Correspondence analysis based biclustering and joint visualization of cells and genes for single cell transcriptomic data

Dissertation

zur Erlangung des Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

am Fachbereich Mathematik und Informatik

der Freien Universität Berlin

vorgelegt von

Yan Zhao

Berlin, 2024

The Book of Nature is written in the language of mathematics.

— Galileo Galilei

Erstgutachter: **Prof. Dr. Martin Vingron**

Zweitgutachter: **Prof. Dr. Tim Beißbarth**

Tag der Disputation: 26.07.2024

Acknowledgements

Starting my PhD journey just four months before the onset of the SARS-COV-2 pandemic brought about numerous challenges and opportunities. The past four years have been a long and unique voyage, during which I experienced several significant milestones and first-time experiences. These included staying home for extended periods without leaving the house, publishing my first first-author paper, visiting Europe for the first time, developing an R package , and more. Although the journey had its moments of hardship, it was truly exhilarating and enriching. I would like to express my heartfelt appreciation to all those who have supported me in both my personal life and the scientific endeavors presented in this work.

First and foremost, I extend my sincere gratitude to my supervisors, Prof. Martin Vingron and Prof. Yuhui Hu. I am grateful to both of them for providing me with the opportunity to enroll in this joint PhD program and for their unwavering support and guidance throughout my studies.

I would like to thank Prof. Yuhui Hu for introducing me to several captivating research projects that broadened my horizons and shaped my research interests in the field of biological sciences. I am particularly grateful for the chance to lead the project on "SARS-Cov-2", which allowed me to contribute to public health and fulfill my aspirations. Although this work is not included in this thesis, it was the first project I completed during my PhD journey. I would also like to express my gratitude to my colleagues who contributed to this project, including Dr. Jing Sun, Yunfei Li, Zhengxuan Li, Yu Xie, Ruoqing Feng, and Prof. Jincun Zhao, for their valuable efforts. I would also like to express my appreciation to Prof. Wei Chen for engaging in insightful discussions.

To Prof. Martin Vingron, I am thankful for recognizing my abilities and tailoring my education accordingly. Thank you, Martin, for guiding me towards the project presented in this thesis, which aligns with my interests in mathematical applications and bioinformatics. Moreover, I appreciate your mentorship in cultivating my independent thinking skills. Instead of providing explicit solutions, you challenged me to seek answers on my own, enabling me to perceive problems holistically and discover efficient

solutions. Your support and optimism have given me the confidence to persist in my research endeavors.

I would like to acknowledge the funding and scholarship provided by Prof. Yuhui Hu and Prof. Martin Vingron. Your financial support has allowed me to sustain a decent life during the pandemic, enabling me to focus on my research work.

I am grateful to my collaborator, Clemens Kohl, for his valuable discussions and contributions to this project. Additionally, I extend my thanks to my colleagues, Daniel Rosebrock and Qinan Hu, for their warm discussions and support. Their involvement has played a significant role in the success of this project.

I would like to express my gratitude to all my colleagues within our department. I am thankful to Gözde Kibar, Aybuge Altay, Clemens Kohl, Dr. Hossein Moeinzadeh, Dr. Maryam Ghareghani, Dr. Prabhav Kalaghatgi, Dr. Stefan Haas, Dr. Peter Arndt, Yufei Zhang, Dr. Wanying Wu, Emel Comak, Nico Alavi, Ekin Deniz Aksu, Dr. Ekta Shah, Dinesh Adhithya Haridoss, Dr. Eldar Abdullaev, as well as past members Meng Zhang, Jieyi Di, Tris Rapakoulia, Robert Schöpflin, and Daniel Rosebrock. I deeply appreciate all of your support and companionship throughout my journey. I would also like to extend my thanks to Dr. Anne-Dominique Gindrat, Martina Lorse, and all the other staff members at our institute who have provided assistance and support.

To my family and friends, I am incredibly grateful for your unwavering love, support, and understanding. I would like to thank my sibling, Zihan Zhao, for assuming additional responsibilities in taking care of and accompanying our parents during my absence from home. It was challenging for me to travel back home during the pandemic, and I am grateful to my sister for making sacrifices for our family. Thanks to her support, I was able to focus on my research and successfully complete my studies. I also want to express my appreciation to my parents for their enduring love and the tremendous efforts they have made to support my dreams. Their encouragement and belief in me have been a constant source of motivation.

Yan Zhao

July, 2023, Berlin

Table of Contents

List of Figures

List of Tables

Acronyms

APL Association Plot. 7

ARI Adjusted Rand Index. 25, 63, 85, 87, 90, 94–99, 103, 118

ATAC-seq Assay for Transposase-Accessible Chromatin with sequencing. 1

BCSpectral Spectral BiClustering algorithm. 35

biMAP MAPping of biclusters. 6, 79–81, 99, 103, 104, 107, 108, 111, 114, 118

CA Correspondence Analysis. iv, 4–7, 12–15, 29–31, 39, 45, 49–51, 56, 59, 60, 62, 67, 79, 81, 99, 102, 103, 112, 118

cabiMAP correspondence analysis factor based MAPping of biclusters. 82, 99, 111, 116, 118

CAbiNet Correspondence Analysis based biclustering on Networks. v, ix, x, 4–8, 37, 42, 51, 67–70, 72, 73, 79, 80, 83, 85, 87–92, 94, 99, 102, 103, 111, 114–119, 133

CCA Cheng and Church Algorithm. 117

CHIP-seq Chromatin Immunoprecipitation Sequencing. 1

GO Gene Ontology. 91

KEGG Kyoto Encyclopedia of Genes and Genome. 91

kNN k-Nearest Neighbour graph. 22, 68, 71, 79

MSR Mean Squared Residue. 33

PBMC Peripheral Blood Mononuclear Cells. 65

PC Principal Component. 4

PCA Principal Component Analysis. 4, 5, 7, 10–12, 14, 38, 39, 49, 79

QUBIC QUaliative BIClustering algorithm. 34

RF Random Forest. 96

RNA-seq RNA sequencing. 3, 11, 32, 33, 36, 48, 83

scATAC-seq single-cell Assay for Transposase-Accessible Chromatin with sequencing. 42

scRNA-seq single-cell RNA sequencing. 1–4, 6, 7, 11, 23, 31–33, 35–37, 42, 43, 45, 61, 63, 67, 83, 84, 90, 94, 112, 117, 118, 133

