classification performance. The data was obtained by sequencing RNA in cells from the pancreas organ of different individuals and generated using inDrop protocols under the Illumina HiSeq 2500 platform. This dataset has been widely used in the publications of many scRNA-seq stud-

Table 4.1: Summary of the Pancreas dataset.

| **No. of Cell Types** | **No. of Cells** | **No. of Genes** |
|---|---|---|
| 14 | 8,569 | 20,125 |

ies [31], [57], [7], and [93]. The data including the transcriptomes of over 12K single pancreatic cells from four human donors and two mice. Cells in these six sub-populations are composed of a uniform distribution of fifteen distinct cell types including rare ghrelin-expressing epsilon-cells, vascular cells, activated pancreatic stellate cells, exocrine cell types, quiescent, Schwann cells, and four types of immune cells. As such, we used the data taken from one human donor with accession number GSM2230757 as a resource with fourteen distinct cell types to model the case wherein a new data point projected onto an annotated data set from the same tissue. The dataset includes gene expression profiles of 20,125 genes in 1,937 pancreatic cells. The statistics of dataset are listed in Table 5.1.

### 4.2.2 Data Pre-processing

This step includes basic filtering, normalization, log transformation, and scaling. To perform pre-processing, we followed the standard pre-processing pipeline [68] and [45] in the Python package, Scanpy [118]. First, we filtered out 5,387 low-expressed genes are detected in less than three cells, obtaining 14,739 genes. Investigating the distribution of the data shows low-quality cells. Based on Fig. 4.2, the number of genes expressed in the count matrix is mainly between 500 and 4,000 genes, and the distribution of several expressed genes over total count per cell is dense for less than 4,000 genes. As such, we filtered seven genes, i.e. the points above 4,000, in Fig. 4.2 $y$-axis, to remove those low-quality cells. Since the scRNA-seq data are expressed at different levels, normalization is desirable. The normalization applied in this study consists of translating numeric columns'

values in a dataset to a standard scale without distorting the ranges of values. We normalized the data using the Counts Per Million (CPM) normalization (Equation 4.1) combined with logarithmic scaling on the data:

$$CPM = readsMappedToGene \times \frac{1}{totalReads} \times 10^6 \tag{4.1}$$

where $totalReads$ is the total number of mapped reads of a sample, and $readsMappedToGene$ is the number of reads mapped to a selected gene. After per-gene quantification, we selected a subset of highly-variable genes to use in downstream analyses. To this end, we chose a routinely used strategy mentioned in [5], and defined the set of highly-variable genes given a normalized dispersion amount greater than 0.5 to obtain 2,546 genes.

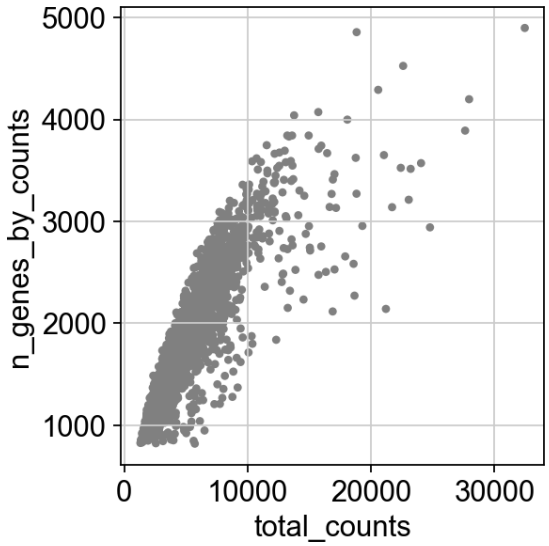


Figure 4.2: Distribution of genes in a read count matrix per total counts of reads. The number of genes expressed in the count matrix is mainly between 500 and 4,000 genes, and the distribution of several expressed genes over total count per cell is dense for less than 4,000 genes. As such, we filtered seven genes, i.e. the points above 4,000, to remove those low-quality cells.

### 4.2.3 Feature Selection

An essential step in machine learning methods is to identify a representative set of features that can largely affect the performance of a classifier and its computational complexity. It focuses on selecting the most informative features from a given dataset. In scRNA-seq data analysis, feature selection or gene extraction becomes an essential component due to the curse of dimensionality. This step is essential to avoid the problem of curse of dimensionality, and is achieved by dropping noise such as Mitochondrial genes, housekeeping genes, and other uninformative genes in the underlying feature vectors. The core motivation behind gene selection in cell type identification is that cell types are often distinguished from each other by only a few significant genes known as marker genes.

In this study, we used the Correlation-based Feature Selection (CFS) evaluation method for feature selection and then, found the list of marker genes using the reduced number of genes.

#### Correlation-based Feature Selection

Correlation-based feature selection (CFS) consists of ranking feature subsets in a search space including all possible feature subsets, an appropriate correlation measure, and a heuristic search strategy, GreedyStepwise. The CFS method selects those subsets of features that are strongly correlated with one of the classes, yet uncorrelated with each other. In this study, we consider genes as feature sets and cell types as classes. Irrelevant features are disregarded because of their low correlation with the classes. Redundant features are strongly correlated with one or more of the resting features, and so they are removed. The approval of a feature relies on the latitude to which it predicts only those classes that are not predicted by other features.

The heuristic "merit" of a feature subset $S$, $MS$, is given based on Equation 4.2:

$$M_s = \frac{k\overline{r_{cf}}}{k + k(k-1)\overline{r_{ff}}} \tag{4.2}$$

where $S$ is a feature subset that encompassing $k$ features. Also, $\overline{r_{cf}}$ is the mean correlation between feature $(f \in S)$ and classes, and $\overline{r_{ff}}$ is the average of inter-correlation between two features.

We found 2,000 genes as the most informative genes on normalized data using CFS method.

### 4.2.4 Creating a Gene Similarity Network via Self-Organizing Maps

To find marker genes, the method starts with the feature selection method discussed in the previous section. In this step, the number of features is initially set to 1,000. Finally, the thirteen marker genes were selected by maintaining high accuracy to create a gene similarity network via Self-Organizing Maps (SOM).

The SOM algorithm is mainly used for the visualization of the intrinsic relationships among data points. The input nodes in a SOM represent dimensions in the dataset and the output nodes are typically two-dimensional. Consider dataset $S1 = {s_{1j}}^{(i)}_{i,j=1}^{(n,m)}$, which contains genes $j = 1,2,3,...,m$ and cells $i = 1,2,3,...n$. A SOM is learned via an unsupervised clustering algorithm, which takes input cell vectors, and groups them based on similarities among their features (genes). The resulting graph is considered as a GSN on which similar points are placed in the same group and the distances among points representing the similarities in terms of biological pathways or functionality. In this study, we use the concept of "template" instead of GSN. As such, the "template" shows a representation per cell based

on the relationship among the most relevant genes. This step is shown in Fig. 4.1, step 1, ”Create SOM template”.

### 4.2.5 Cell Type Classification

The original idea of CNN is to map the image data to output the corresponding class. Although from a machine learning point of view scRNA-seq datasets are not presented in image format, we transform the high-dimensional sample points onto a low-dimensional map. Afterwards, to create an image per cell, the SOM ”templates” are colored using the corresponding gene expression values of selected marker genes for every single-cell in the input data (step 2, Create images, in 4.1). We follow the standard color scheme of images in which the three color channels include red, green, and blue, where each channel is represented via 256 distinct values. In this work, we used gene expression values as color channels to color ”templates”. It is worth mentioning that due to varying gene expression values in different cells, the marker genes are not equally informative for all cells. In the end, we have ”colored templates” as output images revealing the relationships among marker genes for each cell separately. Afterwards, the CNN uses these transformed images as inputs to classify cells into different cell types (Final step in 4.1).

We use somNet, a Python package that contains a SOM implementation, to create the templates. We update the SOM network neurons based on the Euclidean distance between gene $g_{ij}$ and the feature center $c_j$, as in Equation 4.3:

$$d_j = \sqrt{\sum_{i=1}^{n}(g_{ij} - c_{ij})} \tag{4.3}$$

where $n$ is the number of samples, $j$ is the current gene feature in the feature vector $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_j, \mathbf{v}_m\}$, and $m$ is the number of features. In somNet, neurons with smaller $d_j$

values are declared the winners of the competition to be the representatives of the data. In other words, that neuron is known as ”best matching unit” (BMU).

In addition, to avoid overfitting, we employed regularization and optimization methods in our trained model. To this end, we applied the dropout ratio of 0.2 in the first and second convolution layers, and 0.5 in the third convolution layer. Due to the sparsity of scRNA-seq data, the ADAM optimizer was applied during the training phase for faster convergence.

We used Keras 2.4.3 on TensorFlow 1.15.2 to learn this model. We ran the code on Google Colab utilizing TPU with 8 cores. Also, we used WEKA to apply the feature selection step to the dataset.

## 4.3 Results and Discussion

### 4.3.1 Experimental Results

Thirteen marker genes of the ”template” which found by the SOM are the following: ENG, PAX6, FKBP1A, PCSK2, GPX3, RBP4, HLA-B, S100A11, IFITM3, TM4SF1, IL32, TM4SF4, and IRX2. For example, Fig. 4.3 shows how the genes are arranged in the GSN template for a sample from Class 2 after coloring.

We trained our model on the created images per cell through the structure depicted in Fig. 4.1. The model was run 300 times inside 10-fold cross-validation, where the model learns from 90% of the samples and is tested on the remaining 10%. We evaluated our experimental results by four commonly used classification performance metrics, namely accuracy, precision, recall and F1-score. We calculated the performance of each class separately and then reported the average performance. Findings revealed stable and high prediction results with an average of 98-99.8% accuracy in a two-CONV layer CNN architecture.

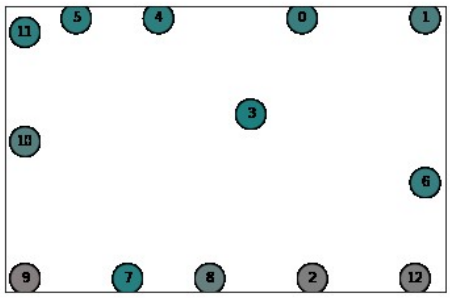


Figure 4.3: Image created for one single sample from Class 2. Thirteen marker genes which found by the SOM are arranged in the GSN template after coloring. We follwo the standard color scheme of images in which the three color channels include red, green, and blue, where each channel is represented via 256 distinct values. In this work, we used gene expression values as color channels to color "templates". It is worth mentioning that due to varying gene expression values in different cells, the marker genes are not equally informative for all cells.

Table 4.2: Performance metrics of proposed method.

| **Precision** | **Recall** | **F1-score** |
|---|---|---|
| 0.97 | 0.96 | 0.97 |

Also, other performance metrics, including precision, recall and F1-Score were in a high range, presented in Table 4.2. High prediction metrics ranging from 94%-100% for precision, 89%-100% for recall, and 94%-100% for F1-score can be seen across various classes. However, the average accuracy of our model is 98%.

The detailed structure of the CNN network that we used for our experiments is as follows:

- 32 windows of size $3 \times 3$ pixels in the first convolution layer followed by a rectified linear operator, ReLU, take the input samples.
- A max-pooling layer takes the maximal value of $2 \times 2$ regions with two-pixel strides and a local response normalization layer.

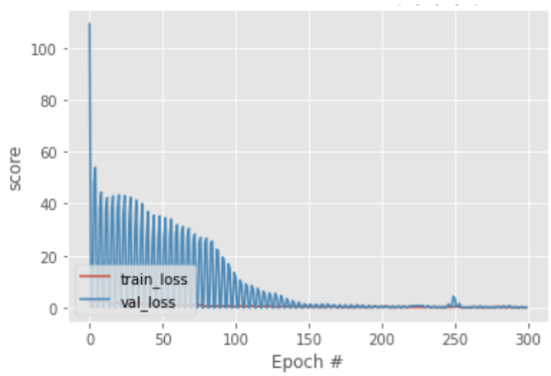


Figure 4.4: The plot of decreasing loss score by increasing the number of epochs ranging from 0 to 300. It can be observed that the model reaches its highest accuracy after 150 epochs.

- The output of the previous layer is then passed by another convolution layer containing 32 windows of size $3 \times 3$ pixels followed by ReLU, a max-pooling layer, and a normalization layer in which hyperparameters are the same as before.

- Two fully connected layers, FC, composed the next layer. The first FC layer contains 128 neurons, followed by ReLU with a dropout regularization technique. The second FC layer receives the output of the former FC layer and the output is six neurons, followed by ReLU with dropout.

- Finally, six output of the second FC is fed to a Softmax layer that assigns a probability to each cell type, where the Softmax selects the class with the maximum probability.

In each CNN, the initial learning rate was set to 0.05. Fig. 4.4 depicts the plot of decreasing loss score by increasing the number of epochs ranging from 0 to 300. It can be observed that the model reaches its highest accuracy after 150 epochs.

The findings show that only 13 marker genes are enough to obtain very accurate predictions. As such, the proposed method shows a lot of potential to apply it to predict various

Table 4.3: Muraro Pancreas Acinar Cell gene set along with gene description.

| Gene | Description |
|---|---|
| IFITM3 | interferon induced transmembrane protein 3 Source:HGNC Symbol;Acc:HGNC:5414 |
| FKBP1A | FKBP prolyl isomerase 1A Source:HGNC Symbol;Acc:HGNC:3711 |
| HLA-B | major histocompatibility complex, class I, B Source:HGNC Symbol;Acc:HGNC:4932 |
| IL32 | interleukin 32 Source:HGNC Symbol;Acc:HGNC:16830 |
| TM4SF1 | transmembrane 4 L six family member 1 Source:HGNC Symbol;Acc:HGNC:11853 |
| S100A11 | S100 calcium binding protein A11 Source:HGNC Symbol;Acc:HGNC:10488 |

cell types on a larger scale. The results are confirmed and validated by investigating the literature for the target cell types. Besides, the other significant results of using the proposed method on cell type identification are the power of finding marker genes automatically with high accuracy without searching in the literature. The results can be confirmed and validated by examining biological literature for the target cell types. Biological validation and interpretation of our results are discussed in the next section.

### 4.3.2 Biological validation

Investigating GSEA [99], we found six and five genes (overall 11 genes) related to Pancreas cell types with our thirteen highly-variable genes found in the reduced space (template) on two different gene sets, namely 'Muraro Pancreas Acinar Cell' and 'Muraro Pancreas Alpha Cell', respectively. These findings highlight the power and effectiveness of the proposed method to identify marker genes in scRNA-seq datasets. The genes related to two different cell types contained in our dataset, Acinar and Alpha, along with the description of their functionality are presented in Tables 4.3 and 4.4, respectively.

Table 4.4: Muraro Pancreas Alpha Cell gene set along with gene description.

| Gene | Description |
|---|---|
| GPX3 | glutathione peroxidase 3 [Source: HGNC Symbol; Acc: HGNC: 4555] |
| TM4SF4 | transmembrane 4 L six family member 4 [Source: HGNC Symbol; Acc:HGNC: 11856] |
| PAX6 | paired box 6 [Source: HGNC Symbol; Acc: HGNC: 8620] |
| IRX2 | iroquois homeobox 2 [Source:HGNC Symbol;Acc:HGNC:14359] |
| PCSK2 | proprotein convertase subtilisin/kexin type 2 [Source:HGNC Symbol;Acc:HGNC:8744] |

Thus, our method provides a new tool for discovering novel cell types as well as relevant genes.

## 4.4 Conclusion and Future Work

We have proposed a deep learning approach to identify cell types from single-cell RNA-seq data. Our proposed method uses a combination of a self-organizing map and a convolutional neural network to perform dimensionality reduction, feature selection, and classification, concurrently. By creating a template using the SOM learning algorithm, which contains the most informative genes in the gene similarity network template, we found the transformed representation of cells using only 13 genes on a two-dimensional space. Next, using a CNN, we detected populations of cell types in the human pancreas on the test dataset with an accuracy rate of 98%.

As a future extension, scRNA-seq data can be integrated with other data such as cell location, which is usually missed during the single-cell sequencing process, to obtain more reliable results in large-scale experiments. Indeed, to find all cell types in the body, both currently known and novel unknown cell types, it is required to provide multi-omics in

the input layer. Moreover, the proposed approach can be considered as an unsupervised clustering algorithm, because of the competitive and incremental learning nature of SOMs. Therefore, a large number of unlabeled samples can be used in conjunction with a small number of labelled scRNA-seq data to cell type identification on larger scale data.

# Chapter 5

# Comparative Analysis of Supervised Cell Type Detection in Single-Cell RNA-seq Data

## 5.1 Introduction

Tumor heterogeneity is a common phenomenon in studying different types of cancer. In this regard, novel techniques such as single-cell RNA sequencing (sc-RNA sequencing) can be used to detect unknown tumors and consequently drug discovery, better treatment, diagnosis, and prognosis. Thus, one of the first fundamental steps to perform an in-depth analysis of single-cell sequencing data consists of identifying cell types. Hidden diversity and characteristics of a particular cell type can be found via deferentially expressed genes or marker genes.

Supervised or unsupervised learning approaches can effectively be used to identify various cell types depending on the dataset, annotated or unannotated, respectively. Typically,

in single-cell RNA-seq downstream analysis, clustering techniques are used to reveal well-separated clusters of cells and annotate them manually with different cell types using canonical markers and reference databases. Different clustering methods try multiple parameters to achieve higher performance. Setting up the clustering parameters, such as the number of clusters, is a challenging point [108]. For example, several clustering methods are compared in [31]. Among them, SC3 [56], CIDR [65], Ascend [96], SAFE-clustering [123], and TSCAN [51] all posses built-in methods for estimating the optimal number of clusters. However, Ascend and CIDR underestimated the number of clusters, whereas SC3 and TSCAN tend to overestimate. Moreover, manually annotating the obtained clusters using differential expression analysis is time-consuming and non-reproducible in clustering methods.

On the other hand, classification techniques have increasingly developed to identify cell types automatically instead of manually annotating clusters of cells. In addition to this, different feature selection techniques can be used to avoid the "curse of dimensionality" and select a reduced number of the significant marker genes. A comparative study in abdelaal2019comparison discussed 22 supervised techniques, including random forest classifier (RF), *k*-nearest neighbor (*k*-NN), support vector machine (SVM). One of the challenges covered in this study is feature selection. Three different cell-specific purpose feature selection techniques have been used, including random gene selection, highly variable genes (HVG) selection, and selecting genes based on the number of dropouts (zero expression). They benchmarked their experiments based on the number of features. The findings show that the performance of the classifiers highly depends on the number of cells and genes, selected marker genes, and dataset complexity. In this study, we used general-purpose techniques, instead of cell-specific ones, to compare three state-of-the-art feature selection

techniques combined with three popular classifiers to complement the feature selection step. We also biologically validate cell type marker genes identified by the best feature selection method.