SIMBA Single-cell embedding along with features. 42

SNN Shared Nearest Neighbour graph. ix, 6, 68–71, 79, 81, 99, 103, 108, 111

SVD Singular Value Decomposition. 8, 35

t-SNE t-distributed Stochastic Neighbor Embedding. 4, 39–41

UMAP Uniform Manifold Approximation and Projection. 4, 40, 41, 79, 82

Unibic Unified Biclustering. 35

1 | Introduction

During the era of genomic biology, significant advances have been made in high-throughput sequencing techniques, enabling the investigation of various facets of biological processes (Soon, Hariharan, & Snyder, 2013). For example, DNA microarrays techniques (TAUB, DeLEO, & Thompson, 1983; Pease et al., 1994; Shalon, Smith, & Brown, 1996; Mirzabekov, Lysov, Shick, & Dubiley, n.d.; Pollack et al., 1999; Churchill, 2002), the next generation high-throughput RNA sequencing, single-cell RNA sequencing (scRNA-seq) (Waern, Nagalakshmi, & Snyder, 2011; Kodzius et al., 2006; Ingolia, Ghaemmaghami, Newman, & Weissman, 2009) and spatial transcriptomic sequencing techniques (Rao, Barkley, França, & Yanai, 2021; Chen et al., 2022a) have been developed to analyze gene expression patterns under diverse experimental conditions or within specific cell types. Another technique named as Assay for Transposase-Accessible Chromatin with sequencing (ATAC-seq) has been developed to study the accessibility of DNA sequence (Buenrostro, Giresi, Zaba, Chang, & Greenleaf, 2013). For other aspects of biological processes, high-throughput mass spectrometry protein profiling has been developed for proteomics studies and Chromatin Immunoprecipitation Sequencing (CHIP-seq) has been designed for histone modification study (Robertson et al., 2007) and so on. All these sequencing techniques collectively contribute to enhancing our comprehension of the intricate molecular mechanisms underlying biological processes.

Over the past decade, a significant volume of scRNA-seq data has been generated along with the development of scRNA-seq techniques. The scRNA-seq technique usually works as demonstrated in Fig. 1.1. It begins with dissecting cells from biological tissues, organs or organisms (Fig. 1.1-1). Then the cells are disassociated and captured by microfluidic devices, e.g. droplet-based platforms. The cells are usually integrated with unique molecule indices (UMIs) which allows to trace the origin of each cell (Fig. 1.1-2). The messenger RNAs (mRNAs) are extracted and converted to complementary DNAs (cDNAs) and the amplified cDNAs are sequenced by high-throughput sequencing machines (Fig. 1.1-3). Then the sequenced reads and UMIs are mapped to reference genome and gene expression levels are quantified by toolkits, generating a gene expression count matrix with genes as rows and cells as columns (Fig. 1.1-4).

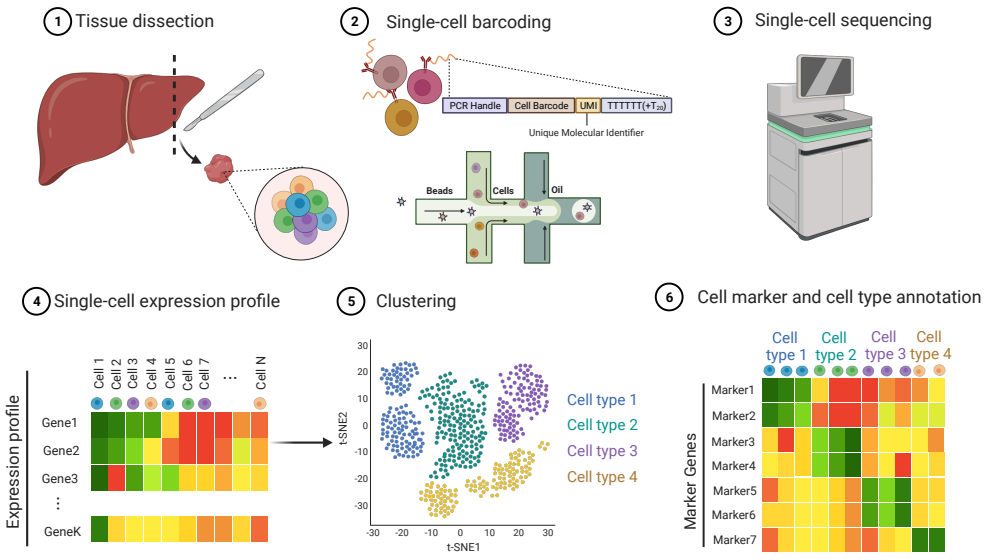

Figure 1.1: **Single cell RNA-sequencing analysis work flow.**

This data can cover diverse species, tissues, developmental stages, sequencing protocols and batch effects when the data is generated by different labs. This introduces new challenges to data analysis as well as computational method development. Consequently, several scRNA-seq analysis toolkits have been developed to assist scRNA-seq data analysis, such as Seurat (Hao, Hao, Andersen-Nissen, et al., 2021), Monocle (Cao et al., 2019) and Scanpy (Wolf, Angerer, & Theis, 2018a). These packages typically share similar workflows that encompass various steps, such as quality control, data normalization, batch correction, feature selection, dimension reduction, clustering (Fig. 1.1-5), differential gene expression analysis and cell-type annotation by detected marker genes (Fig. 1.1-6). While these methods already allow researchers to study the cellular heterogeneity, they still face certain common issues.

A common issue encountered in standard pipelines is the presence of the “double-dipping" problem. Typically, in scRNA-seq analysis algorithms, clustering is performed initially, followed by the identification of markers for each cluster using statistical tests applied to the identified clusters. Upon closer examination of this process, it becomes evident that the clusters are initially differentiated based on features, that is the clusters are driven by these specific features. Subsequently, null hypotheses are formulated, e.g. “the expression levels of gene X in cluster A and cluster B are drawn from the

same distribution". Statistical tests are then conducted on all the features, and those features that exhibit significant p-values and contribute to the differentiation between clusters are recognized as markers for the clusters, thereby aiding in the annotation of cell types. This clustering and marker identification approach follows a circular logic, i.e. the "double-dipping" or "snooping" problem.

To address this issue, one solution is the utilization of biclustering algorithms, which enable the simultaneous grouping of both row and column items. The goal of most of the biclustering algorithms is to detect the green blocks (biclusters) as shown in Fig. 1.1-6, where darker green indicates a higher expression level of genes. This approach helps eliminate the problem of statistical inference inherent in the traditional methodology. It recognizes both cell clusters and cluster-specific genes at a single step, which circumvents the "double-dipping" problem.

There have been many exsiting biclustering algorithms developed for transcriptomic data analysis. The first biclustering algorithm was developed for microarray gene expression analysis by Cheng and Church since 2000 (Cheng & Church, 2000). From then on, more and more biclustering algorithms have emerged to detect subsets of both genes and conditions that share similar patterns in both DNA microarray and the bulk RNA-seq data. However, most of them are developed for microarray assays and bulk RNA-seq data analysis. These types of techniques measure the accumulative expression levels of DNAs/RNAs and the readouts of them are dense matrices with genes as rows and samples as columns. As for scRNA-seq data, the readout matrix are quite sparse, due to the limitation of scRNA-seq library preparation schemes and sequencing bias, e.g. missing labeling of UMIs, losing of RNA segments, sequencing error and so on. The sparsity of scRNA-seq gene expression matrix can be as high as 90%. The existing biclustering algorithms don't take this characteristic of scRNA-seq data into account, so some of the existing biclustering algorithms are not suitable for scRNA-seq data analysis.

Over the past decade, the spatial transcriptomic sequencing techniques have been developed, generating even sparser and noiser data than scRNA-seq (Rao et al., 2021; Chen et al., 2022a). The vast amounts of scRNA-seq data and spatial transcriptomic data has been generated, covering data from different species, tissues, developmental

stages and sequencing protocols. The size of these data sets can be very large, with number of cells raise up to millions This poses new challenges to the scalability of biclustering algorithms. Besides, while scRNA-seq data analysis aims to uncover the cellular heterogeneity and define distinct cell types, some biclustering algorithms only focus on identifying small subsets of gene-cell biclusters, leaving a significant number of cells whose cell types remain unclassified. Therefore, biclustering algorithms developed for scRNA-seq data is needed.

For the routine scRNA-seq analysis pipelines, the cell are always visualized by Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP). Nonlinear embedding techniques such as t-SNE and UMAP can visualize high-dimensional cell coordinates in a two-dimensional map. However, these methods are limited to visualizing either cells or genes separately.

The linear embedding approach PCA has a merit over the non-linear embedding approaches like t-SNE and UMAP. It allows for a simultaneous embedding of the genes and cell with a biplot. A PCA biplot visualizes cells and genes with principal component values and loadings respectively. Loadings refer to the weight of a gene in a Principal Component (PC). However, in a PCA biplot, the scales for the cells' PC scores and the genes' loadings are different, which makes interpretation of genes and cells in this planar challenging. Correspondence Analysis (CA) biplot addresses this issue by rescaling the coordinates of cells and genes and presenting them in the same space. This improves the interpretability of cell-gene relationships compared to a PCA biplot (M. Greenacre, 2007). Nonetheless, linear methods, including biplots, often discard significant information when dealing with large and complex datasets. In single-cell transcriptomic data, the first two PCs typically explain only a small portion of the variance. Thus, a large number of dimensions must be retained to adequately represent the data, compromising visual interpretability. Therefore, a non-linear visualization approach is demanded for a joint visualization of genes and cells. This will provide a more intuitive understanding of the detected biclusters.

To address these limitations, we propose Correspondence Analysis based biclustering on Networks (CAbiNet), a method that facilitates joint visualization and co-

clustering of cells and genes in a planar embedding. Instead of projecting the data into a new space in which the covariance is maximized along the first component by PCA, CA projects data into a new space maximizing the discrepancies of features from mean by applying a different scaling. This scaling scheme allows CA to recognize the genes which are highly associated with cells and be capable of dealing with the high sparsity of scRNA-seq data. CA projects the data with two kinds of scalings, they are principal coordinates and standard coordinates. After the data has been appropriately projected, a suitable large number of dimensions is selected to reduce the dimensionality of data. Following this reduction, clustering is typically performed in the lower-dimensional space to identify distinct groups of cells. Chapter 2 provides more information on the study background of this field, including dimension reduction approaches like CA and PCA, existing clustering algorithms, gene module detection algorithms and biclustering algorithms, and visualization methods.

In Chapter 3, I will discuss about the clustering in CA space. The first section lists the simulated and experimental scRNA-seq data sets that have been used. Since CA is sensitive to outliers, I will further discuss the preprocessing/normalization of the data in the following section. The standard coordinates, principal coordinates and association ration in CA space are then illustrated with a simulated data set. In the last section of this chapter, principal coordinates, standard coordinates and singular vectors are compared to determine which one is the the best for clustering.

In Chapter 4, I will illustrate how we leverage the properties of CA to construct a cell-gene graph where nodes comprise both cells and genes, how the graph is pruned and how cell-gene clusters are detected from this graph. I will then demonstrate new biclustering visualization approaches in Chapter 5, the biMAP and cabiMAP. Both of them allow an intuitive observation of cells and genes in a two dimensional planar.

CAbiNet, serving as a biclustering algorithm, offers the capability to not only simultaneously co-cluster cells and genes, but also detect gene modules within the data. In Chapter 6, the performance of CAbiNet as a biclustering algorithm will be benchmarked against existing biclustering algorithms to showcase its accuracy and computing speed. Additionally, the performance of CAbiNet in gene module detection will also be evaluated.

Optimizing clustering results is a critical aspect of all clustering algorithms, regardless of whether they are used for traditional clustering or biclustering. Different combinations of parameters can yield varying clustering results, making it difficult to determine which result is the most suitable for a given dataset, especially in cases where ground truth clustering is not available. Therefore, it is crucial to develop a methodology that can effectively find out proper parameters and optimize clustering performance. I propose a random forest regression model to predict the clustering quality to get the locally optimized clustering results and this can be found in Chapter 7.

The effectiveness of CAbiNet in accurately co-clustering and embedding cells and genes into a two-dimensional space will be demonstrated using simulated and experimental scRNA-seq and spatial transcriptomic datasets in Chapter 8. I will showcase how the resulting biclusters, biMAPs and cabiMAPs generated by CAbiNet. I will illustrate how CAbiNet can expedite cell type annotation and facilitate the discovery of cell types. Our examples encompass small data sets with well-defined cell types, as well as complex developmental data sets, highlighting the capability of biMAP to generate informative visualizations even for intricate biological experiments.