## 5.2 Materials and Methods

### 5.2.1 Framework

Three general-purpose feature selection methods, namely ANOVA F-value, Chi-squared, and information gain (IG), along with three state-of-the art classification methods, including SVM, *k*-NN, and RF, are used in our experiments to identify cell types automatically. A comparative study on scRNA-seq data is done in this work. To this end, we followed the pipeline depicted in Fig. 5.1. First, we performed pre-processing steps, including filtering, normalization, and scaling. Then, to find the best parameters for the classification methods, hyperparameter tuning and optimization were done on pre-processed data. The most informative features were extracted in the feature selection step. Three classifiers combined with three feature selection algorithms were evaluated to find the best model. Finally, cell types are predicted by the method with higher accuracy.

It is worth mentioning that although there a other state-of-the-art classification methods including deep learning ones, feeding this group of methods require rich labeled datasets which is the main limitation of sc-RNA seq datasets. We used the Scikit-learn in Python version 3.7 to perform the feature selection and classification methods [83].

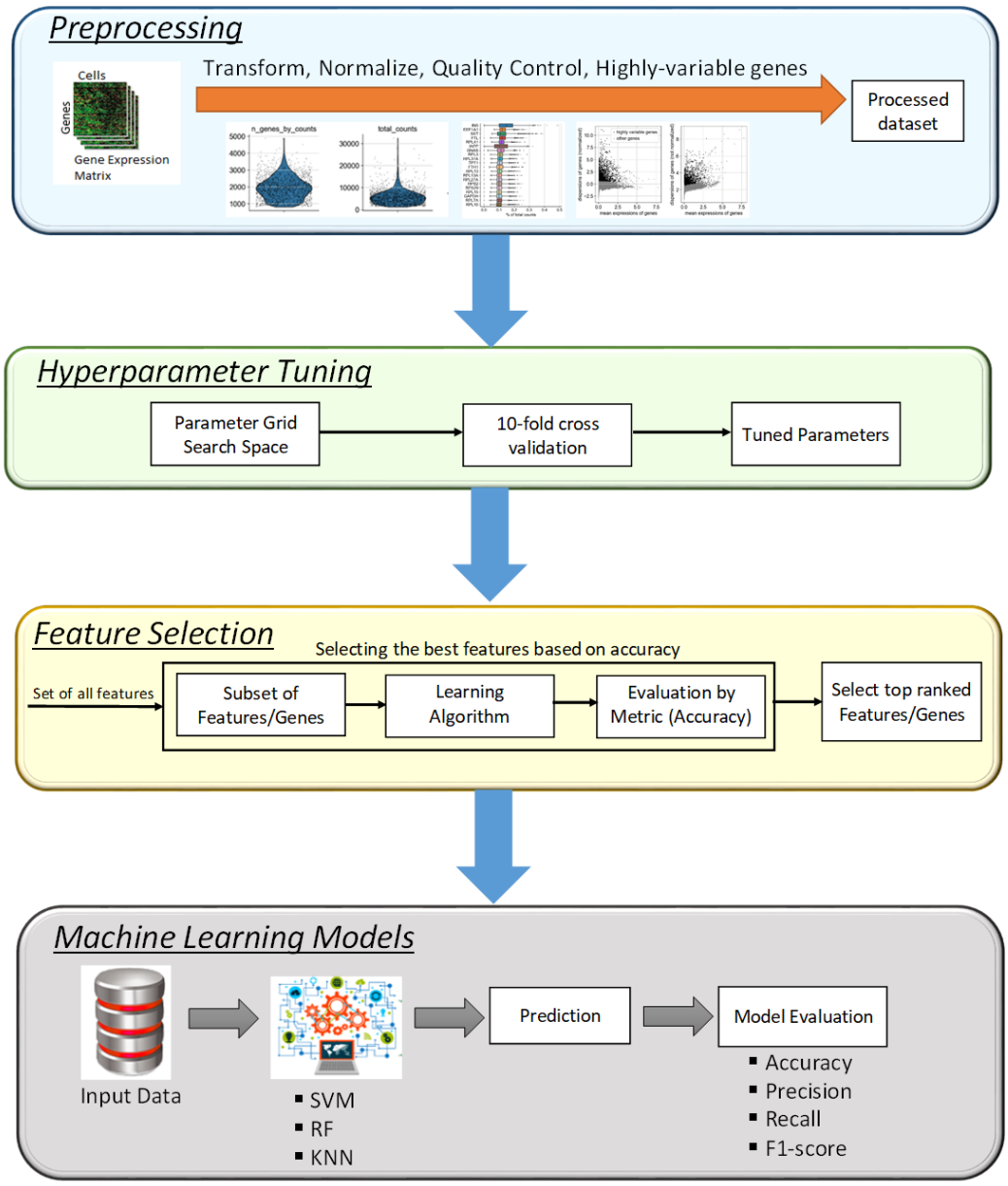


Figure 5.1: Pipeline overview of the experiments.

Table 5.1: Details of the datasets studied in this work.

| Dataset | Tissue | Accession # | Cell Types # | Cells # | Genes # |
|---|---|---|---|---|---|
| Baron-human1 (Data1) | Human-Pancreas | GSM2230757 | 8 | 1,937 | 20,125 |
| Baron-human2 (Data2) | Human-Pancreas | GSM2230758 | 8 | 1,724 | 20,125 |
| PBMC (Data3) | Prepheral Blood | 10X V2 | 9 | 23,154 | 22,280 |

### 5.2.2 Dataset

Public, annotated scRNA-seq data sets with the accession number of GSM2230757 and GSM2230758 under series GSE84133 [10], and PBMC 10X V2 were extracted from NCBI's Gene Expression Omnibus [32] and used in this article to evaluate the classification performance. These datasets include transcripts of pancreatic and peripheral blood cells from human donors. Pancreatic cells are divided into eight groups of previously characterized cell types: alpha, beta, acinar, delta, quiescent, activated pancreatic stellate cells, endothelial, and ductal cells. The existence of these cell types is validated with immuno-histochemistry stains [10] so that it can be a good resource for the discovery of cell types. Also, the PBMC dataset includes nine different cell types. The details of datasets are listed in Table 5.1.

### 5.2.3 Data Pre-processing

Raw read count matrices generated using next-generation sequencing technologies contain low-quality sequencing information based on the expression levels. Pre-processing step (Fig. 5.1) is to ensure removing any weakly expressed genes or low-quality cells, including damaged, dead, or degraded during sequencing, and are represented by a low number of expressed genes in the read count matrices. To perform pre-processing, we followed

the standard pre-processing pipeline in scRNA-seq data analysis [68]. According to this pipeline, cells with less than 200 expressed genes, and genes expressed in less than three cells are filtered out. In Data1, for example, we first filtered out 5,387 low-expression genes that were detected in less than three cells and kept 14,739 genes. Further analysis of the data distribution showed low-quality cells and led to removing seven cells. After per-gene quantification, we selected a subset of highly variable genes to use in downstream analyses. To this end, we chose a common strategy routinely used [5] and defined the set of highly variable genes given a normalized dispersion higher than 0.5 after normalization and obtained 2,546 genes at the end. We used Scanpy [118], a specifically designed package to work with scRNA-seq datasets, for pre-processing steps.

### 5.2.4 Feature Selection

In scRNA-seq data analysis, feature selection or gene selection can be an essential component due to the curse of dimensionality. The primary motivation behind feature selection or gene selection in cell type identification is that cell types are often distinguished by only a few essential genes known as biomarkers. This study investigated three general-purpose feature selection approaches, including Analysis of Variance (ANOVA) F-value, Chi-squared, and information gain (IG) to select a sorted list of genes. The best number of genes for the training model is chosen by calculating the model's performance for top $k$ genes where $k = 100, 200, 300,$ and 400. We evaluated the accuracy of the methods by varying the number of marker genes based on different computational approaches.

### Analysis of Variance (Anova) F-value

ANOVA F-value assumes that there is a linear relationship between variables and target, and also the variables are normally distributed. It uses F-tests to statistically measure the ratio of two variances, i.e. how far the data points are dispersed from the mean. The results show the statistical significance of the test. F-value is a very important part of ANOVA and is calculated by the Equation 5.1.

$$F = \frac{\sigma_1^2}{\sigma_2^2} \tag{5.1}$$

where $F$ is the F-value, $\sigma_1$ is the larger sample variance and $\sigma_2$ is the smaller sample variance.

### Chi-squared

Pearson's Chi-squared test or just Chisquared test is a statistical test applied to the categorical features to test the relationships among them. It is suited for non-negative variables and mostly boolean, frequencies, or counts. It uses frequency distribution of the features to determine the correlation or association among them. The test calculates chi-squared statistics i.e. the expected frequencies of the observations and then determines whether the observed frequencies match the expected frequencies. The Equation 5.2 shows how this method calculates the correlation among features.

$$\chi^2 = \Sigma \frac{(ObservedFrequency - ExpectedFrequency)^2}{Expected} \tag{5.2}$$

where $\chi^2$ is Chi-squared.

**Information Gain**

Information Gain is defined in terms of uncertainty. The lesser the information gain, the higher the uncertainty. If $IG(X) > IG(Y)$, it means feature $X$ will be better and preferred where $IG(X)$ represents the information gain from feature $X$. The relevance of feature is estimated by considering the information gain for each feature and choosing the one with maximum value. It is defined as the difference between prior uncertainty and uncertainty after considering feature $X$ as shown in Equation 7.2 [88].

$$IG(X) = \sum_i U(P(C_i)) - E\left[\sum_i U(P(C_i|X))\right] \tag{5.3}$$

where $U$ represents uncertainty function, $P(C_i)$ represents probability of class $C_i$ before considering feature $X$ and $P(C_i|X)$ represents posterior probability of class $C_i$ after considering feature $X$.

### 5.2.5 Evaluation Metrics

We applied the most commonly used evaluation metrics, namely accuracy, precision, recall, and F-score to systematically estimate and compare the performance of different methods. To this end, we used 10-fold cross-validation to test and train the model.

## 5.3 Results and Discussion

### 5.3.1 Parameter Optimization

To select the best parameters of the classifiers ($K$-NN, RF, and SVM), we used a Bayesian model-based optimization approach with Gaussian as an adaptive hyperparameter search.

It is a fast approach compared to grid search and random search. We employed Bayesian search to tune hyperparameters, which rather than scanning the hyperparameter space mindlessly (as in the grid or random search), this strategy emphasizes the use of knowledge obtained in one step to discover the next set of hyperparameters that would improve model performance. This method, in an iterative manner, continues until the optimal result is obtained. Since it prioritizes hyperparameters that appear more promising from previous steps, the Bayesian technique is able to find the best hyperparameters in less time (fewer iterations) than grid search and random search.

The best parameter based on the optimization results for each classification method for Data1, Data2 and Data3 are presented in Table 5.2.

Table 5.2: The best parameters for each method obtained using Bayesian Optimization for the datasets.

| **Method** | **Best Parameters Found** | | | |
|---|---|---|---|---|
| | | **Data1** | | |
| **K-NN** | $k$= 5 | | | |
| **RF** | $n\_estimators$ = 359 | $max\_depth$ = 41 | $criterion$ = 'gini' | $max\_features$ = 'sqrt' |
| **SVM** | $C$ = 0.5 | $gamma$ = 0.2 | $kernel$ = 'linear' | |
| | | **Data2** | | |
| **K-NN** | $k$ = 4 | | | |
| **RF** | $n\_estimators$ = 100 | $max\_depth$ = 1 | $criterion$ = 'gini' | $max\_features$ = 'sqrt' |
| **SVM** | $C$ = 0.5 | $gamma$ = 0.2 | $kernel$ = 'linear' | |
| | | **Data3** | | |
| **K-NN** | $k$ = 6 | | | |
| **RF** | $n\_estimators$ = 495 | $max\_depth$ = 54 | $criterion$ = 'gini' | $max\_features$ = 'sqrt' |
| **SVM** | $C$ = 0.1 | $gamma$ = 0.2 | $kernel$ = 'poly' | $degree$ = 2 |

For selecting the best value of $k$ for the $k$-NN classifier, the following values of the $k$ = (4,5,6) in the search space are inspected. The quality of the result is determined by $k$ with

the highest average accuracy of the three feature selection methods.

For RF, the following values for the search space are investigated: $n\_estimators$ = (100, 500), $max\_features$ = (sqrt, log2), $max\_depth$ = (1, 60) and $criterion$ = (gini, entropy). The $n\_estimators$ parameter are the number of trees to be considered. The parameter $max\_features$ are the maximum number of features to be considered for individual tree. $max\_depth$ parameter is the maximum depth of the tree where maximum depth is defined as the longest path from root node to the leaf node and the parameter $criterion$ is the function which is used to evaluate the quality of split.

RF, by default, uses built-in feature selection methods, including 'Ginni' and 'entropy'. To ensure that each method uses its approach for classification, we allowed RF to use this ability during the training process with a list of selected features using the feature selection methods.

For SVM, the following values for the search space are inspected: $C$ = (0.1,0.5,1), $gamma$ = (0.1, 0.2, 0.3), $degree$ = (1,8) and $kernel$ = ($rbf$, $poly$, $linear$). The regularization parameter, aka the cost of misclassification, $C$, is a degree of importance that is given to the misclassifications error. SVM seeks a trade-off to maximize the margin among the classes and minimize the number of misclassifications. The larger the value of C, the larger is the miss-classification cost. Kernels are functions used to solve non-linear problems by making a curvative hyperplane to separate classes. The parameter *gamma* decides the curvature in the decision boundary in non-linear kernels, where a large value of gamma means more curvature, i.e., softer and tends to overfit the data.

### 5.3.2 Classification Results

To investigate the effect of the selected features (genes) as a form of prior knowledge, we evaluated the performance of the classifiers based on the different number of selected features using three different approaches. We examined $k$ features where $k = 100, 200, 300,$ and 400 to determine the best number of features to optimize the performance of the classifier. The best value of $k$ with the highest accuracy of a combination of each feature selection and classification method for Data1 is shown in Table 5.3.

For Data1, the $k$-NN classification method results reveal a high accuracy of 96.11% with 400 features when using the Information Gain feature selection method. The RF classification method for Data1 indicates high accuracy with 400 features for all three feature selection methods. A combination of this classifier with IG gives the best accuracy of 97.05%. Observing the results of the SVM classification method for Data1, all three feature selection methods reveal high accuracy with 400 features. Again, SVM combined with IG gives the highest accuracy of 98.08%. Among all the combinations, SVM combined with IG shows highest performance with 98.08% accuracy for Data1.

For Data2 among all the combination, $k$-NN classification method achieves high accuracy (94.66%) with 200 features selected from IG feature selection method, RF and IG combination achieves high accuracy (96.06%) with 400 features, and lastly, SVM achieves high accuracy with 400 features (98.09%) selected from IG feature selection method. For Data2, SVM coupled with IG provides the best performance, with 98.09% accuracy.

For Data3, SVM achieves highest performance (84.91% accuracy) with 200 features selected from Anova feature selection method. In general, SVM outperformed the other

Table 5.3: Classification accuracy obtained by three classification methods combined with feature selection methods through selected features for Data1.

| **Method** | **Features** | **Accuracy %** |
|---|---|---|
| | ***k*-NN** | |
| **ANOVA F-value** | 400 | 95.65 |
| **Chi-squared** | 400 | 93.99 |
| **Information Gain** | 400 | 96.11 |
| | **Random Forest (RF)** | |
| **ANOVA F-value** | 400 | 96.74 |
| **Chi-squared** | 400 | 96.84 |
| **Information Gain** | 400 | 97.05 |
| | **SVM** | |
| **ANOVA F-value** | 400 | 97.72 |
| **Chi-squared** | 400 | 96.79 |
| **Information Gain** | 400 | 98.08 |

two classification methods for all three datasets.

To generalize our experiments, we used two datasets with the same number of genes and the different number of cells (i.e., Data1 and Data2), and another dataset with the higher number of cells and genes (i.e., Data3) comparatively. Other metrics are presented in Figures 5.3.2, 5.3, and 5.4.

Among all combinations of classification and feature selection methods, SVM combined with IG significantly outperformed other approaches. High accuracy of 98.08% for Data1 means that the features that have been selected are highly correlated and significantly help fulfill our primary objective.

Our results highlight the power of the SVM classifier combined with the IG as the best approach. Also, it shows that the performance of classifiers highly depends on the selected marker genes using different techniques.

[width=0.95]Chapters/Chapter05/Images/diagrams$_1$.*eps*

Figure 5.2: Average performance of the SVM classifier combined with three feature selection methods.

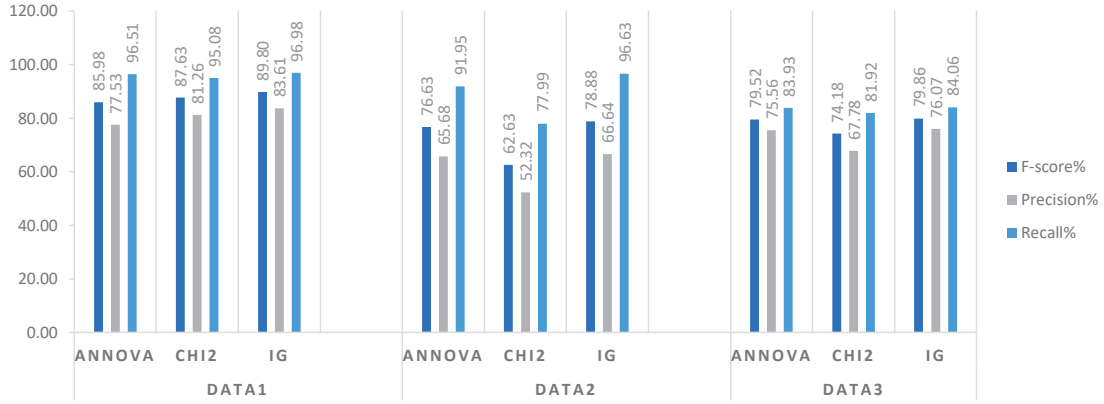


Figure 5.3: Average performance of the *k*-NN classifier combined with three feature selection methods.