CAbiNet has been implemented as an R package and can be freely obtained from GitHub (`https://github.com/VingronLab/CAbiNet`). The package is fully compatible with popular scRNA-seq analysis pipeline *SingleCellExperiment*, including those available on Bioconductor. It is worth noting that the aspects related to distance measurements in CA and the creation of a cell-gene graph, and applying community detection methods to co-cluster cells and genes in the cell-gene graph by spectral clustering were initiated by me and my advisor, Martin Vingron. Additionally, my colleague Clemens Kohl contributed to CAbiNet by implementing Shared Nearest Neighbour graph (SNN) graph strategy and adding on the gene pruning function. The construction of the R package CAbiNet and its benchmarking were collaborative efforts between Clemens Kohl and me. We made equal contribution to this aspect of the project.

Lastly, a comprehensive discussion will be presented in Chapter 9, covering the strengths and limitations of all the developed algorithms. Proposed enhancements and future directions will also be explored.

2 | Background

2.1 Overview

This chapter provides a comprehensive overview of the literature and theoretical foundations that form the basis of the research presented in this thesis. It discusses the advancements and challenges in the field of single-cell RNA sequencing (scRNA-seq) data analysis, biclustering and bicluster visualization methods, highlighting the need for improved methods to address key issues.

The chapter begins by introducing a widely used analysis method Principal Component Analysis (PCA) in high-throughput sequencing data analysis. A concise explanation is then provided regarding the methodology and its application in the analysis of single-cell RNA sequencing (scRNA-seq) data. Furthermore, the chapter introduces PCA biplots and examine the advantages and disadvantages associated with their utilization.

Next, a comprehensive introduction to Correspondence Analysis (CA) is presented. CA, similar to PCA, serves as a dimension reduction technique and facilitates the generation of biplots that incorporate both features and conditions. The fundamental principles of CA lay the basis for the biclustering and visualization methods employed in CAbiNet. Additionally, an introduction to the Association Plot (APL) visualization method, specifically designed for exploring dimension-reduced CA data, will be given. APL will serve as complementary evidence to support the findings and results obtained through CAbiNet.

As a dimension reduction method, CA has been applied to denoise the data, the clustering then is done in the dimension reduced space to identify the groups of conditions. Various clustering methods, each employing different scaling techniques, have been implemented within the framework of CA. This chapter will provide an introduction to these methods.

Besides the clustering algorithms mentioned above, CAbiNet utilizes two community detection algorithms to identify the clusters. The algorithms used by CAbiNet will be illustrated in this chapter.

CAbiNet is developed as a comprehensive solution encompassing biclustering and

biclustering visualization techniques. To provide a comprehensive overview of biclustering and visualization methods, this chapter will give a brief introduction of existing biclustering algorithms. The limitations and drawbacks associated with these methods are emphasized, underscoring the necessity of proposing a novel algorithm like CAbiNet. Additionally, the chapter introduces various embedding algorithms that are relevant to the context of the research.

Overall, this chapter serves as a foundation for the thesis, setting the stage for the subsequent chapters where novel methods and their evaluations will be presented.

2.2 Singular Value Decomposition

2.2.1 Singular Value Decomposition

In algebra, the Singular Value Decomposition (SVD) refers to a factorization of a matrix (Golub & Reinsch, 1971). Suppose we have a matrix $\mathbf{X}$ with m rows and n columns, the SVD of matrix $\mathbf{X}$ will decompose it into left singular vectors, singular values and right singular vectors. That is

$$\mathbf{X} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^\mathbf{T}, \tag{2.1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are matrices consist with columns which are singular vectors and $\mathbf{D}_\alpha$ is a diagonal matrix with singular values ranking from largest to smallest as the entries.

The singular values range from 0 to 1, that is $1 \geq \alpha_1 \geq \alpha_2 \geq \dots \alpha_N \geq 0$, where $N = min\{m, n\}$.The dimension of matrix $\mathbf{D}_\alpha$ is determined by the rank of matrix $\mathbf{X}$. In some cases where there are some rows(columns) proportional to each other, the rank of matrix can be smaller than the smaller dimension of the matrix. In such situations, reducing the original space to the number of rank of matrix preserves all the information in data. For most of the cases, the rank of matrix $\mathbf{X}$ equals $min\{m, n\}$.

Eigenvalues of matrix $\mathbf{X}$ are the square of singular values in $\mathbf{D}_\alpha$, which provide insight into the amount of preserved information in each component. A higher eigenvalue indicates a greater amount of explainable information retained in the corresponding dimensions.

Each pair of columns in matrix $\mathbf{U}$ are orthogonal to each other, so is that in matrix $\mathbf{V}$. That is

$$\mathbf{U}^T\mathbf{U} = \mathbf{V}\mathbf{V}^T = \mathbf{I}, \tag{2.2}$$

where $\mathbf{I}$ is an identity matrix with ones as diagonal and zeros as other elements.

Since the singular vectors are orthogonal to each other, they form the basis of a new space. Therefore, the left and right singular vector matrices $\mathbf{U}$ and $\mathbf{V}$ can be used as rotation matrices to transform the original data $\mathbf{X}$ into new spaces.

2.2.2 Number of dimensions

As mentioned, the eigen values are calculated as the square of singular vectors. Typically, a significant portion of the information is captured by the first several principal components, the amount of information vanishes in the higher dimensions. Therefore, a dimension reduction is usually done to reduce the noise in data. The dimension reduced space should be large enough to retain the data characteristics in the meanwhile. To determine the appropriate number of dimensions to retain, various approaches have been developed.

One strategy is to calculate the percentage of information that each dimension preserves. It is the ration between eigen value and the sum of all the eigen values. That is

$$r_i = \frac{\alpha_i^2}{\Sigma_{i=1}^K \alpha_i^2}. \tag{2.3}$$

All r_i sum up to 1. The first several dimensions which sum up to occupy 80% or 90% of the eigen values are retained.

One more method is to calculate the mean of eigen values, the dimensions whose eigen values are above the average level are retained. That is to preserve the dimension is, which satisfy

$$\alpha_i^2 > \frac{\Sigma_{i=1}^K \alpha_i}{K}. \tag{2.4}$$

Another method is the “elbow rule”, where a curve connecting the eigen values in decreasing order is plotted (known as a scree plot). The appropriate number of dimensions to retain is determined by identifying the turning point or“elbow” in the curve

(the **elbow rule**). Beyond this point, the slope of the curve vanishes, indicating that additional dimensions contribute little to the inertia.

2.3 Principal Component Analysis

Principal Component Analysis (PCA) is a multivariate statistical model that has been widely used for data analysis in many research fields. Its origin can be dated back to Pearson (Pearson, 1901) or even Cauchy (Cauchy, 1829; Grattan-Guinness, 2000), Jordan (Jordan, 1874), Cayley, Silverster and Hamilton (Stewart, 1993; Boyer & Merzbach, 2011). Hotelling was the one who firstly termed it as *principal component analysis* (Hotelling, 1933; Abdi & Williams, 2010). PCA serves to denoise and pull the most essential information from the data. By reducing the data dimensionality, PCA retains the primary information in the first few dimensions, leading to improved computational efficiency for subsequent clustering analyses.

Suppose we have a data table $\mathbf{A}$ with m rows (representing samples) and n columns (representing features). In the first step of PCA, the entries $a_{i,j}(i = 1, 2, \dots m, j = 1, 2, \dots n)$ in $\mathbf{A}$ are centered by substrating the mean of whole data sets. Following this, depending on the specific normalization applied, two main types of PCA can be distinguished: covariance PCA and correlation PCA. Covariance PCA involves dividing the entries of the data by either $\sqrt{m}$ or $\sqrt{m-1}$, while correlation PCA divides them by the standard deviation. Both of the schemes are widely used. Suppose the standardized matrix is $\mathbf{X}$ and the entries in $\mathbf{X}$ are $x_{i,j}$, the two variants of PCA can be noted as

$$\begin{aligned} CovariancePCA : x_{i,j} &= \frac{a_{i,j} - \overline{a}_i}{\sqrt{m}} \\ or : x_{i,j} &= \frac{a_{i,j} - \overline{a}_i}{\sqrt{mm-1}}, \\ CorrelationPCA : x_{i,j} &= \frac{a_{i,j} - \overline{a}_i}{\sigma(\mathbf{A})}, \end{aligned} \tag{2.5}$$

where $\overline{a}_i$ is the mean of the i-th row in matrix $\mathbf{A}$.

Then SVD can be performed on the standardized matrix $\mathbf{X}$, that is

$$\mathbf{X} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^\mathbf{T}. \tag{2.6}$$

Using $\mathbf{V}$ as rotation matrix, the matrix $\mathbf{X}$ can be projected into a new space in which the columns in matrix $\mathbf{U}$ are the unit basis of the space, that is

$$\mathbf{P} = \mathbf{X}\mathbf{V} = \mathbf{U}\mathbf{D}_\alpha. \tag{2.7}$$

Rows of $\mathbf{P}$ give the embedding of row items (i.e. samples) in a new space in which variance along the first component are most preserved. According to Equation 2.7, the new embedding of rows can be understood as combination of original rows with weights given by $\mathbf{V}$.

Similarly, rotating the transformed original matrix $\mathbf{X}^T$ with the orthogonal matrix $\mathbf{U}$, the column items can be projected into a new space where the first principal component preserves the most variance among column items. The embedding of column items is given by

$$\mathbf{Q} = \mathbf{X}^\mathbf{T}\mathbf{U} = \mathbf{V}\mathbf{D}_\alpha. \tag{2.8}$$

Eigenvalues of matrix $\mathbf{X}$ are calculated as the square of singular values in $\mathbf{D}_\alpha$. For the covariance PCA, the eigen values tell how numch variance are retained in each dimension. The dimensions that contain most of the variance are usually retained for down-stream analysis, e.g. clustering of data. The number of dimensions is usually determined by the ways introduced in Section 2.2.2.

PCA has been widely used in analyzing transcriptomic data, including microarray data, bulk RNA sequencing (RNA-seq) data and scRNA-seq data (Alter, Brown, & Botstein, 2000; de Haan et al., 2007; Marini & Binder, 2019; Townes, Hicks, Aryee, & Irizarry, 2019; Tsuyuzaki, Sato, Sato, & Nikaido, 2020). PCA condenses the information into a dimension reduced space and gives new representation of features and conditions in this space. The embedding of items in the lower dimensions is then used to find out the clusters of items, which is one of the most common practices in large biological data analysis.

2.4 Correspondence Analysis

2.4.1 Introduction to Correspondence Analysis

PCA can be generalised to Correspondence Analysis (CA) (Abdi & Williams, 2010). CA is a method to represent data matrix in a new space, enabling the visualization and observation of the relationship among rows, columns and between rows and columns (M. Greenacre, 2017). CA was proposed by a German American statistician Herman Otto Hirschfeld and later on developed independently by two French statisticians, Jean-Paul Benzecri (Cazes, Chouakria, Diday, & Schektman, 1997) in the 1970s. CA has been widely used in many research fields, e.g. ecology (Ter Braak & Verdonschot, 1995), sociology (Clausen, 1998), marketing (Bendixen, 1996) and so on. During the past two decades, many researchers have contributed to CA (M. J. Greenacre, 1984; Řeháková, 1986; M. Greenacre, 2007; Beh & Lombardo, n.d.). Based on Greenacre's book (M. Greenacre, 2017), I will introduce what is the canonical CA and how it works.

The computation of CA involves several steps. Let $\mathbf{A}$ be a matrix with positive values and with m rows and n columns. Firstly, CA transforms the matrix $\mathbf{A}$ into a frequency/proportion matrix $\mathbf{P}$ in which element p_{ij} in $i-th$ row and $j-th$ column can be written as

$$p_{ij} = \frac{a_{ij}}{a_{++}}. \tag{2.9}$$

Here, a_{ij} denotes the value in the $i-th$ row and $j-th$ column in matrix $\mathbf{A}$, and a_{++} the grand total of $\mathbf{A}$. Considering $\mathbf{A}$ as a gene expression count matrix with genes as rows and cells/conditions as columns, each element in $\mathbf{P}$ represents the observed probability that a gene is expressed in a cell/condition. The expected probability of each entry can be denoted by the multiply of row and column masses, that is

$$e_{ij} = r_i * c_j, \tag{2.10}$$

where r_i is the sum over row i in $\mathbf{P}$, and c_j the sum of $j-th$ column of $\mathbf{P}$:

$$r_i = \sum_{j=1}^{m} p_{ij}, \quad c_j = \sum_{i=1}^{n} p_{ij}. \tag{2.11}$$

Then the CA calculates Pearson residuals

$$s_{ij} = \frac{p_{ij} - e_{ij}}{\sqrt{e_{ij}}} \tag{2.12}$$

to indicate how much an observed value is different from the expected probability. Taking the gene expression matrix as an example, the more a gene is specifically highly expressed in a cell/condition, the larger the Pearson residual is.