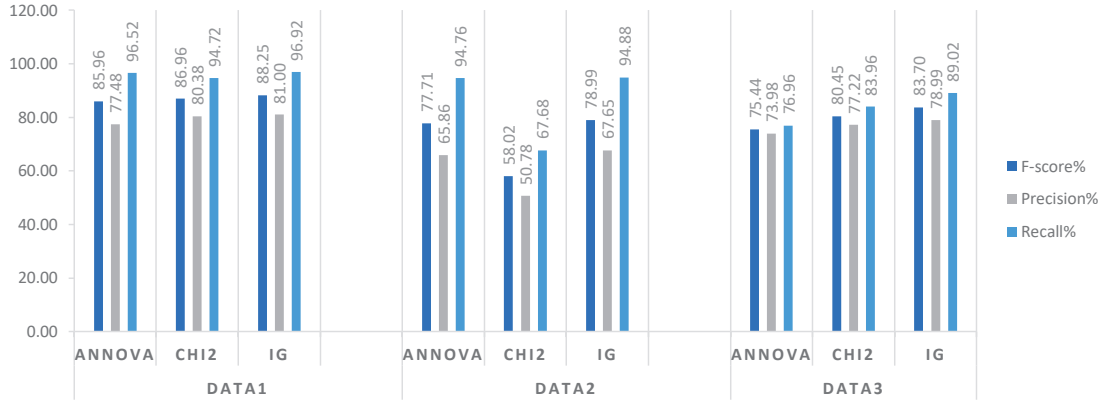


Figure 5.4: Average performance of the RF classifier combined with three feature selection methods.

### 5.3.3 Biological Validation

We evaluated the performance of our method for detecting cell types using the high-ranked features or deferentially expressed genes through investigating the current literature and reference databases. By investigating GSEA [99] on the result of Data1, we found 9 out of 20 overlapped genes between Pancreas gene sets, ”Muraro Pancreas Endothelial Cell”, and

Table 5.4: Muraro Pancreas Endothelial Cell gene set.

| Gene Symbol | Description |
|---|---|
| IFITM3 | interferon induced transmembrane protein 3 [Source:HGNC Symbol;Acc:HGNC:5414] |
| IGFBP4 | insulin like growth factor binding protein 4 [Source:HGNC Symbol;Acc:HGNC:5473] |
| IFITM2 | interferon induced transmembrane protein 2 [Source:HGNC Symbol;Acc:HGNC:5413] |
| COL4A1 | collagen type IV alpha 1 chain [Source:HGNC Symbol;Acc:HGNC:2202] |
| SPARC | secreted protein acidic and cysteine rich [Source:HGNC Symbol;Acc:HGNC:11219] |
| IGFBP7 | insulin like growth factor binding protein 7 [Source:HGNC Symbol;Acc:HGNC:5476] |
| VIM | vimentin [Source:HGNC Symbol;Acc:HGNC:12692] |
| TM4SF1 | transmembrane 4 L six family member 1 [Source:HGNC Symbol;Acc:HGNC:11853] |
| HLA-B | ”major histocompatibility complex, class I, B [Source:HGNC Symbol;Acc:HGNC:4932]” |

top genes found by our method. The list of 9 overlapped genes, along with the description of their functionality, is depicted in Table 5.4. Moreover, we conducted a biological validation on the other datasets, Baron Human2 (Data2) and PBMC(Data3). The results are depicted on Tables 5.5 and 5.6. Overall, our results show the power of our method to identify the cell types using a list of marker genes in scRNA-seq datasets.

## 5.4 Conclusion and Future Work

This work focuses on the supervised identification of cell types using feature selection methods combined with classification techniques on an annotated dataset. Investigating similarities among features using three state-of-the-art feature selection methods to reduce

Table 5.5: Muraro Pancreas Ductal Cell gene set.

| **Gene Symbol** | **Description** |
|---|---|
| CDC42EP1 | CDC42 effector protein 1 [Source:HGNC Symbol;Acc:HGNC:17014] |
| PMEPA1 | ”prostate transmembrane protein, androgen induced 1 [Source:HGNC Symbol;Acc:HGNC:14107]” |
| TACSTD2 | tumor associated calcium signal transducer 2 [Source:HGNC Symbol;Acc:HGNC:11530] |
| KRT7 | keratin 7 [Source:HGNC Symbol;Acc:HGNC:6445] |
| SDC4 | syndecan 4 [Source:HGNC Symbol;Acc:HGNC:10661] |
| KRT19 | keratin 19 [Source:HGNC Symbol;Acc:HGNC:6436] |
| FLNA | filamin A [Source:HGNC Symbol;Acc:HGNC:3754] |
| IFITM3 | interferon induced transmembrane protein 3 [Source:HGNC Symbol;Acc:HGNC:5414] |
| SERPING1 | serpin family G member 1 [Source:HGNC Symbol;Acc:HGNC:1228] |
| COL18A1 | collagen type XVIII alpha 1 chain [Source:HGNC Symbol;Acc:HGNC:2195] |

the dimension of the feature space helps enhance the classification task and overcome its inherent computational complexity. Finding similarities can result from linear or non-linear relationships among the features, data distribution, or data entropy. Biologically speaking, the similarity is defined by structural, functional, or evolutionary relationships among the genes that lead to finding the most accurate class for a new test sample. In our experiments, we have demonstrated that genes in our dataset that have similar expression patterns were grouped in highly-scored classes. Identifying biomarker genes that are differentially expressed among different cell types is done in the feature selection step. This work highlights the power of using only a sub-group of highly effective genes to find cell types. Thus, we can take advantage of disregarding a considerable number of uninformative genes for identifying the corresponding cell types. Moreover, there are some potential future avenues to find cell types automatically using scRNA-seq data. For example, conducting a com-

Table 5.6: Travaglini Lung Ereg Dendritic Cell gene set.

| **Gene Symbol** | **Description** |
|---|---|
| HLA-DPB1 | ”major histocompatibility complex, class II, DP beta 1 [Source:HGNC Symbol;Acc:HGNC:4940]” |
| TYROBP | transmembrane immune signaling adaptor TYROBP [Source:HGNC Symbol;Acc:HGNC:12449] |
| HLA-DPA1 | ”major histocompatibility complex, class II, DP alpha 1 [Source:HGNC Symbol;Acc:HGNC:4938]” |
| AIF1 | allograft inflammatory factor 1 [Source:HGNC Symbol;Acc:HGNC:352] |
| LST1 | leukocyte specific transcript 1 [Source:HGNC Symbol;Acc:HGNC:14189] |
| FCER1G | Fc fragment of IgE receptor Ig [Source:HGNC Symbol;Acc:HGNC:3611] |
| HLA-DQB1 | ”major histocompatibility complex, class II, DQ beta 1 [Source:HGNC Symbol;Acc:HGNC:4944]” |
| CST3 | cystatin C [Source:HGNC Symbol;Acc:HGNC:2475] |
| FCN1 | ficolin 1 [Source:HGNC Symbol;Acc:HGNC:3623] |
| VCAN | versican [Source:HGNC Symbol;Acc:HGNC:2464] |
| HLA-DRB1 | ”major histocompatibility complex, class II, DR beta 1 [Source:HGNC Symbol;Acc:HGNC:4948]” |
| GPX1 | glutathione peroxidase 1 [Source:HGNC Symbol;Acc:HGNC:4553] |
| GZMB | granzyme B [Source:HGNC Symbol;Acc:HGNC:4709] |

prehensive experiment using a more significant number of samples obtained from different tissues shows potential in enhancing the results on a larger scale.

# Chapter 6

# Unsupervised Identification of SARS-CoV-2 Target Cell Groups via Nonlinear Dimensionality Reduction on Single-cell RNA-Seq Data

## 6.1 Introduction

Single-cell sequencing is an emerging technology used to capture cell information at a single-nucleotide resolution and by which individual cell can be analyzed separately [41]. As of now, all available single-cell RNA-seq (scRNA-seq) data have been generated for different purposes [44]. However, these high-dimensional and sparse data lead to some analytical challenges. Analyzing scRNA-seq data can be divided into two main categories: at the cell level and gene level. Finding cell sub-networks or highly deferentially expressed

tissue-specific gene lists is one of the common challenges at the cell level [91]. Arranging cells into clusters to find the heterogeneity in the data is arguably the most significant step of any scRNA-seq data downstream analysis. This step could be used to distinguish tissue-specific sub-networks based on identified gene sets. Indeed, cell clustering aims to identify cell sub-types based on the patterns embedded in gene expression without prior knowledge at the cell level. Since the number of genes that are profiled in scRNA-seq data is typically large, cells tend to be located close to each other following nonEuclidean, but a complex relationship in high-dimensional spaces [55]. Therefore, traditional clustering algorithms are unsuitable for this challenge, and hence, are not able to efficiently separate individual cell types. To alleviate this problem of the curse of dimensionality, several algorithms have been proposed to accurately cluster cells from scRNA-seq profiles.

Dimensionality reduction techniques have been widely used in several studies of large-scale scRNA-seq data processing [29]. Most of the previous studies use principal component analysis (PCA). However, there was no advantage in keeping the clustering performance after the changes in the data in lower dimensions [36]. Other works have also employed PCA as a pre-processing step to remove cell outliers for performing dimensionality reduction and visualization. Moreover, several studies have used unsupervised clustering models to identify rare novel cell types. For instance, the hierarchical clustering algorithm divides large clusters into smaller ones or merge each data points into larger clusters progressively. This algorithm has been employed to analyze scRNA-seq data by BackSPIN [126] and pcaReduce [136], through dimension reduction after each division or combination in an iterative manner. $k$-Means which is one of the most common clustering algorithms, has been employed in the Monocle, specifically for analyzing scRNA-seq data [87]. Also, the authors of [119] used the Louvain algorithm, which is based on com-

munity detection techniques to analyze complex networks [43]. However, to achieve ac-

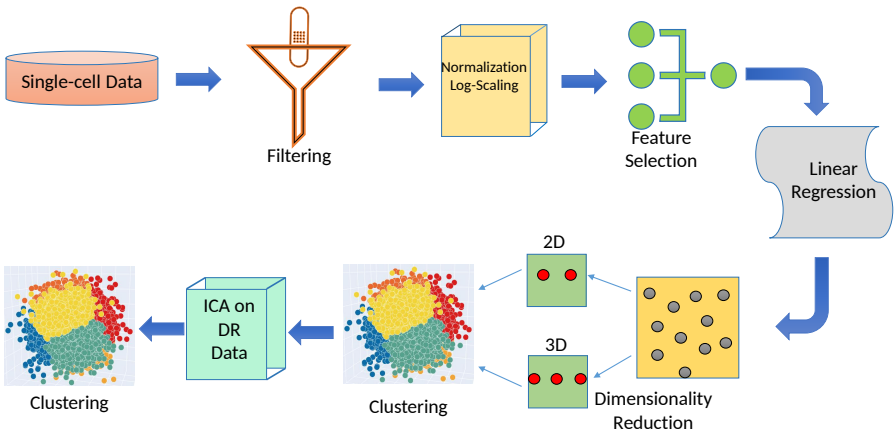


Figure 6.1: Block diagram of the proposed approach.

ceptable clustering performance on scRNA-seq data, other comprehensive studies indicated that hybrid models, designed as a combination of clustering and dimensionality reduction techniques, tend to improve the clustering results [36]. They learned 20 different models using four dimensionality reduction method including PCA, non-negative matrix factorization (NMF), filter-based feature selection (FBFS), and Independent Component Analysis (ICA). They also used five clustering algorithms such as $k$-means, density-based spatial clustering of applications with noise (DBSCAN), fuzzy $c$-means, Louvain, and hierarchical clustering. Their experiments highlight the positive effect of hybrid models and showed that using feature-extraction methods could be a good way to improve clustering performance. Their experimental results indicate that Louvain combined with ICA performed well in small feature spaces.

In this paper, we proposed a model to obtain efficient and meaningful clusters of cells

from large-scale COVID-19 scRNA-seq data. We focus on the combination of unsupervised dimensionality reduction followed by conventional clustering methods. We investigated different non-linear dimensionality reduction and manifold learning methods such as standard Locally Linear Embedding (LLE), modified LLE, and Laplacian eigenmaps. Also, ICA is employed to enhance visualization and clustering of the data, and combined with *k*-means clustering. Experimental results on a well-known scRNA-seq dataset show the power of modified LLE and ICA on clustering data in very low dimensions, providing very high accuracy and enhanced visualization.

## 6.2 Materials and Methods

The block diagram of our proposed approach is depicted in Fig. 6.1. Based on the main pipeline, the scRNA-seq data is pre-processed based on the number of cells and the number of genes. Filtered data is then normalized and scaled. Highly variable genes are extracted as part of the feature selection step, and linear regression is one of the most widely-used methods to correct technical artifacts present in the data based on the total counts per cell and mitochondrial percentage as discussed in [119] [70]. The data obtained at this point is then processed to reduce the dimensions of the feature space into two or three dimensions; afterwards, *k*-means clustering is applied. Besides, We performed ICA on the lower-dimensional data followed by *k*-means clustering to achieve meaningful clusters and enhanced visualization.

### 6.2.1 Dataset

The data used in this study is a gene expression profile dataset extracted from NCBI's Gene Expression Omnibus [103], accession number GSE148729 [121]. The data contains 27,072 gene expression profiles of 48,890 human lung cell lines, which were sequenced using Illumina NextSeq 500. In this dataset, different cell lines from lung tissue, which is one of the main cellular components in the immune system, were contaminated with SARS-CoV-1 and SARS-CoV-2 and sequenced at different time slots to study the impact of infection on immune system over time.

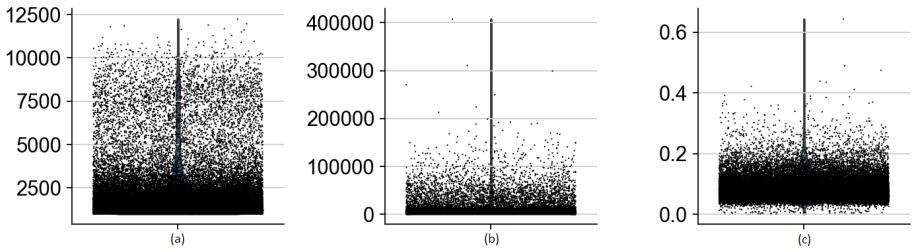


Figure 6.2: (a) The number of expressed genes, (b) the total counts per cell, and (c) the percentage of mitochondrial genes.

### 6.2.2 Data Pre-processing and Quality Control

This step includes filtering out genes and cells based on quality metrics, normalization and scaling, feature selection, and quality control. The Python package, Scanpy, is used to perform pre-processing and quality control. To this end, we follow the typical scRNA-seq analysis workflow, as described in [70]. As shown in Fig. 6.1, the first step of pre-processing is to filter poorly expressed genes. Low-quality cells that are dyed, degraded, or damaged during sequencing are represented by a low or large number of expressed genes. As such, we filtered out 6,066 genes expressed in less than three cells and cells with less than 200

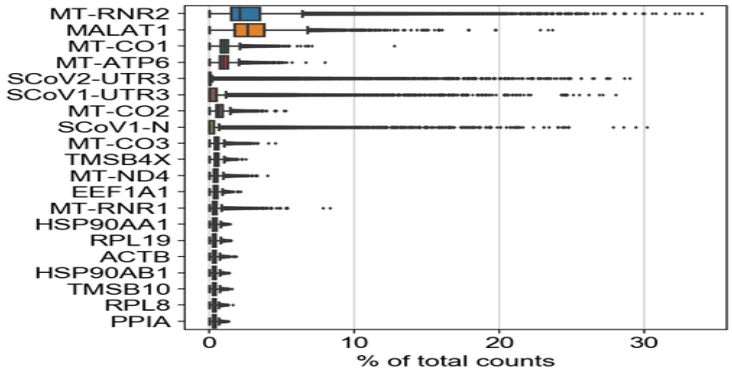


Figure 6.3: Top 20 highly-variable genes before normalization.

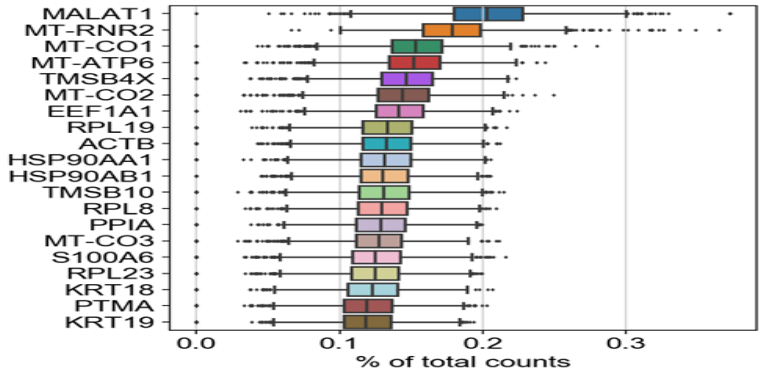


Figure 6.4: Top 20 highly-variable genes after normalization.

expressed genes. Moreover, we removed a large number of mitochondrial genes, which are the result of damaged cells [50], [48]. To remove low-quality cells, we investigated the distribution of data to estimate quality control metrics. Based on Fig. 6.2, the number of expressed genes, i.e., the left plot (Fig. 2a) of the figure are mainly between 500 and 2,500 genes. Also, the distribution of the proportions of mitochondrial genes, i.e., the right plot (Fig. 2c) of the figure, contains very extreme values, above 0.05. We extracted the number of genes that are less than 2,500 and mitochondrial genes less than 5%. Plot in the middle (Fig. 2b) represents total number of samples per cell.

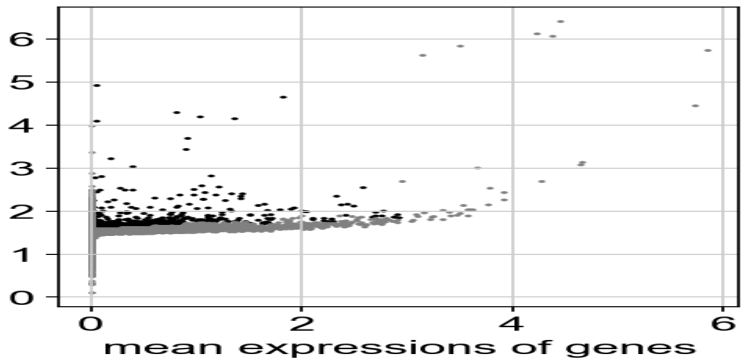


Figure 6.5: Dispersion of genes before normalization.

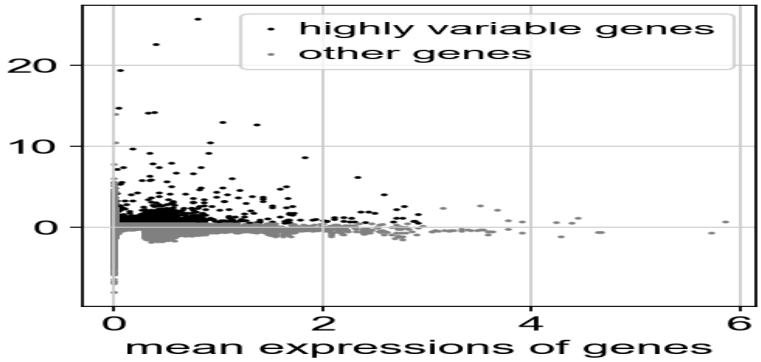


Figure 6.6: Dispersion of genes after normalization.