Then the Pearson residuals matrix $\mathbf{S}$ is submitted to singular value decomposition (SVD), factoring it into product of three matrices

$$\mathbf{S} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^T. \tag{2.13}$$

Columns of U are the left singular vectors of $\mathbf{S}$, columns of $\mathbf{V}$ are the right singular vectors. $\mathbf{D}_\alpha$ is a diagonal matrix with singular values ranking from largest to smallest as the entries. The dimension of $\mathbf{D}_\alpha$ is determined by the rank of matrix $\mathbf{S}$, we denote it as $K, K \leq min\{I, J\}$.

Re-scaling the singular vectors, the row and column items in $\mathbf{P}$ can be transformed into new low dimensional spaces, call it CA spaces. Re-scaling the singular vectors with different weights gives different coordinate systems: standard coordinates and principal coordinates. The standard coordinates are obtained by weighting singular vectors with square root of row or column masses.

Standard coordinates $\mathbf{\Phi}$ of rows:

$$\mathbf{\Phi} = \mathbf{D}_r^{-\frac{1}{2}}\mathbf{U}. \tag{2.14}$$

Standard coordinates $\mathbf{\Gamma}$ of columns:

$$\mathbf{\Gamma} = \mathbf{D}_c^{-\frac{1}{2}}\mathbf{V}. \tag{2.15}$$

Furthermore, re-scaling the standard coordinates gives the principal coordinates of

rows and columns. Principal coordinates $\mathbf{F}$ of rows:

$$\mathbf{F} = \mathbf{D}_r^{-\frac{1}{2}}\mathbf{U}\mathbf{D}_\alpha = \mathbf{\Phi}\mathbf{D}_\alpha \tag{2.16}$$

Principal coordinates $\mathbf{G}$ of columns:

$$\mathbf{G} = \mathbf{D}_c^{-\frac{1}{2}}\mathbf{V}\mathbf{D}_\alpha = \mathbf{\Gamma}\mathbf{D}_\alpha. \tag{2.17}$$

2.4.2 Inertia

The total inertia in the Pearson Residual matrix $\mathbf{S}$ is defined as the sum of squares of entries, which is

$$inertia = trace(\mathbf{S}\mathbf{S}^{\mathbf{T}}) = \sum_{\mathbf{i=1}}^{\mathbf{I}}\sum_{\mathbf{j=1}}^{\mathbf{J}} \frac{(\mathbf{p_{ij}} - \mathbf{r_i c_j})^{\mathbf{2}}}{\mathbf{r_i c_j}}. \tag{2.18}$$

The sum of squares of singular values (i.e the eigen values) of matrix $\mathbf{S}$ also recovers the total inertia:

$$inertia = \sum_{k}^{K} \alpha_k^2 = \sum_{k}^{K} \lambda_k. \tag{2.19}$$

Rows and columns have the same total inertia. The contributions of each dimension to to the total inertia is:

$$contribution : \frac{\lambda_k}{\sum_k^K \lambda_k} \tag{2.20}$$

Different with PCA which maximizes the variance, CA maximize the inertia of data. The first dimensions preserve most of the inertias. But similar with PCA, the dimension reduction can also be done as the ways mentioned in Section 2.2.2.

2.4.3 CA biplot

The row and column items can not only be plotted separately in the low dimensional space, they can also be visualized in a single plot, called the biplot. CA allows a combination of the row and column visualization in a single planar. The rows and columns can either be plotted with principal coordinates or the standard coordinates,

which gives four different combinations in total, namely four kinds of biplots. Depending on which coordinates are used, the biplots can be categorized into two classes: the symmetric biplots and asymmetric biplots.

- **Symmetric biplot**: both row items and column items are with either principal coordinates or standard coordinates.
- **Asymmetric biplot**: row items are with principal coordinates and column items with standard coordinates, or row items are with standard coordinates and row items with principal coordinates.

An illustration of the biplots of a simulated data set can be found from the next Chapter in Section 3.3.

2.4.4 χ^2 distance

A proper choice of distance measure is a key step in finding clusters from a CA space. As mentioned above, χ^2 statistic measures overall discrepancies between observed and expected frequencies. The χ^2 distance between two rows is given by

$$\chi^2(i,i') = \sum_j^J \frac{(\frac{p_{ij}}{r_i} - \frac{p_{i'j}}{r_{i'}})^2}{c_j}. \tag{2.21}$$

The χ^2 distance between two columns is

$$\chi^2(j,j') = \sum_i^I \frac{(\frac{p_{ij}}{c_j} - \frac{p_{ij'}}{c_{j'}})^2}{r_i}. \tag{2.22}$$

The χ^2 distance between one row profile and row average vector (the vector of columns masses c_j) can be written as

$$\chi^2 = \sum_j^J \frac{(\frac{p_{ij}}{r_i} - c_j)}{c_j}. \tag{2.23}$$

Similarly, the χ^2 distance between one column profiles (m_{ij}/m_{i+} or p_{ij}/c_j) and column average vector (the vector of rows masses r_i) can be written as

$$\chi^2 = \sum_{i}^{I} \frac{(\frac{p_{ij}}{c_j} - r_i)}{r_i}. \tag{2.24}$$

With those notations, the total inertia formula can be explained by a weighted sum of χ^2 distances between row profiles and row average profile, which is

$$Inertia = \frac{\chi^2}{m_{++}} = \sum_{ij} \frac{(p_{ij} - r_i c_j)^2}{r_i c_j} = \sum_{ij} r_i \frac{(\frac{p_{ij}}{r_{[i}} - c_j)^2}{c_j}. \tag{2.25}$$

Similarly, it can also be written as a weighted sum of χ^2 distances between column profiles and column average profile, which is

$$Inertia = \sum_{ij} c_j \frac{(\frac{p_{ij}}{c_{[j}} - r_i)^2}{r_i}. \tag{2.26}$$

2.4.5 χ^2 Statistic

It is clear that the observed frequencies are always going to be different from the expected frequencies, though it is assumed the rows/columns are homogeneous. χ^2 Statistic gives a measure of how large the discrepancies between observed and expected frequencies are. Accumulating the discrepancies between all the observed entries and their corresponding expected frequencies results the χ^2 Statistic

$$\chi^2 = \sum \frac{(observed - expected)^2}{expected}, \tag{2.27}$$

which can be written by the frequency table elements as

$$\chi^2 = \sum_{ij} \frac{(m_{ij} - m_{++} r_i c_j)^2}{m_{++} r_i c_j} = \sum \frac{m_{++}(p_{ij} - r_i c_j)^2}{r_i c_j}. \tag{2.28}$$

The larger this value, the more discrepant the observed and expected frequencies are, i.e. the less convinced that the assumption of homogeneity is correct. Furthermore, the square root of the χ^2 statistic is defined as χ^2 distance.