Then, we normalized the data using the Counts Per Million (CPM) normalization combined with logarithmic scaling on the data:

$$CPM = readsMappedToGene \times \frac{1}{totalReads} \times 10^6 \tag{6.1}$$

where $totalReads$ is the total number of mapped reads of a sample, and $readsMappedToGene$ is the number of reads mapped to a selected gene.

At this point, we extracted highly variable genes (HVGs) as a part of the feature selection step, aiming at minimizing the search space. We then removed any random noise and held genes that highlight relevant biological information. Highly-variable genes are those

genes that are expressed more or less in some cells compared to other ones. Quality control makes sure that the differences occur because of biological differences and not technical noise. The simplest approach to compute such a variation is to quantify the variance of the expression values for each gene across all samples. Here, we use log-normalized data because we want to ensure having the same log-values in the clustering and dimensionality reduction follow a consistent analysis through all steps. To perform feature selection, a good mean-variance relationship is desired. Also, a good trade-off value would help select the subset of genes that keep useful biological knowledge, while removing noise. There are several widely-used approaches to find the best threshold. Based on Figs. 6.6 and 6.5, we used a minimum of 0.5 for normalized dispersion, a maximum mean of 3, and a minimum mean of 0.0125 to select relevant genes. Finally, we obtained 2,194 genes with 3,791 cells for downstream analysis. The normalized dispersion is obtained by scaling the mean and standard deviation of the dispersion for genes falling into a given bin for the mean expression of genes. This means that for each bin of mean expression, highly-variable genes are selected. The 20 top genes extracted before and after normalization are shown in Figs. 6.4 and 6.3.

### 6.2.3 Dimensionality Reduction

High-dimensional gene expression data is unprecedentedly rich and should be well-explored. In a single-cell expression profile, each gene appears as a dimension of the data. As such, dimensionality reduction techniques tend to summarize biological features in fewer dimensions. With two genes, we can obtain two-dimensional points, each representing a cell. To reduce the number of individual dimensions, we aim to perform dimensionality reduction to obtain the most informative genes compressed into a smaller number of dimensions. As

a result, we are able to perform the downstream analysis with less computational effort. In this regard, we used some of the dimensionality reduction and manifold learning techniques such as LLE, Laplacian eigenmaps, and ICA on this dataset. Here, high-dimensional data is reduced to two and three dimensions. As a result, we obtain the most informative components, which are further used for clustering.

### Locally Linear Embedding

LLE succeeds in discovering the underlying structure of the manifold when used for dimensionality reduction. This technique is empowered by preserving "locality" of the data, when reduced to lower dimensions. In addition, LLE is capable of generating highly nonlinear embeddings. Consider the sample points in a high-dimensional space, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, where $\{\mathbf{x}_j, j \in N\}$ and the weight matrix is represented by $\mathbf{W} = \{w_{ij}\}$. First, a directed graph $\mathcal{G} = (\mathbf{X}, \mathbf{E}, \mathbf{W})$ is constructed, where the edges of the graph, $\mathbf{E} = \{e_{ij}\}$, represent the neighbourhood relations among sample points, $\mathbf{X}$, in the high-dimensional space. Next, the weights $\mathbf{W} = \{w_{ij}\}$ are assigned to the edges of the graph. The optimal weights $\mathbf{W} = \{w_{ij}\}$ are computed by solving the following constrained least-squared problem [131]:

$$\min \ \mathbf{x}_i - \sum_{k \in K_n} w_{kn} \mathbf{x}_k \quad \text{s.t.} \sum_{k \in K_n} w_{kn} = 1 \,. \tag{6.2}$$

In the second step, the weights are assigned to each edge of the graph, and each sample is considered as a small linear patch of the sub-manifold. Finally, the weights are computed locally and linearly in the data by reconstructing each input pattern from its $k$-nearest neighbours, where the reconstruction error, $\varepsilon_r$, is calculated in terms of the mean squared error (MSE) as follows:

$$\varepsilon_r = \sum_{i=1}^{n} \mathbf{x}_i - \sum_{k \in K_i} w_{ki} \mathbf{x}_k{}^2 \tag{6.3}$$

Modified LLE (MLLE), is an enhanced version of standard LLE and has been shown to be closely related to Local Tangent Space Alignment (LTSA) [112]. MLLE attempts to exploit the dense relations that exist in the embedding space.

### Other Dimensionality Reduction Methods

The Laplacian eigenmaps is a computationally effective approach to nonlinear dimensionality reduction that possesses locality-preserving properties and a natural connection to clustering [14]. Laplacian eigenmaps are similar to LLE. Given the input samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, the $k$ nearest neighbours are computed as the first step of the algorithm.

Typically, the weights are constant, such as $w_{ij} = 1/k$ or $w_{ij} = e^{-(\frac{||x_i - x_j||^2}{s})}$ where $s$ is the scalable parameter. Let $\mathbf{D} = \{d_{ij}\}$ be the diagonal matrix of elements $d_{ii} = \sum_{j=1}^{n} w_{ij}$. The final step is to minimize the reconstruction loss, $\varepsilon_r$, of the outputs, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n, \}$.

$$\varepsilon_r = \sum_{ij} \frac{w_{ij} \mathbf{y}_i - \mathbf{y}_j{}^2}{\sqrt{\mathbf{d}_{ii} \mathbf{d}_{jj}}} \tag{6.4}$$

With this function, nearby points are mapped to their nearest outputs by considering the weights $\mathbf{W}$. The minimum loss is computed from $m+1$ eigenvectors of the matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$ corresponding to the smallest eigenvalues of $\mathbf{L}$. The matrix $\mathbf{L}$ is a symmetrical, normalized form of the Laplacian, given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$. As in LLE, the eigenvectors corresponding to zero eigenvalues are discarded and the remaining $n$ vectors are included to obtain the outputs $\mathbf{y}_i$ in $\mathbb{R}^n$

ICA is a dimensionality reduction method used to analyze multivariate data [46]. ICA learns an efficient linear transformation of the data and attempts to find the underlying components and sources present in the data by its simple statistical properties assumptions. Unlike other methods, the underlying vectors of the transformation are assumed to be independent of each other, and it uses a non-Gaussian structure of the data, which is important to retrieve the underlying components of the transformed data as follows:

$$\begin{aligned} \mathbf{r} &= \mathbf{As} \\ \mathbf{Y} &= \mathbf{AX} \end{aligned} \tag{6.5}$$

where $\mathbf{r}$ and $\mathbf{s}$ are vectors and $\mathbf{A}$ is the matrix whose rows are orthogonal to each other. However, ICA assumes that the rows are linearly independent, and not necessarily orthogonal. As such, it leads to more informative components than PCA.

Moreover, ICA does not require to know the output of the system to break the data into some measurements. The transformed data can then be used for cluster analysis to find a group of genes with similar expression patterns.

### 6.2.4 Cell Clustering

Clustering is done via $k$-means, which is the most popular clustering technique. This algorithm progressively finds a pre-determined number of $k$ cluster centers by minimizing the sum of the squared Euclidean distances between each center and its closest neighbour. The clusters can be denoted as $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_k\}$. This work includes a methodology that cooperatively considers ICA and $k$-means for clustering the cells.

### 6.2.5 Cluster Annotation

To annotate the cell clusters we obtained, we first extracted the top 25 differentially expressed genes as markers in each cluster using the Wilcoxon rank sum test. Then, we found the corresponding cell types of each group of marker genes in each cluster. CellKb is a search tool that collects curated cell types manually from the literature. Its knowledge base includes 403 manually curated publications from over 7,000 studies published between 2013 and 2020 to extract 1,802 different cell types. Specific marker genes of cell types in CellKb wer extracted directly from gene signature from the Human Protein Altas and MSig-db.

### 6.2.6 Parameter Optimization

With the aim of preserving locality, the number of neighbours used to construct the neighbourhood graph is a very important parameter in manifold learning techniques. In this work, this parameter has been learned by running the algorithm several times on the data, in a range from 4 to 16, and found 11 is the best number nearest neighbours for our experiments. Also, we use the Euclidean distance metric as the weights of the edges. Another critical step in any clustering algorithm is determining the number of clusters, $k$. Validity indices help measure how good the clustering is. For our dataset, we ran the validity of indices and the Silhouette score for a range of 4 to 14 and found 7 as the optimal number of clusters for this data [90].

### 6.2.7 Performance Evaluation

Generally speaking, the best clustering is the one that maintains high intra-cluster distance and gives the most compact clusters. In this work, we use the Silhouette coefficient, which

is an evaluation metric that measures either the mean distance between a sample point and all other points in the same cluster or all other points in the next nearest neighbour cluster. Consider a set of clusters $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_k\}$, output by a clustering algorithm, k-means in our case. The Silhouette coefficient, $SH$, for the $i^{th}$ sample point in cluster $\mathbf{C}_j$, where $j = 1, ..., k$, can be defined as follows:

$$SH(\mathbf{x}_i) = \frac{b(\mathbf{x}_i) - a(\mathbf{x}_i)}{max(a(\mathbf{x}_i), b(\mathbf{x}_i))}, \tag{6.6}$$

where $a$ is the mean distance between point $\mathbf{x}_i$ and all other points inside the cluster (intra-cluster distance), and $b$ is the minimum mean value of the distance between a sample point $\mathbf{x}_i$ and the nearest neighbour cluster, and are calculated as:

$$\begin{aligned} a(\mathbf{x}_i) &= \frac{1}{|\mathbf{C}_k| - 1} \sum_{\mathbf{x}_j \in \mathbf{C}_k, i \neq j} d(\mathbf{x}_i, \mathbf{x}_j) \\ b(\mathbf{x}_i) &= \min_{k \neq i} \frac{1}{|\mathbf{C}_k|} \sum_{j=1}^{k} d(\mathbf{x}_i, \mathbf{x}_j). \end{aligned} \tag{6.7}$$

We also used Calinski-Harabasz (CH) and Davies-Bouldin (DB) validity of indices to assess the clustering performance. Calinski-Harabasz score [20], is a score used to evaluate the model where a higher score tells better-defined clusters. CH score is the ratio of the sum of between-clusters dispersion and of inter-cluster dispersion for all clusters that is as follows:

$$CH = \frac{tr(\mathbf{S}_B)}{tr(\mathbf{S}_W)} \times \frac{n-k}{k-1} \tag{6.8}$$

in which $n$ is size of input samples, $tr(\mathbf{S}_B)$ is the trace of the between-group dispersion

matrix and $tr(\mathbf{S}_W)$ is the within-cluster dispersion.

Davies-Bouldin index [25] is another validity index defined as the average of the similarity measure of each cluster. DB is computed as follows:

$$DB = \frac{1}{k}\sum_{i=1}^{k} max_{i\neq j} s_{ij}\,, \tag{6.9}$$

where $s_{ij}$ is the ratio between within-cluster distances and between cluster distances, and is calculated as $s_{ij} = \frac{w_i + w_j}{d_{ij}}$. The smaller DB value the better clustering, and as such, we aim to minimize Equation (6.9). Here, $d_{ij}$ is the Euclidean distance between cluster centroids $\mu_i$ and $\mu_j$, and $w_i$ is the within-cluster distance of cluster $\mathbf{C}_k$.

Overall, we used the Silhouette score to evaluate the clustering performance whereas CH and DB indices used to verify and find the optimal parameters, namely the best number of clusters.

## 6.3 Results and Discussion

### 6.3.1 Clustering Results

After applying manifold learning techniques on the data for dimensionality reduction, we performed $k$-means. The results are depicted in Table 6.1, where the clustering score ranges from 0 to 1. A score close to 1 represents good quality clustering, with 1 being the best, while a score near zero indicates that the clusters are not well defined. We observe that using MLLE the clusters are obtained with a score of 0.94 and that is the best clustering obtained from our experiments. As we can see in Fig.6.10, the cells are compactly bounded in their clusters and decent separation between the clusters. Also, two-dimensional ICA on three-dimensional MLLE data has been shown to provide the best visualization and clustering

score of 0.943 because the three-dimensional representation is carried to two-dimensional and the clusters are well characterized as shown in Fig. 6.12.

Table 6.1: Comparison of $k$-means clustering score using different dimensionality reduction techniques.

| **DR Technique** | **2D $k$-means** | **3D $k$-means** |
|---|---|---|
| Standard LLE | 0.623 | 0.683 |
| **Modified LLE** | **0.938** | **0.937** |
| Laplacian eigenmap | 0.700 | 0.782 |

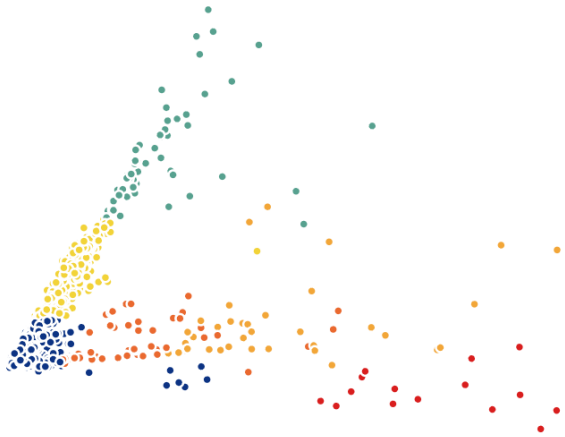

Figure 6.7: $k$-means applied on two-dimensional Laplacian eigenmaps; outliers have been removed to enhance visualization.

More precisely, two- and three-dimensional Laplacian eigenmaps, which are depicted in Fig. 6.7 and 6.9, show good cluster separation and enhanced visualization of the data, with clustering scores of 0.70, and 0.782, respectively. We can see in Fig. 6.7 that cells are more scattered between the clusters using two-dimensional Laplacian eigenmaps and it is hard to capture cells to form compact clusters, whereas three-dimensional Laplacian eigenmaps give better clustering result. Also, when we applied only ICA with $k$-means, we obtained below-average results compared to other techniques as shown in Fig. 6.8 with clustering score 0.357. This is because ICA is limited to linear transformations, whereas

manifold learning techniques consider data locality. As such, the latter can reveal complex relationships among the data points in higher-dimensional spaces. Therefore, we applied ICA on the dimensionally-reduced data because we observed interesting ”lines” or ”axes” in the three-dimensional data, and that led us to think that we could apply ICA to learn the linearly-independent, not necessarily orthogonal, components of the distribution of the data, and we witnessed slight improvement with clustering scores in MLLE and Standard LLE as it is displayed in Table 6.2. Applying ICA revealed some hidden, complex relationships among the cells in the clusters which are not noticeable in three dimensions. As such, we observed a significant improvement of the clustering score using Laplacian eigenmaps since there is more dispersion of the clustering of cells in Fig. 6.11. We also note more compact clusters than those of the two and three-dimensional clustering whose scores are depicted in Table 6.1.

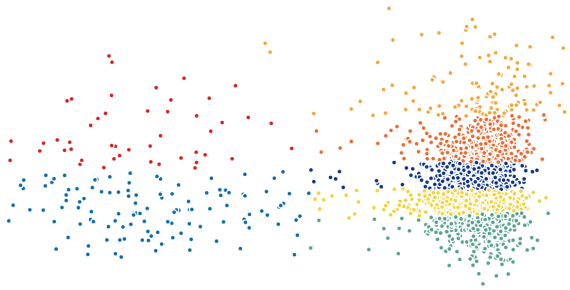

Figure 6.8: *k*-means applied on two-dimensional ICA.

Table 6.2: Results of manifold learning techniques followed by ICA and *k*-means clustering.

| **DR Technique** | **2D ICA-*k*-means** on 2D DR data | **2D ICA-*k*-means** on 3D DR data. |
|---|---|---|
| Standard LLE | 0.628 | 0.690 |
| Modified LLE | 0.930 | **0.943** |
| Laplacian eigenmap | 0.700 | 0.826 |

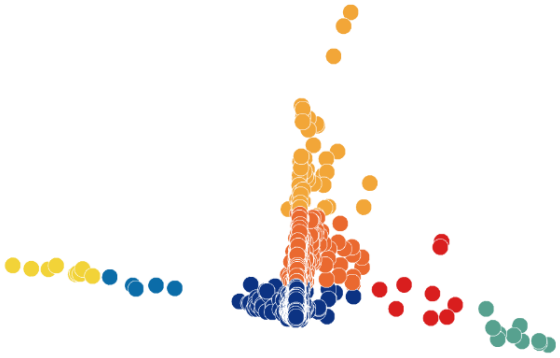

Figure 6.9: *k*-means clustering on three-dimensional Laplacian eigenmaps.

Table 6.3: Cell types identified by our proposed method.

| **Cell Type** |
|---|
| Proneural glioma stem-like cell |
| Th17/iTreg-stimulated CD4+ central memory T cell |
| Stem/Club/Hillock epithelial cell |
| Club cell |

### 6.3.2 Biological Assessment of the Results

The results obtained by CellKb [82] through finding overlapped genes in the literature are listed in Table 6.3. The results show several cell types involved in immune system pathways. It is well-known that one of the main SARS-CoV2 targets is the immune system function. For example, CD4+ T cells are found on the surface of immune cells and are key cells in response to the viral infection [21]. Also, the results show that Club cells that are found in the small airways of the lungs are involved in the TAP2 binding pathway at a molecular level. TAP2 is a gene that encodes the protein antigen peptide transporter 2. In immunology, the presence of antigens in the body normally triggers an immune response. Moreover, the epithelial cells show enzyme inhibitor activity in the molecular function re-

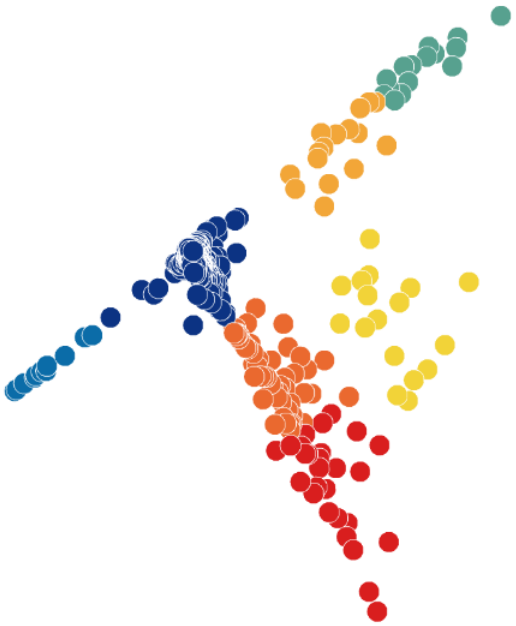

Figure 6.10: $k$-means clustering on three-dimensional MLLE.

sults. In addition, we obtained a list of overlapped marker genes that are involved in Herpes simplex virus 1 (HSV-1) infection and Influenza A pathway (Table 6.4). These results can be used for subsequent medical treatment or drug discovery through finding similar diseases in terms of functionality. Moreover, although numerous findings suggested potential links between HSV-1 and Alzheimer's disease (AD), a causal relation has not been demonstrated yet [26].

To summarize the results, performing ICA on transformed data after applying manifold learning techniques provides improved clustering results. Moreover, modified LLE combined with $k$-means leads to a more untangled view of the data and the corresponding clusters. Such non-linear dimensionality reduction methods have shown to be very powerful as they preserve the locality of the data from higher dimensions to lower dimensions. Evaluating the incidence of ICA as visualization and further reduction step shows even bet-

Figure 6.11: Two-dimensional ICA + $k$-means clustering is performed on three-dimensional Laplacian eigenmaps data; outliers have been removed to enhance visualization.

Figure 6.12: Two-Dimensional ICA + $k$-means clustering performed on three-dimensional MLLE data; outlliers have been removed to enhance visualization.

Table 6.4: Marker genes found in similar diseases.

| **Disease** | **Marker genes** |
|---|---|
| Influenza A | {RSAD2, IFIH1, MX1, STAT1}<br>{MX2, IRF7, TNFSF10, OAS1}<br>{DDX58,NFKBIA,OAS2}<br>{CXCL10,EIF2AK2,PML}<br>{ICAM1,CXCL8,OAS3,STAT2} |
| Herpes simplex virus 1 infection | {IFIH1,HLA-B,STAT1,IRF7}<br>{TAP1,OAS1,DDX58,NFKBIA}<br>{OAS2,STAT2,EIF2AK2,SP100}<br>{PML,HLA-E,B2M,OAS3,HLA-F} |

ter results and the best possible clustering scores. As such, this trend leads to a research avenue that involves a combination of enhanced nonlinear manifold learning techniques

such as MLLE, followed by linear methods such as ICA, which has shown to be more powerful than conventional, statistics-based methods such as PCA.

## 6.4 Conclusion and Future Work

This work focuses on the identification of SARS-CoV-2 target cell groups using manifold learning and clustering techniques on unlabeled data. The use of clustering validity and performance measures helps to find the best clusters that are the result of combining dimensionality reduction and clustering techniques. Identifying similarities that may be a result of structural, functional, or evolutionary relationships among the genes is the main goal of clustering the cells. In our proposed two-step clustering method, we have demonstrated that genes in our dataset that have similar expression patterns were grouped in highly-scored clusters in lung tissue cell data, achieving more than 90% accuracy. Efficient nonlinear dimensionality reduction and manifold learning techniques help improve the clustering results significantly and enhance visualization in a reduced space. There are some potential applications for investigating scRNA-seq data, even beyond COVID-19. As a further analysis in the future, we aim to identify biomarker genes that are differentially expressed among different clusters of cells. Using multiple datasets with batch effect correction can improve the results as well. As such, this can lead to enhance the accuracy of classification of the cells, as a supervised learning technique, using gene expression patterns of each sub-network. Using sub-networks, we can take advantage of avoiding employing a considerable number of uninformative genes to classify the underlying cells. Moreover, performing gene set enrichment analysis to annotate a set of highly-variable genes obtaining from each cluster can reveal biomarker genes that are involved in different gene ontology terms related to COVID-19. This work attempts to highlight the power of combining linear

methods such as ICA and manifold learning techniques such as MLLE for clustering to pave the way for further research in the future.

# Chapter 7

# SEGCECO: Subgraph Embedding of Gene expression matrix for prediction of CEll-cell COmmunication

## 7.1 Introduction

In the graph domain, link prediction is the problem of predicting the existence of a connection between two entities in a network. Given a network with various nodes connected to one another, we want to predict if two nodes are connected or are likely to connect in the future. With graph neural networks (GNN), we use not only network structural information, such as connections between nodes, but also individual node characteristics including the feature set of the node. Predicting friendship links among users in a social network, predicting co-authorship links in a citation network, and predicting interactions between genes and proteins in a biological network are some examples of link prediction.

On the other hand, cell-cell interactions regulate organism development by cell functions. A disease may occur when cells do not interact properly or decode molecular messages improperly. Thus, identifying and quantifying inter-cellular signaling pathways has become a common analysis carried out across a variety of fields [8].

With the rapid advancement of single-cell RNA sequencing technologies, researchers are becoming more interested in inferring cell-cell communication from single-cell (scRNA-seq) data. There are a variety of computational tools and resources including ProximID [16], CellChat [52], CellTalker [23], iTalk [115], SingleCellSignalR [17], CellPhoneDB [33], SpotSC [113], and scTensor [104], among others, which are available to predict cell-cell communication (CCC) using gene expression profile obtained from scRNA-seq data.

Generally, in scRNA-seq data analysis, cells are clustered based on their gene expression profiles, and cell types are determined and assigned to clusters based on the known marker genes. CCC tools mostly predict the inter-cellular communications, on the other hand, based on ligand-receptor interactions between pairs of clusters, i.e., cell types, in which one cluster is the source and the other is the target. The majority of the tools are made up of two main components: 1) a prior knowledge resource of intercellular interactions and 2) a method for estimating CCC based on known interactions and the present dataset. Each tool uses different methods, such as permutation of cluster labels, differential combinations, regularizations, and scaling, depending on the input datasets. These approaches result in a varied scoring system which makes it difficult to compare and evaluate the performance of CCC methods. Thus, selecting the appropriate tool to produce the best results is challenging [27]. A recent review study [8] discusses several existing tools

for measuring cell-cell communication.

In this work, to predict cell-cell communication, we resort to various approaches that have been successfully used for other existing link prediction problems, such as prediction of social connections between users in social networks [62]. Traditional approaches include heuristic methods such as common neighbors (CN) [80], Adamic Adar (AA) [2], and Resource Allocation (RA) [134]. Heuristic link prediction methods use network structure, i.e. network topology information, in the prediction process. Existing algorithms can be classified based on the maximum hop of neighbors required to calculate the score [129]. Common neighbors (CN), for example, are **first-order heuristics** that involve the target nodes' one-hop neighbors. Also, some supervised approaches are used for connection prediction, including support vector machine (SVM), baggings, and naives bayes, which are used to model the problem as a binary classification in which extraction of edge features is fundamental.

Moreover, recent methods are mostly built on top of node embedding methods (e.g., DeepWalk [84], node2vec [40], and structural deep network embedding [110]), with the edge representation constructed from the interaction between corresponding node embeddings.

We discovered that some methods perform well on certain types of networks. For instance, every heuristic technique is based on some assumptions and works based on the extracted pattern from the network topology, which is why there is no single heuristic method that works well for all types of networks. Thus, this is a significant drawback in heuristic

approaches. The same can be said about latent approaches, which achieve high accuracy in some types of networks but low accuracy in others. Thus, deciding on the best link prediction approach is usually a trial-and-error process.

On the other hand, Weisfeiler-Lehman Neural Machine (WLNM) [128] is considered as a state-of-the-art among link prediction methods based on its performance. It is a new approach based on the subgraph extraction around both target nodes $u$ and $v$. The local enclosing subgraph for a node pair ($u$, $v$) is the subgraph induced from the network by the union of $u$ and $v$'s neighbors up to $h$ hops. The hop is the maximum distance that node features can travel. This approach gives higher accuracy than heuristic and latent methods but requires additional computation time and memory.

In addition, SEAL (Learning from Subgraphs, Embeddings, and Attributes for Link Prediction) [129] is also a subgraphing method that addresses a number of weaknesses that WLNM has. To begin with, it enables learning not only from subgraph structures but also from latent and explicit node attributes, allowing it to incorporate a variety of information. Secondly, the fully-connected neural network in WLNM is replaced by a GNN that enables graph feature learning improvement. SEAL derived $\gamma$ decaying theory and proved that a small number of hops is enough to extract high-order heuristics and outperform WLNM. As a result, we choose SEAL as the baseline for predicting links between cells in our proposed framework, SEGCECO. It is a novel method that predicts cell-cell communication in scRNA-seq data via a gene expression attributed graph convolutional network. To our knowledge, this is the first time that graph-based methods are used for prediction of cell-cell communication prediction.

Also, to obtain more precise results, nodes in cell-cell communicating networks (CCN) represent the cells instead of groups of cells, i.e., cell types in our pipeline. Thus, the edges denote the connections (ligand-receptor interactions) between individual cells.

Our study aims to discover cell interactions, with nodes representing cells in the CCN and edges representing cell-cell interactions. Thus, we use similarity matrix-based optimization for scRNA-seq data analysis tool (SpotSC) [113] to perform such a task. Once the CCN network is constructed, our main goal is to predict links among the cells.

## 7.2 Materials and Methods

### 7.2.1 Datasets

The datasets used in this study are publicly available annotated scRNA-seq data from human and mouse pancreas tissue, drawn from the NCBI's Gene Expression Omnibus with the accession number GSE84133 [12]. The datasets were generated by following inDrop method under Illumina HiSeq 2500 to determine the transcriptomes of over 12,000 individual pancreatic cells from four human donors and two mice strains.

Table 7.1 depicts the details of datasets including tissue, the accession number, the number of cells, and the number of genes.

### 7.2.2 Proposed Method

Given complex, high-dimensional scRNA-seq data, we aim to predict cell-cell interactions by creating a pipeline that analyzes single-cell data and converts it to a graph format, per-

Table 7.1: Details of the datasets used in this work including tissue, the accession number, the number of cell types, the number of cells, and the number of genes.

| Dataset | Tissue | Accession # | # Cells | # Genes |
|---|---|---|---|---|
| Baron-human1 (BHuman1) | Human-Pancreas | GSM2230757 | 1,937 | 20,125 |
| Baron-human2 (BHuman2) | Human-Pancreas | GSM2230758 | 1,724 | 20,125 |
| Baron-human3 (BHuman3) | Human-Pancreas | GSM2230759 | 3,605 | 20,125 |
| Baron-human4 (BHuman4) | Human-Pancreas | GSM2230760 | 1,303 | 20,125 |
| Baron-Mouse1 (BMouse1) | Mouse-Pancreas | GSM2230761 | 822 | 14,878 |
| Baron-Mouse2 (BMouse2) | Mouse-Pancreas | GSM2230762 | 1,064 | 14,878 |

forming the prediction using GNNs. We consider the gene expression profile from scRNA-seq data and convert it to an undirected attributed graph, $G$, in which cells and cell-cell interactions are represented by nodes and edges respectively. More formally, given an undirected attributed graph $G$= $(V, E, \mathbf{X})$ where $V$ is a finite set of nodes (cells), $E$ is a finite set of edges (cell-cell interactions), in which $e_{ij} = (v_i, v_j) \in E$ and $x_{v_i}$ is the attribute vector associated with the node $v_i \in V$. Also, $A = (a_{ij})_{N\times N}$ represents the adjacency matrix of graph $G$, where $a_{ij} = 1$ if $e_{ij} \in E$ and $a_{ij} = 0$ otherwise, and $N$ is the number of nodes. We aim to predict the likelihood of a connection between $v_i$ and $v_j$.

Our proposed method consists of three main steps: 1) Preprocessing step (Figure S1.a), 2) Cell-cell network (CCN) construction (Figure S1.b), and 3) Applying the GCN (Figure S1.c). Before applying the GCN, the primary step is to preprocess the data (Section 7.2.2). Once the data is preprocessed, a CCN is constructed using SoptSC (Section 7.2.2) in Step 2 (Figure S1.b). The last module of the pipeline is using GCN for prediction. This is the main step of our framework which consists of four main phases before feeding a GCN: 1) Feature (gene) selection in the pooling layer, 2) Subgraph extraction, 3) Node information matrix construction, and 4) Deep Graph Convolutional Neural Network (DGCNN) learning. All these phases along with are explained in the next few sections.

**Step1: Data Preprocessing**

Prior to scRNA-seq data analysis, a critical step is to preprocess the data to reduce the effects of noise in the samples. To this end, we followed a standard preprocessing pipeline in scRNA-seq data analysis [71]. This step includes basic filtering, normalization, log transformation and scaling, as shown in the first step of the pipeline depicted in Figure S1.a. Low-quality cells would hamper downstream analysis. These cells may have been damaged or dead during the sequencing process, and are represented by the low number of expressed genes. Based on the pipeline [71], cells with less than 200 expressed genes, and genes expressed in less than three cells should be filtered out. For example, in BHuman1, we filtered out 5,387 low-expressed genes that are detected in less than three cells and kept 14,739 genes. We further investigated the distribution of the data, (Figure S3), as a data-specific quality-control step to filter low-quality cells. The number of genes expressed in the count matrix is typically between 500 and 4,000 genes, with a dense distribution of the number of expressed genes over the total count per cell for less than 4,000 genes. As such, we filtered out seven cells to remove low-quality ones. This step is performed to remove low-quality cells and poorly expressed genes.

Normalization is performed to balance the data by bringing it to a common scale without changing any values or losing any information. The top genes in the dataset are visualized before and after normalization in Figure S4.a and S4.b, respectively. The Counts Per Million (CPM) normalization method is used to normalize the data. Once normalization is performed, data matrices are $log(x+1)$ transformed. After per-gene quantification, we selected a subset of highly variable genes to use in downstream analyses as they are informative of the variability in the data. To achieve this, we chose a commonly used technique in [6] and defined the set of highly variable genes given a normalized dispersion higher than

0.5 after normalization, yielding 2,546 genes. For preprocessing. We used Scanpy [118], a specifically designed package to analyze scRNA-seq datasets.

**Step2: Cell-cell Network (CCN) Construction**

SoptSC (Similarity-matrix based optimization for single-cell data analysis) successfully performs multiple inference tasks such as unsupervised clustering, pseudo-temporal ordering, lineage inference, and marker gene identification based on a cell–cell similarity matrix. The cell-cell similarity matrix $S$ is learned from the original scRNA-seq data matrix, i.e., gene expression matrix $X$ of size $m \times n$ with $m$ genes and $n$ cells, using a low-rank representation model [135]. The element $S_{ij}$ (=$S_{ji}$) of similarity matrix $S$ measures the degree of similarity between cell $i$ and cell $j$ [113]. Also, a cell-cell communication graph $G$ is constructed using adjacency matrix $A$, which is derived from similarity matrix $S$, where $A_{ij} = 1$ if $S_{ij} > 0$, or $A_{ij} = 0$ otherwise. In this work, we constructed the cell-cell communication network using this method.

**Step 3:1: Gene Selection in Pooling Layer**

Downsampling is crucial in graph analysis, which is included in the pooling layer of our method framework. The pooling layer consists of selecting genes (with a threshold $\tau$) by Information Gain (IG) feature selection. This step provides the node attribute information (side information) of each individual node, i.e., explicit features. Information gain (IG), as a feature selection method, computes the reduction in entropy by splitting the dataset based on a given value of a random variable and measures how important or relevant the feature is. This is done by estimating the information gained from each variable and choosing the one with the maximum value. Based on Equation (7.1), the largest information gain is equal

to the smallest entropy. IG is calculated by subtracting the weighted entropy values from the original entropy values by the following Equation (7.2). In other words, IG measures how changes to the dataset affect the distribution of the classes or target variables.

$$H(X) = -\sum p(X)\log p(X)\,, \tag{7.1}$$

where for dataset $\mathbf{X} = \{x_i\}$, $H(X)$ is the probability of randomly picking an element of the class.

$$I(X,a) = H(X) - H(X|a)\,, \tag{7.2}$$

where *I(X,a)* represents the information gain in dataset $\mathbf{X} = \{x_i\}$ for variable $a$, $H(X)$ is the original entropy of $X$ and $H(X|a)$ is the conditional entropy for the given variable $a$.

**Step 3:2: Enclosing Subgraph Extraction**

The subgraph induced from the network by the union of $u$ and $v$'s neighbours up to $k-$ hops is called the enclosing subgraph for a node pair ($u$, $v$). The hop is the distance that node features can traverse in one hop. Enclosing subgraph extraction involves extracting the local enclosing subgraphs around the target nodes $u$ and $v$. The enclosing subgraph is extracted from the training data, which contains both positive (existent) and negative (non-existent) sets of sampled links, based on $h$-hop neighbours for the target nodes $u$ and $v$. Figure S5 depicts an example of the 1-hop enclosing subgraphs for target nodes $(A,B)$ and $(C,D)$.

**Step 3:3: Node Information Matrix Construction**

In the node information matrix, each row corresponds to the node's feature vector, which is represented as $X$. In the SEAL [129] approach, there are three components in the node information matrix:

**Structural Information: Node Labeling** Nodes are labelled based on their structural roles using a graph labelling method, the Double-Radius Node Labeling (DRNL) algorithm (explained in [129]). The main purpose of this step is to label every node in the enclosing subgraph in order to distinguish the target nodes between which a link should be expected. Later, in the DGCNN learning step, nodes will be sorted in a sort pooling layer, based on their structural roles, indicated by node labels. Labels are assigned to nodes in such a way that the target nodes $u$ and $v$ are labelled 1. Also, the radius of node $i$ with respect to target nodes, namely $(d(i,u),d(i,v)$, can be used to define its position. Thus, nodes on the same orbit are given the same label. In other words, larger labels are assigned to nodes that have a larger radius (farther nodes) with respect to target nodes. This algorithm can be better understood by following the diagram of Figure S6, which satisfies the following conditions:

1. if $d(i,x)+d(i,y) \neq d(j,x)+d(j,y)$, then $d(i,x)+d(i,y) < d(j,x)+d(j,y) \Leftrightarrow f_l(i) < f_l(j)$;

2. if $d(i,x)+d(i,y) = d(j,x)+d(j,y)$, then $d(i,x)d(i,y) < d(j,x)d(j,y) \Leftrightarrow f_l(i) < f_l(j)$.

where $f_l(i)$ is the label assigned to node $i$ and $(d(i,x),d(i,y))$ is the double radius.

**Latent Information: Node Embedding** The node embedding methods, i.e., Node2Vec [40], LINE [101] and Spectral Clustering (SC) (explained in Section 7.3.1), give the feature representation of nodes in a graph. Thus, an additional step is required to learn the features of the edges from node embedding in order to predict links as a binary classification problem.

**Negative injection** trick, as explained in [129], is used to help to generate the node embeddings. This consists of adding the negative (non-existent) set of sampled links, $E_n$, to the positive (existent) set of sampled links, $E$, and generating the embeddings on $G' = (V, E \cup E_n)$. The node embedding method used in our method is Node2vec [40].

**Explicit Information: Node Attributes** Both latent and explicit features of each node are included in the node information matrix $X$ for its corresponding row in $X$. Latent feature learning methods learn low-dimensional latent representation or embedding for each node using matrix factorization [127]. The adjacency or Laplacian matrix derived from the graph can be used for matrix factorization. On the other hand, the explicit features contain side information about the individual nodes, available in the form of Node Attributes. The SEAL method [129] shows significant improvement in performance when combining both latent and explicit features.