2.4.6 Relationship between inertia, Pearson Residuals and χ^2 Statistic

Recall the definition of total inertia (2.19), it can also be re-written as the χ^2 Statistic divided by the ground total of observed values in M.

$$Inertia = \frac{\chi^2}{m_{++}} = \sum_{ij} \frac{(p_{ij} - r_i c_j)^2}{r_i c_j} = \sum_{ij} s_{ij}^2, \tag{2.29}$$

which is the sum of squares of Pearson Residuals. If all the profiles are identical and thus lie at the same point (their average), all chi-square distances are zero and the total inertia is zero. On the other hand, maximum inertia is attained when all the profiles lie exactly at the vertices of the profile space, in which case the maximum possible inertia can be shown to be equal to the dimension of the space.

2.4.7 Reconstitution formula

Due to Equation 2.13, 2.14,2.15,2.16,2.17, the relationship between original count and the standard and principal coordinates of rows and columns can be written as

$$\frac{p_{ij} - r_i c_j}{r_i c_j} = \sum_{k=1}^{K} f_{ik}\gamma_{jk} + \epsilon_{ij}. \tag{2.30}$$

This is named as reconstitution formula, which can also be written as

$$p_{ij} = r_i c_j \left(1 + \sum_{k=1}^{K} f_{ik}\gamma_{jk} + \epsilon_{ij}\right). \tag{2.31}$$

2.4.8 Association Plots

Association Plot (APL) is a recently developed method to visualize associations in high-dimensional correspondence analysis biplots (Gralinska & Vingron, 2023; Gralinska, Kohl, Fadakar, & Vingron, 2022).

AP is designed for data sets which have well-defined column (sample) clusters, it permits to visualize association between samples (columns) and genes (row profiles). In AP, a cluster of conditions $C = j_1, j_2, \ldots, j_K$ can be represented by the centroid of

its condition vectors $\omega_{j_l}, l = 1, 2, \ldots, K$ in CA-space. The centroid is defined as $\overrightarrow{X}$,

$$\overrightarrow{X} = \frac{1}{K} \sum_{l=1}^{K} \omega_{j_l}. \tag{2.32}$$

Take row asymmetric map as an example, the AP is calculating association ratio between row principle coordinates f_{ik} and column standard coordinates γ_{jk}. Let $\mathbf{f}_i$ donating *i*-th row (gene) vector in CA-space, then the association between $\mathbf{f}_i$ and the selected clusters C is

$$a(\mathbf{f}_i, C) = \frac{1}{K} \sum_{l=1}^{K} \langle \mathbf{f}_i, \omega_{j_l} \rangle + \epsilon = \frac{1}{K} \sum_{k=1}^{K^\star} \sum_{l=1}^{K} f_{ik} \gamma_{kj_l} + \epsilon. \tag{2.33}$$

where $K^\star$ is the reduced dimension, which is determined by the strategies mentioned in Section 2.2.2.

Notably, when $K^\star = rank(\mathbf{S}^T\mathbf{S})$, $\gamma_j (for j \notin C)$ is orthogonal to $\overrightarrow{X}$, $a(\gamma_j, C)$ theoretically equals zero.

Now use $\overrightarrow{v}$ to represent any row or column vector. AP visualizes the association ratios in a 2-D space, by projecting the vector to the centroid of the defined cluster of items $\phi(\overrightarrow{v})$. Suppose the angle between vectors $\overrightarrow{v}$ and $\overrightarrow{X}$ is $\theta(\overrightarrow{v})$, then the 2-dimensional Association Plot for cluster C will contain points $(x(\overrightarrow{v}), y(\overrightarrow{v}))$,

$$\begin{aligned} x(\overrightarrow{v}) &= |\overrightarrow{v}| cos(\phi(\overrightarrow{v})) \\ y(\overrightarrow{v}) &= |\overrightarrow{v}| \sqrt{1 - cos^2(\phi(\overrightarrow{v}))}. \end{aligned} \tag{2.34}$$

For any user-defined clusters, AP will calculate the centroid of the cluster and project all the other profiles, both rows and columns, to the centroid. With the projection $(x(\overrightarrow{v}), y(\overrightarrow{v}))$, the points can be visualized as an association plot.

Points closer to the x-axis, indicating a smaller value of $y(\overrightarrow{v})$, are more indicative of the chosen cluster's specificity, whereas larger values suggest that other clusters also exhibit a competitive presence. Hence, the gene points that more to the right of x-axis, the more associated with the cells in the observed cluster. In addition to the association ratio, a supplementary heuristic statistic denoted as S_α is introduced. It can be written

as

$$S_\alpha(x, y) = x - \frac{y}{\tan \alpha}. \tag{2.35}$$

where x and y are the coordinates of a point in the association plot. When points have the same projection onto x-axis in the association plot, this statistic prioritizes points with smaller projection onto the y-axis. α is determined by permutation of the data. The data is firstly permutated, then the association plot of the permutated data is drawn. There is usually a 'V'-shape cloud of points close to origin in the association plot. Angle between points and the x-axis is calculated. The angle α that delineates 1% of the points to the right of the V is denoted as the cutoff used in Equation 2.35.

The S_α score aims to assign higher scores to points further to the right, while simultaneously decreasing scores as one moves upward. This selection of S_α enables differentiation among observations that would otherwise possess identical association ratios with respect to a cluster.

2.5 Clustering methods

Clustering is an algorithm to group a set of data points into clusters, such that data points in the same cluster are more similar to each other comparing with those from other clusters. The clustering algorithms are aimed at uncovering hidden structure of the data. Many clustering algorithms have been proposed, including connectivity based clustering methods, e.g. hierarchical clustering (Murtagh & Contreras, 2012, 2017); centroid based methods, such as K-means clustering (Hartigan & Wong, 1979) and spherical K-means clustering (Hornik, Feinerer, Kober, & Buchta, 2012); density based algorithms (Ester, Kriegel, Sander, & Xu, 1996); graph based algorithm, e.g. Louvain clustering (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008), Leiden clustering (Traag, Waltman, & Van Eck, 2019) and spectral clustering (Von Luxburg, 2007); and so on. Depending on the characteristics of data points and the purpose of study, the clustering algorithms can be applied.