### Step 3:4: Learning Deep Graph Convolutional Neural Network

Deep Graph Convolutional Neural Network (DGCNN) [130] is a deep learning architecture which is designed to operate on graph data. DGCNN is divided into three main parts:

1) Graph Convolution layers: Localized graph convolutions are used to extract the hidden feature information of nodes from the graph. DGCNN consists of multiple convolutional layers in which the output of each convolutional layer is passed to a hyperbolic tangent ($tanh$) non-activation function. A GCN with four convolutional layers is shown in Figure S7.

2) SortPooling Layer: In the SortPooling layer, the unordered node attributes of the graph from the spatial graph convolutions layer are fed as the input. The main purpose of this layer is to sort the feature descriptors, each of which represents a node. Rather

than summing up these node features, the SortPooling layer arranges them in a consistent order and outputs a sorted graph representation with a given size. Then, it can be read and trained by standard convolutional neural networks. Nodes are sorted using graph labelling methods, based on their structural roles, in descending order using the last layer's output, $Z^h$. Once the nodes are sorted, the next step is to unify the sizes of the output tensor. The main intention behind it is to unify the graph sizes to $k$ by deleting the last $n-k$ rows if $n > k$, or adding $k-n$ zero rows otherwise. The output of the SortPooling Layer is shown in Figure S8.

3) Traditional Convolutional and Dense Layers: These layers are used to make a prediction based on the sorted graph representations generated by the SortPooling layer. The architecture of DGCNN is shown in Figure S9. Given the adjacency matrix $A \in \{1,0\}^{nXn}$ of graph $G$ with $n$ number of nodes and each node containing the $c$ dimensional feature vector as well as the node information matrix $X \in R^{nXc}$ of an enclosing subgraph with each row representing the node, DGCNN employs the following convolution layer:

$$Z = f(\tilde{D^{-}}1\tilde{A}XW), \tag{7.3}$$

where $\tilde{A} = A + I$, $I$ is the identity matrix, $\tilde{D}$ is the diagonal degree matrix with $\tilde{D_{i,i}} = \sum j\tilde{A_{i,j}}$, $W$ is a trainable graph convolutional parameters, $f$ is a non-linear activation function, and $Z \in R^{nXc'}$ is the output activation matrix.

The graph convolution incorporates each node's hidden representation by aggregating attribute information from its neighbours. The graph convolution can be split into four different steps:

1. Linear feature transformation is applied to the node information matrix $X$, by multiplying it by $W$.

2. Node information is propagated to neighbouring nodes as well as the node itself by $\tilde{A}XW$.

3. Each row is normalized by multiplying it by $D^{-1}$.

4. Non-linear activation function is applied to obtain the output.

## 7.3 Performance Evaluation

We conduct extensive experiments to evaluate the performance of our proposed method. We included seven methods for comparison: four state-of-the-art latent feature learning methods (Section 7.3.1) including Node2Vec [40], LINE [101], Deepwalk [84] and SC, as well as WLNM [128], GAE [114] and VGAE [54] (Section 7.3.2).

We use the binary operators (proposed in Node2Vec) in our evaluation process. These operators over the corresponding feature vectors of nodes $u$ and $v$, i.e., $f(u)$ and $f(v)$, are utilized to generate the edge/link embedding $g(u,v)$ for edge $e = (u,v)$.

- **Average:**

$$fx(u) \boxplus f(v) = \frac{f(u) + f(v)}{2}. \tag{7.4}$$

- **Hadamard:**

$$f(u) \boxdot f(v) = f(u) * f(v). \tag{7.5}$$

- **Weighted-L1:**

$$||f(u) \cdot f(v)||_{\overline{1}} = |f(u) - f(v)|. \tag{7.6}$$

- **Weighted-L2:**

$$||f(u) \cdot f(v)||_{\overline{2}} = |f(u) - f(v)|^2. \tag{7.7}$$

The original code for Node2vec, Deepwalk, and LINE is used in our comparison study. The node embeddings generated from these methods are used to generate the link embeddings (as explained in Section 7.3.1). Then, we used logistic regression as the classifier to predict the links. To evaluate the performance with the other methods including GAE, VGAE, WLNM, we used the default settings. We used different hyperparameters for each method as described in Table S2. To implement the core of our method, we used the base implementation of the SEAL method. Then, to evaluate the performance of the results, we used 90%-10% of data as training and testing sets respectively. To generate FPR/TPR distribution of the datasets, we have taken the mean of the corresponding values.

We used Area Under Curve (AUC), accuracy, precision, recall, F1-score and receiver operating characteristic curve (ROC curve) as evaluation metrics. To calculate the evaluation metrics, we used training and testing data which consists of both positive (existent) and negative (non-existent) links. As a negative set, we randomly chose an equal number of unconnected pairs of nodes from the network with no edge connection between them. We arbitrarily remove 10% of links as testing data and the remaining 90% are used as training data. The statistical information of the network extracted from datasets (discussed in Section 7.2.1) is shown in Table S1.

### 7.3.1 Latent Feature Methods

Given a network $G$ with a finite set of nodes (or vertices) $V$ and a finite set of edges $E$, latent features are the features in the low dimensional representations of nodes $V$ computed using matrix factorization [127]. The matrix can be the adjacency matrix or the Laplacian matrix derived from the network $G$. Node2vec [40], LINE [101], and DeepWalk [84] are examples of network/node embedding algorithms which we use as latent feature methods

to learn latent features. In [85], these network embedding methods were found to implicitly factorize some network matrix representation. These methods are summarized as follows:

1) Node2Vec: The Node2vec [40] model for latent feature learning is an application of the Word2vec paradigm. The latter is a framework for word embedding used to learn continuous feature representations of nodes in networks using the skip-gram model. The goal is to optimize a neighbourhood-preserving likelihood objective in order to learn these representations. As an extension of the skip-gram architecture of networks, Node2vec is an embedding approach that works on neighbour nodes and generates low-dimensional embedding by converting graph/network into numerical representations. A second-order random walk approach is used to generate the numerical representation of the nodes in the graph. The idea behind Node2Vec is to use flexible, biased random walks that can trade-off between local and global network views. This approach returns feature representations that maximize the likelihood of preserving network neighbourhoods of nodes in a $d$-dimensional feature space [40].

2) DeepWalk: DeepWalk [84] learns $d$- dimensional latent feature representations using local information obtained from uniform random walks. To capture network topology information, Deepwalk introduced an unsupervised strategy that learns features that capture the graph structure independently of the labels' distribution, rather than mixing the label space as part of the feature space [84].

3) LINE: LINE [101] is a network embedding model designed for embedding very large-scale information networks, which contain millions of nodes and billions of edges. This method generates low-level embeddings by preserving both the first-order and second-order proximity of nodes. Furthermore, this method incorporates a novel edge-sampling technique that improves the efficiency of the model [101].

4) Spectral Clustering (SC): Spectral Clustering, SC, is a matrix factorization [127] technique that performs an eigendecomposition of a graph $G$, more specifically, the normalized Laplacian matrix $L$, and takes top $k$ eigenvectors as the feature representation of nodes, i.e., node embedding vectors, $Z$. The edge score is calculated as the sigmoid function, $Z \times Z^T$.

### 7.3.2 Graph-based Methods

1) WLNM (Weisfeiler-Lehman Neural Machine): WLNM is a subgraph-based link prediction method that extracts the enclosing subgraphs around the target nodes to learn graph structure features for link prediction. The number of nodes in the subgraph is denoted by the user-defined integer $K$. The Palette-WL algorithm, a variant of WL that is fast and order-preserving, is used to label nodes. The enclosing subgraph is then represented as an adjacency matrix by WLNM. A fully-connected neural network is trained on these adjacency matrices, together with their labels, to learn the existence of links. WLNM has three steps: a) enclosing subgraph extraction, b) subgraph pattern encoding and c) neural network training.

2) SEAL (Learning from Subgraphs, Embeddings and Attributes for Link Prediction: SEAL framework for link prediction learns general graph structure features from local subgraphs rather than complete networks. The method takes as input the enclosing subgraphs around the links and returns the likelihood that the links exist. SEAL consists of three steps: a) enclosing subgraph extraction, b) node information matrix creation, and c) GNN learning. The default GNN used in SEAL is DGCNN (Deep Graph Convolutional Neural Network) [Section 7.2.2] [129].

Table 7.2: Comparison of our method with latent methods.

| Operator | Method | BHuman1 | BHuman2 | BHuman3 | BHuman4 | BMouse1 | BMouse2 |
|---|---|---|---|---|---|---|---|
| Average | Node2vec | 0.4999 | 0.5162 | 0.5035 | 0.5177 | 0.5049 | 0.5127 |
| | LINE | 0.5061 | 0.5034 | 0.5053 | 0.4968 | 0.5142 | 0.5093 |
| | DeepWalk | 0.5029 | 0.5052 | 0.5 | 0.5148 | 0.5099 | 0.513 |
| | SC | 0.4728 | 0.5464 | 0.5361 | 0.531 | 0.5043 | 0.5312 |
| Hadamard | Node2vec | 0.9748 | 0.9766 | 0.9833 | 0.9711 | 0.9564 | 0.9726 |
| | LINE | 0.7077 | 0.7908 | 0.5696 | 0.8279 | 0.8474 | 0.8494 |
| | DeepWalk | 0.956 | 0.9634 | 0.9514 | 0.9615 | 0.9558 | 0.9635 |
| | SC | 0.9392 | 0.9625 | 0.9501 | 0.9623 | 0.9589 | 0.9648 |
| Weighted L1 | Node2vec | 0.9887 | 0.9885 | 0.9917 | 0.9851 | 0.9798 | 0.9846 |
| | LINE | 0.7204 | 0.7474 | 0.5528 | 0.8421 | 0.894 | 0.8848 |
| | DeepWalk | 0.9867 | 0.9857 | 0.9859 | 0.982 | 0.9813 | 0.9812 |
| | SC | 0.9743 | 0.9757 | 0.9696 | 0.9716 | 0.969 | 0.9694 |
| Weighted L2 | Node2vec | 0.9896 | 0.9895 | 0.9919 | 0.9862 | 0.9802 | 0.9846 |
| | LINE | 0.7243 | 0.7474 | 0.5603 | 0.8487 | 0.8989 | 0.8865 |
| | DeepWalk | 0.9869 | 0.9866 | 0.9857 | 0.9825 | 0.9823 | 0.9822 |
| | SC | 0.9748 | 0.9752 | 0.9687 | 0.9736 | 0.9752 | 0.9729 |
| | SEGCECO | 0.9985 | 0.9980 | 0.9989 | 0.9982 | 0.9975 | 0.9972 |

## 7.4 Results and Discussion

Overall, compared to other methods, our method achieves improvement in performance in terms of AUC. Table 7.2 shows the performance (AUC) of our method and latent methods. For all the datasets, Node2vec outperformed all other approaches for three of the four operators. It means Node2vec excels in generating low-dimensional embeddings of nodes in networks and achieving a neighbourhood-preserving objective. Thus, we chose Node2vec as the node embedding method in our framework.

Table 7.3 depicts the performance (AUC) of our method with other GNN-based methods, such as GAE, VGAE, WLNM. Among them, our method performs best with approximately 99% AUC. We anticipate that the improved performance of our method is due to the

proposed pooling layer in the framework, which uses IG as the feature selection method to select the top $\tau$ attributes (i.e., genes) as explicit features in the node information matrix, $X$, resulting in better prediction.

Table 7.3: Comparison of our method with other methods.

| **Datset** | **GAE** | **VGAE** | **WLNM** | **Our method** |
|---|---|---|---|---|
| BHuman1 | 0.9835 | 0.9852 | 0.9832 | 0.9985 |
| BHuman2 | 0.9859 | 0.9805 | 0.9839 | 0.9980 |
| BHuman3 | 0.9876 | 0.9869 | 0.9889 | 0.9989 |
| BHuman4 | 0.9838 | 0.9764 | 0.9773 | 0.9982 |
| BMouse1 | 0.9841 | 0.9764 | 0.9673 | 0.9975 |
| BMouse2 | 0.9838 | 0.9829 | 0.9744 | 0.9972 |

Moreover, Figure S10 plots the ROC curve for DeepWalk, Node2vec, LINE, SC, GAE, VGAE, WLNM, and our method on the BHuman1 dataset. It is noticeable that our method surpasses other approaches since the curve is closer to the top-left corner, indicating better performance.

Robust inferences are essential to minimize false discoveries and help reduce the number of validations to perform, which is especially useful when experiments are expensive.

Here, positive means interacting cells and negative means non-interacting cells. Observing Figure S11 reveals that our method obtained the lowest FPR of 0.0135 among all the approaches, implying that there is a very lower probability that our method will predict non-interacting cells as interacting cells. This, in other words, means when the cells do not have interactions, the chances of inaccurate predictions, i.e., the cells interact, are minimal. Furthermore, our method performs best in predicting actual interactions, that is, when there exist interactions between cells, the method predicts the same. The same behaviour is detected in other datasets as well. The ROC curves, FPR and TPR distribution on other

Table 7.4: Performance metrics of our method for the datasets.

| Dataset | AUC | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| BHuman1 | 0.9985 | 0.9928 | 0.9872 | 0.9987 | 0.9929 |
| BHuman2 | 0.9980 | 0.9903 | 0.9915 | 0.9891 | 0.9903 |
| BHuman3 | 0.9989 | 0.9923 | 0.9925 | 0.9921 | 0.9923 |
| BHuman4 | 0.9982 | 0.9886 | 0.9862 | 0.9913 | 0.9887 |
| BMouse1 | 0.9975 | 0.9854 | 0.9800 | 0.9908 | 0.9854 |
| BMouse2 | 0.9972 | 0.9878 | 0.9800 | 0.9954 | 0.9876 |

datasets can be found in the Supplementary Material (Figure S1 - S10). Thus, it can be concluded that our method yields the best results for all datasets when it regards distinguishing between interacting and non-interacting cells and making predictions.

Moreover, Accuracy, Precision, Recall, and F1-score are the commonly used evaluation metrics to illustrate the performance of the model. Recall evaluates the model for correctly identifying cell-cell communication. Precision shows the percentage of predictions accurately made by the model. Table 7.4 shows the AUC, accuracy, precision, recall, and F1-score of link prediction using our method framework on different datasets. Our method shows a performance of around 99% for all measures, indicating that our model can accurately predict cell-cell interactions and discriminate interacting cells from non-interacting cells. Based on the findings of the above-mentioned comparison, we conclude that our method surpassed all other approaches with 99% AUC, accuracy, and other performance measures across all datasets.

## 7.5 Biological Assessment

To interpret the network inferred, information such as gene ontology or pathway and other metadata associated with the cells in the dataset gain insight into functional relationships between cells. For example, a set of genes involved in a particular pathway are highly co-

expressed across different cells in the network. This could indicate a functional relationship between those cells related to that particular pathway. There are several network analysis toolboxes and resources that can be used to validate the predicted cell-cell interactions. Also, these tools can identify highly connected nodes or subnetworks, which may represent key regulators or effectors of the functional relationships between cells.

GeneMANIA Cytoscape's plugin has been used in this assessment since it provides an interactive network visualization of the predicted functional relationships between the query genes (genes of interest) and other genes based on co-expression, co-localization, and other data sources.

Also, the ReactomeFIViz app helps to find pathways and network patterns related to diseases by accessing Reactome pathways and the Functional Interaction network. Functional Interaction (FI) network is a highly reliable, manually curated pathway-based protein functional interaction network covering over 60% human proteins, and allows the construction of an FI sub-network based on a set of genes.

We ran the ReactomeFIViz to find underlying sub-networks in the query list. The results represent and validate some examples of interactions that GeneMANIA found based on the input. The visualization of the networks is presented in the Supplementary file, figure S22. As shown in Fig. S22, there are four sub-network related to the BHuman1 dataset. Query genes which are indicated with purple, are grouped in four separate clusters. Genes involved in the first cluster were analyzed using Reactome and six out of seven identifiers in the sample were found, where 64 pathways were hit by at least one of them. We listed the important ones in Table S1. For example, in the "Other interleukin signalling" pathway, interleukins are low molecular weight proteins that bind to cell surface receptors and act in an autocrine and/or paracrine fashion. They were first identified as factors produced by

leukocytes but are now known to be produced by many other cells throughout the body. They have pleiotropic effects on cells which bind them, impacting processes such as tissue growth and repair, hematopoietic homeostasis, and multiple levels of the host defence against pathogens where they are an essential part of the immune system [3]. Also, in the ”Cytokine Signaling in Immune system” pathway, cytokines are small proteins that regulate and mediate immunity, inflammation, and hematopoiesis. They are secreted in response to immune stimuli, and usually act briefly, locally, at very low concentrations. Cytokines bind to specific membrane receptors, which then signal the cell via second messengers, to regulate cellular activity [92]. Additionally, we ran GeneMANIA on the input list of the top 20 genes, extracted from the pooling layer which indicates the most effective genes in the predicted interactions among cells. Figure S23 represents the key regulators or effectors of the functional relationships between cells in separate network types, such as co-expression, predicted, shared protein domain, and other factors.

## 7.6 Conclusion

In this paper, we propose a pipeline for performing cell-cell interaction prediction in scRNA-seq data using GCN. This article demonstrates how scRNA-seq data in the form of a gene expression matrix is transformed into a graph representation, i.e., a cell-cell communication network (CCN), in order to predict cell-cell interactions in scRNA-seq datasets.

Our proposed method works with undirected, attributed graphs created from the gene expression profiles of the individual cells. The architecture of our method includes a pooling layer that coarsens the graph attributes from the scRNA-seq data while preserving the global structure of the input graph. The pooling layer is followed by the enclosing subgraph extraction, node information matrix construction, and finally GCN that convolves

over the graph to encode the representation of both local and global attributes. The experimental results have shown that our proposed method outperforms previous state-of-the-art techniques. We evaluated our method using AUC, accuracy, precision, recall, and F1-score evaluation metrics. Findings show a performance of approximately 99% for all performance measures across all the datasets. We empirically proved that our method yields better results in terms of AUC relative to the previously proposed latent and subgraph-based methods. Thus, we conclude that our method outperforms other approaches in predicting cell-cell predictions and distinguishing interacting from non-interacting cells.

Our proposed method also opens up new research opportunities to work with networks in which there is a special structure such as heterogeneous CCN, networks with explicit domain features for nodes and edges, and directed or multi-modal graphs. In addition to the application of cell-cell link prediction, the proposed method could be applied to node classification, node clustering, graph partitioning, and graph classification. We would also foresee applying our method to domains such as disease-gene or drug-target associations, knowledge graph completion, and recommendation systems, among others.

# Chapter 8

# Conclusion and Future Work

## 8.1 Conclusion and Future Work

This comprehensive research addresses various aspects of scRNA-seq data analysis with a focus on cell type identification, classification, and interaction prediction. The following synthesized conclusion encapsulates the main contributions from each study:

Our work has delved into diverse methodologies for unravelling the complexity of scRNA-seq data, emphasizing the identification of cell types through manifold learning and clustering techniques. The proposed two-step representation learning approach, employing $k$-means clustering and Modified Locally Linear Embedding (MLLE), demonstrated enhanced clustering outputs and meaningful organization of cell clusters. Notably, MLLE outperformed UMAP in high-dimensional cytometry, showcasing its efficacy in preserving data locality and improving the visualization of clustering results.

Expanding the scope, we applied manifold learning and clustering techniques to unlabeled data to identify SARS-CoV-2 target cell groups. The proposed two-step clustering method demonstrated high accuracy, paving the way for potential applications beyond

COVID-19, including cancer research, developmental biology, neurobiology, biomarker discovery, and gene set enrichment analysis, among others.

In a parallel investigation, we explored classification models, specifically XGBoost and SVM, for cell type discovery using marker genes. Leveraging sparsity-aware feature selection in XGBoost, we observed a significant boost in predictive accuracy. This study not only validated the effectiveness of ensemble tree models but also emphasized the importance of considering sparsity patterns in scRNA-seq data for precise and meaningful results. However, due to the sparsity or imbalanced nature of scRNA-seq data, there is a possibility that machine learning methods may lead to biased predictions. Considering these limitations as a separate, preprocessing step, for other main problems in scRNA-seq analysis is crucial. Although we did perform a preprocessing step in all studies, it is crucial to consider the fact that we could be missing biological interactions because we filter out these types of cells such as neutrophils and eosinophils, which express less than our filtering threshold. It is important to capture, for example, immune interactions with pancreatic cells, or any cases where immune infiltration matters. Here, lowering the filtering threshold might be a non-feasible solution since it will include more non-viable cells and noise than we would ideally like. Moreover, the technologies themselves also have a major drawback. In almost all omics technologies, we can only see the system as a snapshot because in NGS or any other technique, we have to lyse the cells and produce DNA or RNA from them, and it is not possible to observe dynamic changes; except by examining the cells at different times, which is only possible in cases of cell culture, case by case. Even within identical cells, their expression levels vary based on factors such as environmental conditions, spatial and temporal considerations (such as proximity to the organ's surface or blood supply arteries), and interactions with immune cells. This process is not a simple on/off switch with a fixed

percentage chance of occurrence; rather, certain interactions may not occur consistently due to the influence of homeostasis. The cells engage in interactions for the necessary duration and cease when the optimal level is attained. Therefore, as mentioned above, while we follow a general standard best practice preprocessing pipeline in scRNA-seq data analysis, which includes basic filtering as a general approach, we do need a separate, preprocessing step.

Our exploration extended to the realm of deep learning, proposing a novel approach that combines a self-organizing map (SOM) and convolutional neural network (CNN) for simultaneous dimensionality reduction, feature selection, and classification. The synergy of SOM and CNN showcased remarkable accuracy in identifying cell types, offering a potential unsupervised clustering algorithm for large-scale datasets.

Additionally, we delved into the supervised identification of cell types, employing feature selection methods coupled with classification techniques on annotated datasets. Our findings underscored the power of feature selection in enhancing classification accuracy by identifying informative biomarker genes and disregarding uninformative ones.

However, using only RNA-seq data might not capture the full complexity of cellular heterogeneity or interactability, as it predominantly provides information about gene expression levels. This approach may overlook other critical factors influencing cell identity, such as epigenetic modifications, protein expression, or spatial context. Therefore, alongside the investigation of computational approaches, it is important to consider as much information as possible, which is involved in cellular heterogeneity.

In a novel contribution, we presented a pipeline for predicting cell-cell interactions in scRNA-seq data using Graph Convolutional Networks (GCN). Our proposed method outperformed previous state-of-the-art techniques, achieving approximately 99% accuracy

across various datasets. This work not only advances the understanding of cell-cell interactions but also opens avenues for broader applications, such as disease-gene associations and recommendation systems. While employing GCNN allows for the prediction of higher-order relationships beyond the first-order, it is crucial to interpret these relationships in a biological context and selectively report only those that align with meaningful biological insight revealed in the existing literature. Moreover, GNNs may not fully capture the spatial aspects of cell-cell interactions in tissues using only scRNA-seq data. Integrating spatial information into the graph structure is crucial for a more accurate representation of the physical proximity of cells, which is often essential in understanding their interactions. In addition, the performance of GNNs heavily depends on the quality of the graph representation. Constructing an accurate and biologically relevant graph from experimental or spatial transcriptomic data is a non-trivial task, and inaccuracies or lack of relevant information in the graph may impact the predictive performance of the GNN.

As a whole, this research provides a multifaceted approach to scRNA-seq data analysis, offering insights into cell type identification, classification, and cellular interaction prediction. The findings lay a foundation for future research, suggesting avenues for future works, and applying the proposed methodologies to various domains beyond single-cell sequencing.

To overcome the above-mentioned limitations and enhance the meaning and accuracy of the proposed approaches we are proposing some potential avenues for future research:

1. Integration of Multi-Omics Data:

   A promising avenue for future work involves the integration of multi-omics data. Although the model can be used to infer cell-cell interactions or cell type identification, the generated predictions are still hypotheses to be validated experimentally. Tran-

scriptomics may not fully represent a biologically accurate view of cell-cell communication, as mRNA and protein abundance may be uncoupled by post-transcriptional and post-translational processes. To improve the model, one possible approach is through multi-omics data integration. Borrowing information from other omics data can improve confidence in the results. Novel techniques and methods such as Mass Cytometry and spatial transcriptomics (ST), for example, allow researchers to analyze the transcriptome of tissues in a more real context, with the addition of information about proteins. Also, the limited scale of experiments that focus on the examination of ligand and receptor pairs within specific cell lines or tissues has been addressed by the emergence of advanced techniques in spatial transcriptomics (ST). This valuable information facilitates broader analyses of extracellular interactions on a larger scale. By incorporating additional layers of information, such as ST, DNA methylation, gene mutations, copy number variations or aberrations, chromatin accessibility, protein abundance, or spatial transcriptomics, researchers can obtain a more comprehensive and nuanced understanding of cellular heterogeneity or cellular interactions. A multi-omics approach allows for a more holistic characterization of cellular states and phenotypes, considering both genetic and epigenetic factors. Moreover, integrating diverse data types enables the construction of more robust models, providing a richer context for cell type identification and prediction of cell-cell interaction prediction. By fusing information from multiple "omics" layers, researchers can gain deeper insights into the regulatory mechanisms and functional dynamics underlying cell heterogeneity. This integrative approach could hold the potential to refine the accuracy of the proposed methods and enhance the biological interpretability of the results, paving the way for a more thorough exploration of

complex biological systems. Specifically, in the pipeline proposed for the prediction of cell-cell interactions, there is a possibility to include as many features as possible to the structure of the attributed graph as the input to the GCNN.

2. Leveraging Different Ways of Graph Construction

   Within the realm of predicting cell-cell interactions, there exist numerous avenues for constructing cellular networks. One potential approach is the utilization of self-organizing maps, particularly in incorporating spatial transcriptomic data, leveraging their spatial information to create a more contextually significant graph. This step is crucial as it serves as a pre-processing stage for the proposed method, and the ultimate outcome heavily relies on the quality and relevance of this constructed graph.

3. Large-Scale Experiments and Cross-Tissue Analysis:

   To further amplify the robustness and applicability of the proposed methodologies, it is recommended to undertake more extensive experiments involving a larger and more diverse set of samples obtained from various tissues. Scaling up the scope of the experiments by increasing the number of samples and incorporating a broader tissue may contribute to a more comprehensive understanding of the methodologies' performance across different biological contexts. In parallel, engaging in cross-tissue analysis becomes pivotal, as it allows for the exploration of both shared patterns and distinctive features in cell types and their interactions across diverse tissues. This comparative analysis can unveil commonalities, highlighting conserved cellular behaviors, as well as differences, offering insights into tissue-specific variations. By embracing large-scale experiments and cross-tissue analyses, the methodologies not only gain in scalability and generalizability but also stand better poised to capture the

intricacies of cell types and interactions in a more biologically relevant manner.

4. Handling Dynamic Nature of Cellular Interactions

   GNNs may struggle to adapt to dynamic changes in cellular interactions over time. The inherent static nature of many GNN architectures may not effectively capture the temporal dynamics involved in cell-cell interactions. Therefore, the need to explore other approaches is crucial to deal with dynamic interaction prediction problems; for example, developing GNN architectures explicitly designed to capture temporal dynamics in cellular interactions. This may involve the incorporation of recurrent neural networks (RNNs) or attention mechanisms to model sequential changes in interactions over time. In addition, exploring ensemble methods that combine predictions from multiple GNNs trained on different temporal snapshots could be another possible approach. This can help mitigate the challenges associated with capturing dynamic changes by leveraging complementary information from multiple models. Moreover, developing GNN models that dynamically evolve the underlying graph structure to accommodate changes in cell-cell interactions over time is another potential variant. This could involve mechanisms for edge addition, removal, or weight adjustments based on evolving biological conditions.

5. Model Optimization Techniques to Deal with Computational Resource Limitation

   Training large-scale models on extensive scRNA-seq datasets can be computationally intensive, requiring significant computational resources. This can be a limitation, particularly for research groups with limited access to high-performance computing resources. Exploring model optimization techniques, such as model pruning, quantization, and compression, to reduce the size of GNN models, for example, without

compromising predictive performance could be a possible approach. This can result in faster inference times and decreased resource requirements. Or, exploring transfer learning techniques where pre-trained models on related tasks or datasets are fine-tuned can be considered as well. This can leverage existing knowledge, requiring fewer computational resources for training. While large-scale experiments are crucial somehow to achieve a general and stable model, the processing time for small- or average-scale experiments remains acceptable in light of the costly nature of biological experiments. Ultimately, computational approaches enhance and expedite biological experiments irrespective of scale. They enable researchers to conduct experiments on a selective set of genes or cells rather than a broad range, thereby reducing time and cost.

6. Dealing with the Sparsity Nature of scRNA-seq data

   scRNA-seq data exhibit inherent sparsity and noise. Given the zero-inflated nature of this data, a necessary pre-processing step is essential to prepare it for input into the main model. The effectiveness of classification, clustering, dimensionality reduction, graph neural networks (GNNs), and several other methods is greatly contingent on the quality of their input data. Various imputation approaches can play a crucial role in enhancing the data quality and subsequent performance of these analytical methods.

7. Application to other Disease Contexts:

   In addition to COVID-19, we can apply the developed hybrid methodologies of cell type identification to other diseases such as immune system-related diseases and different types of cancers. In the realm of cellular interactions, predicting the effect of treatment on tumour-immune interaction in small-cell lung cancer could be a great

example of real applications. Since interactions between tumour cells and immune cells can have many effects including tumour-promoting, tumour-suppressive, and metastasis, it is crucial to study healthy and cancerous cell interactions. It may help to unravel the therapeutic targeting of tumour-immune interactions such as immune checkpoint inhibitors. This could lead to a complementary study of cellular heterogeneity in various diseases, uncover potential therapeutic targets, and new diagnostic and personalized treatments.

8. Exploration of Additional Biological Networks:

   Our proposed method in cell-cell interaction prediction also opens up new research opportunities to extend the application of GCN-based approaches to other biological networks in which there is a special structure like heterogeneous networks, for example, disease-gene networks,drug-target networks, or other biological network contexts. In addition to the application of cell-cell link prediction, the proposed method could be applied to node classification, node clustering, graph partitioning, and graph classification.

9. Validation and Reproducibility of the Result to Ensure Meaningful Results:

   Finally, we emphasize the importance of experimental validation, or biological validation to a larger extent, for the proposed methodologies. Typically, the validation of outcomes derived from the proposed computational methods involves a manual process, which is both time-intensive and lacks reproducibility. Implementing an automatic validation through open-access biological contexts ensures the reproducibility of the proposed methods.

10. Identifying Rare Cell Types Our proposed two-step approach to cell type identifica-

tion could help efficiently identify and categorize various cell types within a larger-scale biological sample. This method can particularly benefit the identification of rare cell types, which may be challenging to detect using traditional techniques. Furthermore, by enhancing the visualization of cell groups, it is possible to provide researchers with clearer and more informative representations of cellular populations within the samples. This improved visualization can assist in distinguishing between different cell types and their spatial relationships, thereby aiding in the accurate identification and characterization of rare cell populations. Overall, our approach not only streamlines the process of cell type identification but also enhances the precision and reliability of the results obtained on a larger scale.

# Bibliography

[1] Tamim Abdelaal, Lieke Michielsen, Davy Cats, Dylan Hoogduin, Hailiang Mei, Marcel JT Reinders, and Ahmed Mahfouz. A comparison of automatic cell identification methods for single-cell rna sequencing data. *Genome Biology*, 20(1):1–19, 2019.

[2] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[3] Mübeccel Akdis, Simone Burgler, Reto Crameri, Thomas Eiwegger, Hiroyuki Fujita, Enrique Gomez, Sven Klunker, Norbert Meyer, Liam O'Mahony, Oscar Palomares, et al. Interleukins, from 1 to 37, and interferon-γ: receptors, functions, and roles in diseases. *Journal of allergy and clinical immunology*, 127(3):701–721, 2011.

[4] Axel A Almet, Zixuan Cang, Suoqin Jin, and Qing Nie. The landscape of cell–cell communication through single-cell transcriptomics. *Current opinion in systems biology*, 26:12–23, 2021.

[5] Robert A Amezquita, Aaron TL Lun, Etienne Becht, Vince J Carey, Lindsay N Carpp, Ludwig Geistlinger, Federico Marini, Kevin Rue-Albrecht, Davide Risso,

Charlotte Soneson, et al. Orchestrating single-cell analysis with bioconductor. *Nature Methods*, 17(2):137–145, 2020.

[6] Robert A Amezquita, Aaron TL Lun, Etienne Becht, Vince J Carey, Lindsay N Carpp, Ludwig Geistlinger, Federico Marini, Kevin Rue-Albrecht, Davide Risso, Charlotte Soneson, et al. Orchestrating single-cell analysis with bioconductor. *Nature Methods*, 17(2):137–145, 2020.

[7] Dvir Aran, Agnieszka P Looney, Leqian Liu, Esther Wu, Valerie Fong, Austin Hsu, Suzanna Chak, Ram P Naikawadi, Paul J Wolters, Adam R Abate, et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nature immunology*, 20(2):163–172, 2019.

[8] Erick Armingol, Adam Officer, Olivier Harismendy, and Nathan E Lewis. Deciphering cell–cell interactions and communication from gene expression. *Nature Reviews Genetics*, 22(2):71–88, 2021.

[9] Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems*, 3(4):346–360, 2016.

[10] Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell Systems*, 3(4):346–360, 2016.

[11] Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems*, 3(4):346–360, 2016.

[12] Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell Systems*, 3(4):346–360, 2016.

[13] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[14] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[15] Chuang Bian, Xubin Wang, Yanchi Su, Yunhe Wang, Ka-chun Wong, and Xiangtao Li. scefsc: Accurate single-cell rna-seq data analysis via ensemble consensus clustering based on multiple feature selections. *Computational and structural biotechnology journal*, 20:2181–2197, 2022.

[16] Jean-Charles Boisset, Judith Vivié, Dominic Grün, Mauro J Muraro, Anna Lyubimova, and Alexander Van Oudenaarden. Mapping the physical network of cellular interactions. *Nature Methods*, 15(7):547–553, 2018.

[17] Simon Cabello-Aguilar, Mélissa Alame, Fabien Kon-Sun-Tack, Caroline Fau, Matthieu Lacroix, and Jacques Colinge. Singlecellsignalr: inference of intercellular

networks from single-cell transcriptomics. *Nucleic Acids Research*, 48(10):e55–e55, 2020.

[18] Batuhan Cakir, Martin Prete, Ni Huang, Stijn Van Dongen, Pinar Pir, and Vladimir Yu Kiselev. Comparison of visualization tools for single-cell rnaseq data. *NAR Genomics and Bioinformatics*, 2(3):lqaa052, 2020.

[19] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[20] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[21] Eddie Cano-Gamez, Blagoje Soskic, Theodoros I Roumeliotis, Ernest So, Deborah J Smyth, Marta Baldrighi, David Willé, Nikolina Nakic, Jorge Esparza-Gordillo, Christopher GC Larminie, et al. Single-cell transcriptomics identifies an effectorness gradient shaping the response of cd4+ t cells to cytokines. *Nature communications*, 11(1):1–15, 2020.

[22] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[23] Anthony R Cillo, Cornelius HL Kürten, Tracy Tabib, Zengbiao Qi, Sayali Onkar, Ting Wang, Angen Liu, Umamaheswar Duvvuri, Seungwon Kim, Ryan J Soose, et al. Immune landscape of viral-and carcinogen-driven head and neck cancer. *Immunity*, 52(1):183–199, 2020.

[24] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[25] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

[26] Giovanna De Chiara, Roberto Piacentini, Marco Fabiani, Alessia Mastrodonato, Maria Elena Marcocci, Dolores Limongi, Giorgia Napoletani, Virginia Protto, Paolo Coluccio, Ignacio Celestino, et al. Recurrent herpes simplex virus-1 infection induces hallmarks of neurodegeneration and cognitive deficits in mice. *PLoS pathogens*, 15(3):e1007617, 2019.

[27] Daniel Dimitrov, Dénes Türei, Charlotte Boys, James S Nagai, Ricardo O Ramirez Flores, Hyojin Kim, Bence Szalai, Ivan G Costa, Aurélien Dugourd, Alberto Valdeolivas, et al. Comparison of resources and methods to infer cell-cell communication from single-cell rna data. *BioRxiv*, 2021.

[28] Daniel Dimitrov, Dénes Türei, Martin Garrido-Rodriguez, Paul L Burmedi, James S Nagai, Charlotte Boys, Ricardo O Ramirez Flores, Hyojin Kim, Bence Szalai, Ivan G Costa, et al. Comparison of methods and resources for cell-cell communication inference from single-cell rna-seq data. *Nature communications*, 13(1):3224, 2022.

[29] Chuan Dong, Yan-Ting Jin, Hong-Li Hua, Qing-Feng Wen, Sen Luo, Wen-Xin Zheng, and Feng-Biao Guo. Comprehensive review of the identification of essential genes using computational methods: focusing on feature implementation and assessment. *Briefings in bioinformatics*, 21(1):171–181, 2020.

[30] Chuan Dong, Yan-Ting Jin, Hong-Li Hua, Qing-Feng Wen, Sen Luo, Wen-Xin Zheng, and Feng-Biao Guo. Comprehensive Review of the Identification of Essential Genes Using Computational Methods: Focusing on Feature Implementation and Assessment. *Briefings in Bioinformatics*, 21(1):171–181, 2020.

[31] Angelo Duò, Mark D Robinson, and Charlotte Soneson. A systematic performance evaluation of clustering methods for single-cell rna-seq data. *F1000Research*, 7, 2018.

[32] Ron Edgar, Michael Domrachev, and Alex E Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002.

[33] Mirjana Efremova, Miquel Vento-Tormo, Sarah A Teichmann, and Roser Vento-Tormo. Cellphonedb: inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes. *Nature Protocols*, 15(4):1484–1506, 2020.

[34] Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete. *Physical review*, 47(10):777, 1935.

[35] Nazia Fatima and Luis Rueda. isom-gsn: an integrative approach for transforming multi-omic data into gene similarity networks via self-organizing maps. *Bioinformatics*, 36(15):4248–4254, 2020.

[36] Chao Feng, Shufen Liu, Hao Zhang, Renchu Guan, Dan Li, Fengfeng Zhou, Yanchun Liang, and Xiaoyue Feng. Dimension reduction and clustering models for single-

cell rna sequencing data: A comparative study. *International journal of molecular sciences*, 21(6):2181, 2020.

[37] 10X Genomics. Single cell gene expression dataset by cell ranger 1.1.0, May 2016.

[38] Ali Ghodsi. Dimensionality reduction a short tutorial. *Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada*, 37(38):2006, 2006.

[39] Ali Ghodsi. Dimensionality reduction a short tutorial. *Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada*, 37(38):2006, 2006.

[40] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[41] Dominic Grün, Anna Lyubimova, Lennart Kester, Kay Wiebrands, Onur Basak, Nobuo Sasaki, Hans Clevers, and Alexander Van Oudenaarden. Single-cell messenger rna sequencing reveals rare intestinal cell types. *Nature*, 525(7568):251–255, 2015.

[42] Manuel Guerrero, Francisco G Montoya, Raúl Baños, Alfredo Alcayde, and Consolación Gil. Adaptive community detection in complex networks using genetic algorithms. *Neurocomputing*, 266:101–113, 2017.

[43] Manuel Guerrero, Francisco G Montoya, Raúl Baños, Alfredo Alcayde, and Consolación Gil. Adaptive community detection in complex networks using genetic algorithms. *Neurocomputing*, 266:101–113, 2017.

[44] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell rna sequencing technologies and bioinformatics pipelines. *Experimental & molecular medicine*, 50(8):1–14, 2018.

[45] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell rna sequencing technologies and bioinformatics pipelines. *Experimental & molecular medicine*, 50(8):1–14, 2018.

[46] Aapo Hyvärinen. Independent component analysis: recent advances. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110534, 2013.

[47] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[48] Tomislav Ilicic, Jong Kyoung Kim, Aleksandra A Kolodziejczyk, Frederik Otzen Bagger, Davis James McCarthy, John C Marioni, and Sarah A Teichmann. Classification of low quality cells from single-cell rna-seq data. *Genome biology*, 17(1):1–15, 2016.

[49] Yusuke Imoto, Tomonori Nakamura, Emerson G Escolar, Michio Yoshiwaki, Yoji Kojima, Yukihiro Yabuta, Yoshitaka Katou, Takuya Yamamoto, Yasuaki Hiraoka, and Mitinori Saitou. Resolution of the curse of dimensionality in single-cell rna sequencing data analysis. *Life Science Alliance*, 2022.

[50] Saiful Islam, Amit Zeisel, Simon Joost, Gioele La Manno, Pawel Zajac, Maria Kasper, Peter Lönnerberg, and Sten Linnarsson. Quantitative single-cell rna-seq with unique molecular identifiers. *Nature methods*, 11(2):163, 2014.

[51] Zhicheng Ji and Hongkai Ji. Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. *Nucleic Acids Research*, 44(13):e117–e117, 2016.

[52] Suoqin Jin, Christian F Guerrero-Juarez, Lihua Zhang, Ivan Chang, Raul Ramos, Chen-Hsiang Kuan, Peggy Myung, Maksim V Plikus, and Qing Nie. Inference and analysis of cell-cell communication using cellchat. *Nature communications*, 12(1):1–20, 2021.

[53] Rie Johnson and Tong Zhang. Learning nonlinear functions using regularized greedy forest. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):942–954, 2013.

[54] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[55] Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

[56] Vladimir Yu Kiselev, Kristina Kirschner, Michael T Schaub, Tallulah Andrews, Andrew Yiu, Tamir Chandra, Kedar N Natarajan, Wolf Reik, Mauricio Barahona, Anthony R Green, et al. Sc3: consensus clustering of single-cell rna-seq data. *Nature methods*, 14(5):483–486, 2017.

[57] Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. scmap: projection of single-cell rna-seq data across data sets. *Nature methods*, 15(5):359–362, 2018.

[58] Dmytro Lande, Minglei Fu, Wen Guo, Iryna Balagura, Ivan Gorbov, and Hongbo Yang. Link prediction of scientific collaboration networks based on information retrieval. *World Wide Web*, 23:2239–2257, 2020.

[59] Ping Li. An empirical evaluation of four algorithms for multi-class classification: Mart, abc-mart, robust logitboost, and abc-logitboost. *arXiv preprint arXiv:1001.1020*, 2010.

[60] Runze Li and Xuerui Yang. De novo reconstruction of cell interaction landscapes from single-cell spatial transcriptome data with deeplinc. *Genome biology*, 23(1):1–24, 2022.

[61] Yunjin Li, Qiyue Xu, Duojiao Wu, and Geng Chen. Exploring additional valuable information from single-cell rna-seq data. *Frontiers in cell and developmental biology*, 8:593007, 2020.

[62] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[63] Arthur Liberzon, Aravind Subramanian, Reid Pinchback, Helga Thorvaldsdóttir, Pablo Tamayo, and Jill P Mesirov. Molecular signatures database (msigdb) 3.0. *Bioinformatics*, 27(12):1739–1740, 2011.

[64] Yuval Lieberman, Lior Rokach, and Tal Shay. Castle–classification of single cells by transfer learning: harnessing the power of publicly available single cell rna sequencing experiments to annotate new experiments. *PloS one*, 13(10):e0205499, 2018.

[65] Peijie Lin, Michael Troup, and Joshua WK Ho. Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome Biology*, 18(1):59, 2017.

[66] Zhaoyang Liu, Dongqing Sun, and Chenfei Wang. Evaluation of cell-cell interaction methods by integrating single-cell rna sequencing data with spatial information. *Genome Biology*, 23(1):1–38, 2022.

[67] Malte D Luecken and Fabian J Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular systems biology*, 15(6):e8746, 2019.

[68] Malte D Luecken and Fabian J Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):e8746, 2019.

[69] Malte D Luecken and Fabian J Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular systems biology*, 15(6):e8746, 2019.

[70] Malte D Luecken and Fabian J Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular systems biology*, 15(6):e8746, 2019.

[71] Malte D Luecken and Fabian J Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):e8746, 2019.

[72] Feiyang Ma and Matteo Pellegrini. Actinn: automated identification of cell types in single cell rna sequencing. *Bioinformatics*, 36(2):533–538, 2020.

[73] Wenjing Ma, Kenong Su, and Hao Wu. Evaluation of some aspects in supervised cell type identification for single-cell rna-seq: classifier, feature selection, and reference construction. *Genome biology*, 22:1–23, 2021.

[74] Wenjing Ma, Kenong Su, and Hao Wu. Evaluation of some aspects in supervised cell type identification for single-cell rna-seq: classifier, feature selection, and reference construction. *Genome Biology*, 22, 2021.

[75] Yeganeh Madadi, Aboozar Monavarfeshani, Hao Chen, W Daniel Stamer, Robert W Williams, and Siamak Yousefi. Artificial intelligence models for cell type and subtype identification based on single-cell rna sequencing data in vision science. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2023.

[76] Guo Mao, Zhengbin Pang, Ke Zuo, Qinglin Wang, Xiangdong Pei, Xinhai Chen, and Jie Liu. Predicting gene regulatory links from single-cell rna-seq data using graph neural networks. *Briefings in Bioinformatics*, 24(6):bbad414, 2023.

[77] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[78] Mauro J Muraro, Gitanjali Dharmadhikari, Dominic Grün, Nathalie Groen, Tim Dielen, Erik Jansen, Leon Van Gurp, Marten A Engelse, Francoise Carlotti, Eelco Jp De Koning, et al. A single-cell transcriptome atlas of the human pancreas. *Cell systems*, 3(4):385–394, 2016.

[79] Michael E Nelson, Simone G Riva, and Ana Cvejic. Smash: a scalable, general marker gene identification framework for single-cell rna-sequencing. *BMC bioinformatics*, 23(1):1–16, 2022.

[80] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001.

[81] Vy Nguyen and Johannes Griss. scannotatr: framework to accurately classify cell types in single-cell rna-sequencing data. *BMC bioinformatics*, 23(1):1–13, 2022.

[82] Ajay Patil and Ashwini Patil. Cellkb immune: a manually curated database of hematopoietic marker gene sets from 7 species for rapid cell type identification. *biorxiv*, pages 2020–12, 2020.

[83] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[84] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[85] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 459–467, 2018.

[86] Xiaojie Qiu, Andrew Hill, Jonathan Packer, Dejun Lin, Yi-An Ma, and Cole Trapnell. Single-cell mrna quantification and differential analysis with census. *Nature methods*, 14(3):309–315, 2017.

[87] Xiaojie Qiu, Andrew Hill, Jonathan Packer, Dejun Lin, Yi-An Ma, and Cole Trapnell. Single-cell mrna quantification and differential analysis with census. *Nature methods*, 14(3):309–315, 2017.

[88] Muhammad Summair Raza and Usman Qamar. *Understanding and using rough set based feature selection: concepts, techniques and applications*. Springer, 2017.

[89] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[90] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[91] Rickard Sandberg. Entering the era of single-cell transcriptomics in biology and medicine. *Nature methods*, 11(1):22–24, 2014.

[92] Pere Santamaria et al. Cytokines and chemokines in autoimmune disease: an overview. *Advances in experimental medicine and biology*, 520:1–7, 2003.

[93] Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33(5):495–502, 2015.

[94] Lawrence K Saul and Sam T Roweis. An introduction to locally linear embedding. *unpublished. Available at: http://www. cs. toronto. edu/˜ roweis/lle/publications. html*, 2000.

[95] Åsa Segerstolpe, Athanasia Palasantza, Pernilla Eliasson, Eva-Marie Andersson, Anne-Christine Andréasson, Xiaoyan Sun, Simone Picelli, Alan Sabirsh, Maryam Clausen, Magnus K Bjursell, et al. Single-cell transcriptome profiling of human

pancreatic islets in health and type 2 diabetes. *Cell metabolism*, 24(4):593–607, 2016.

[96] Anne Senabouth, Samuel W Lukowski, Jose Alquicira Hernandez, Stacey B Andersen, Xin Mei, Quan H Nguyen, and Joseph E Powell. ascend: R package for analysis of single-cell rna-seq data. *GigaScience*, 8(8):giz087, 2019.

[97] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.

[98] Min Su, Tao Pan, Qiu-Zhen Chen, Wei-Wei Zhou, Yi Gong, Gang Xu, Huan-Yu Yan, Si Li, Qiao-Zhen Shi, Ya Zhang, Xiao He, Chun-Jie Jiang, Shi-Cai Fan, Xia Li, Murray J. Cairns, Xi Wang, and Yong-Sheng Li. Data analysis guidelines for single-cell rna-seq in biomedical studies and clinical applications. *Military Medical Research*, 2022.

[99] Aravind Subramanian, Heidi Kuehn, Joshua Gould, Pablo Tamayo, and Jill P. Mesirov. GSEA-P: a desktop application for Gene Set Enrichment Analysis. *Bioinformatics*.

[100] Xiaoxiao Sun, Yiwen Liu, and Lingling An. Ensemble dimensionality reduction and feature gene extraction for single-cell rna-seq data. *Nature communications*, 11(1):1–9, 2020.

[101] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

[102] Tatiana Tatusova, Michael DiCuccio, Azat Badretdin, Vyacheslav Chetvernin, Eric P Nawrocki, Leonid Zaslavsky, Alexandre Lomsadze, Kim D Pruitt, Mark Borodovsky, and James Ostell. Ncbi prokaryotic genome annotation pipeline. *Nucleic acids research*, 44(14):6614–6624, 2016.

[103] Tatiana Tatusova, Michael DiCuccio, Azat Badretdin, Vyacheslav Chetvernin, Eric P Nawrocki, Leonid Zaslavsky, Alexandre Lomsadze, Kim D Pruitt, Mark Borodovsky, and James Ostell. Ncbi prokaryotic genome annotation pipeline. *Nucleic acids research*, 44(14):6614–6624, 2016.

[104] Koki Tsuyuzaki, Manabu Ishii, and Itoshi Nikaido. Uncovering hypergraphs of cell-cell interaction from single cell rna-sequencing data. *BioRxiv*, page 566182, 2019.

[105] Koen Van den Berge, Katharina M Hembach, Charlotte Soneson, Simone Tiberi, Lieven Clement, Michael I Love, Rob Patro, and Mark D Robinson. Rna sequencing data: hitchhiker's guide to expression analysis. *Annual Review of Biomedical Data Science*, 2:139–173, 2019.

[106] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[107] Edwin Vans, Ashwini Patil, and Alok Sharma. Feats: feature selection-based clustering of single-cell rna-seq data. *Briefings in Bioinformatics*, 22(4):bbaa306, 2021.

[108] Akram Vasighizaker, Saiteja Danda, and Luis Rueda. Discovering cell types using manifold learning and enhanced visualization of single-cell rna-seq data. *Scientific Reports*, 12(1):1–16, 2022.

[109] Akram Vasighizaker, Sheena Hora, Yash Trivedi, and Luis Rueda. Comparative analysis of supervised cell type detection in single-cell rna-seq data. In *International Work-Conference on Bioinformatics and Biomedical Engineering*, pages 333–345. Springer, 2022.

[110] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234, 2016.

[111] HaiYun Wang, JianPing Zhao, ChunHou Zheng, and YanSen Su. scdssc: Deep sparse subspace clustering for scrna-seq data. *PLOS Computational Biology*, 2023.

[112] Jianzhong Wang. Laplacian eigenmaps. In *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*, pages 235–247. Springer, 2012.

[113] Shuxiong Wang, Matthew Karikomi, Adam L MacLean, and Qing Nie. Cell lineage and communication network inference via optimization for single-cell transcriptomics. *Nucleic Acids Research*, 47(11):e66–e66, 2019.

[114] Yingfeng Wang, Biyun Xu, Myungjae Kwak, and Xiaoqin Zeng. A simple training strategy for graph autoencoder. In *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*, pages 341–345, 2020.

[115] Yuanxin Wang, Ruiping Wang, Shaojun Zhang, Shumei Song, Changying Jiang, Guangchun Han, Michael Wang, Jaffer Ajani, Andy Futreal, and Linghua Wang.

italk: an r package to characterize and illustrate intercellular communication. *BioRxiv*, page 507871, 2019.

[116] Yue J Wang, Jonathan Schug, Kyoung-Jae Won, Chengyang Liu, Ali Naji, Dana Avrahami, Maria L Golson, and Klaus H Kaestner. Single-cell transcriptomics of the human endocrine pancreas. *Diabetes*, 65(10):3028–3038, 2016.

[117] Ziwei Wang, Hui Ding, and Quan Zou. Identifying cell types to interpret scrna-seq data: how, why and more possibilities. *Briefings in Functional Genomics*, 19:286–291, 2020.

[118] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):1–5, 2018.

[119] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

[120] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[121] Emanuel Wyler, Kirstin Mösbauer, Vedran Franke, Asija Diag, Lina Theresa Gottula, Roberto Arsie, Filippos Klironomos, David Koppstein, Salah Ayoub, Christopher Buccitelli, et al. Bulk and single-cell gene expression profiling of sars-cov-2 infected human cell lines identifies molecular targets for therapeutic intervention. *BioRxiv*, 2020.

[122] Yurong Xin, Jinrang Kim, Haruka Okamoto, Min Ni, Yi Wei, Christina Adler, Andrew J Murphy, George D Yancopoulos, Calvin Lin, and Jesper Gromada. Rna se-

quencing of single human islet cells reveals type 2 diabetes genes. *Cell metabolism*, 24(4):608–615, 2016.

[123] Yuchen Yang, Ruth Huh, Houston W Culpepper, Yuan Lin, Michael I Love, and Yun Li. Safe-clustering: Single-cell aggregated (from ensemble) clustering for single-cell rna-seq data. *Bioinformatics*, 35(8):1269–1277, 2019.

[124] Christopher Yau et al. pcareduce: hierarchical clustering of single cell transcriptional profiles. *BMC bioinformatics*, 17(1):1–11, 2016.

[125] Amit Zeisel, Ana B Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, et al. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347(6226):1138–1142, 2015.

[126] Amit Zeisel, Ana B Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz, et al. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347(6226):1138–1142, 2015.

[127] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE Transactions on Big Data*, 6(1):3–28, 2018.

[128] Muhan Zhang and Yixin Chen. Weisfeiler-lehman neural machine for link prediction. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 575–583, 2017.

[129] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[130] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[131] Zhenyue Zhang and Jing Wang. Mlle: Modified locally linear embedding using multiple weights. *Advances in neural information processing systems*, 19, 2006.

[132] Zhenyue Zhang and Jing Wang. Mlle: Modified locally linear embedding using multiple weights. In *Advances in neural information processing systems*, pages 1593–1600, 2007.

[133] Bo Zhou and Wenfei Jin. Visualization of single cell rna-seq data using t-sne in r. *Stem Cell Transcriptional Networks: Methods and Protocols*, pages 159–167, 2020.

[134] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71(4):623–630, 2009.

[135] Liansheng Zhuang, Jingjing Wang, Zhouchen Lin, Allen Y Yang, Yi Ma, and Nenghai Yu. Locality-preserving low-rank representation for graph construction from nonlinear manifolds. *Neurocomputing*, 175:715–722, 2016.

[136] Justina Žurauskienė and Christopher Yau. pcareduce: hierarchical clustering of single cell transcriptional profiles. *BMC bioinformatics*, 17:1–11, 2016.

# Vita Auctoris

Akram Vasighizaker received her Master's in Computer Engineering from Tarbiat Modares University, Tehran, Iran in 2015. Her research interests include Data Science, Machine Learning, Deep Learning, Computational Biology and Bioinformatics, and Data Representation and Visualization.