I will introduce several clustering algorithms that are used, including centroid based clustering algorithms, e.g. K-means clustering and spherical K-means clustering, and graph based clustering algorithm: spectral clustering, Louvain clustering and

Leiden clustering. I will also introduce some widely used metrics for evaluating clustering results.

2.5.1 K-means clustering

K-means clustering is a popular algorithm developed for partitioning a dataset into K distinct, non-overlapping clusters (Hartigan & Wong, 1979). The goal of the algorithm is to group similar data points together and assign them to clusters based on certain features or characteristics. Here's a brief overview of how the K-means algorithm works:

- Initialization: Choose K initial cluster centroids randomly from the data points. These centroids represent the initial cluster centers.
- Assignment: Assign each data point to the cluster whose centroid is the closest based on some distance metric, commonly Euclidean distance. The data points are assigned to the cluster with the nearest centroid.
- Update Centroids: Recalculate the centroids of the clusters by taking the mean of all data points assigned to each cluster. These new centroids represent the updated cluster centers.
- Repeat: Repeat steps 2 and 3 until convergence. Convergence occurs when the centroids no longer change significantly or when a predefined number of iterations is reached.

Selecting the appropriate value for K (the number of clusters) is a critical factor in the K-means algorithm and holds substantial influence over the outcomes. Various techniques (Ray & Turi, 1999; Sugar & James, 2003), such as the elbow method, can be employed to identify an optimal value for K. In the elbow method, K is varied within a specified range (typically from 1 to 20), and the within-cluster sum of squares (WCSS) — representing the sum of the squared distances between points within a cluster and the cluster centroid — is calculated and visualized as a scatter plot. The point at which a distinct bend or "elbow" occurs in this plot indicates a suitable choice for K.

K-means clustering is computationally efficient and works well for many real-world applications. However, K-means clustering has some limitations. It assumes that clusters are spherical and equally sized, which may not be suitable for all types of data. Additionally, the algorithm's performance can be sensitive to the random initialization of centroids.

2.5.2 Spherical k-means clustering

The spherical k-means clustering (Hornik et al., 2012) is a variant of the traditional K-means clustering. It works similarly with the traditional k-means clustering. The only difference is that the distance between data points is calculated by cosine dissimilarity instead of Euclidean distance. The objective of this algorithm is to minimize the within cluster cosine distance. This approach eliminates the bias brought by the length of vectors without losing too much speed of calculation.

2.5.3 Spectral Clustering

Spectral clustering (Von Luxburg, 2007) is a clustering algorithm that leverages graph theory and linear algebra techniques to partition data points into distinct groups, with the aim of grouping together points that exhibit similarity. The process begins by building a graph which consists of data points as nodes and similar nodes are connected with edges. Then the adjacency matrix of this graph is transformed into graph Laplacian, the spectrum of which is further calculated to assist the identification of clusters. More details about this algorithms are offered as below.

For a given set of data points (e.g., columns in the matrix $\mathbf{M}$), similarity among the points can be calculated using metrics such as cosine similarity or Pearson correlation. This similarity measures allow the construction of a graph, denoted as $G = (V, E)$, where vertices V represent the data points, and E represents the edges that connecting the vertices.

The similarity graph can be built up based on different criteria: 1). connecting all the vertices with edges weighted by the similarity scores generates a fully connected graph. 2). Only connect nodes when their similarity score are positive or larger than a threshold, weighted either with the similarity scores or binary values. A graph with bi-

nary weights are called unweighted graph. 3). Alternatively, the similarity graph can be constructed by connecting each vertex to its k nearest neighboring vertices, known as a k-Nearest Neighbour graph (kNN) graph. If the similarity between vertex pairs (x_i, x_j) and (x_j, x_i) is the same, then the similarity graph is an undirected graph and its adjacency matrix is symmetric. Otherwise, the graph becomes directed and the adjacency matrix of the graph is asymmetric.

Suppose $G = (V, E)$ is an undirected graph with non-negative weights. We denote the adjacency matrix of the graph as $\mathbf{W} = (w_{i,j})_{i,j}$, the degree of a vertex $v_i \in V$ is defined as

$$d_i = \sum_{j=1}^{n} w_{i,j}. \tag{2.36}$$

A degree matrix $\mathbf{D}$ is defined as a diagonal matrix with the degrees $d_i, i = 1, \ldots, n$ on the diagonal. The unnormalized graph Laplacian matrix is defined as

$$L = D - W. \tag{2.37}$$

The graph Laplacian can be normalized by several ways, one of which is through symmetric normalization. In this approach, the graph Laplacian is normalized by the degree matrix:

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}. \tag{2.38}$$

Another way of normalization is defined as

$$L_{rw} = D^{-1} L = I - D^{-1} W. \tag{2.39}$$

L_{sym} and L_{rw} are positive semi-definite and can be diagonalized as follows

$$L = V^T \Lambda V, \tag{2.40}$$

where Λ is a diagonal matrix with eigenvalues of the graph Laplacian. Different from the standard eigenvalue decomposition where the eigen values are ordered in decreasing manner, the eigenvalues are in an increasing order: $0 = \lambda_1 \leq \cdots \leq \lambda_n$ with at least one eigenvalue equals 0 for the decomposition of graph laplacian. The columns in matrix V

are eigenvectors which are orthogonal to each other. The eigenvalues again can be used to determine how many clusters are there in the data set and the eigen vectors can be applied with clustering algorithms to detect clusters in the data.

For an undirected graph $\mathbf{G}$ with non-negative weights, the number of connected components in the graph equals the number of eigenvalues equal to 0 of the normalized Laplacian L_{sym} and L_{rw} (Von Luxburg, 2007). This property of the Laplacian can be used to detect the connected components in the graph, namely the clusters in data, hence this algorithm is named as (normalized) spectral clustering.

In some cases, the normalized Laplacian of certain graphs may have only one eigenvalue equal to zero, indicating that the graph is fully connected with only one connected component. However, within this graph, there may still be variations in connectivity strength among the nodes, with some nodes being strongly connected and others weakly connected. To detect the strongly connected components or clusters within this graph, the **eigengap** rule can be applied (Bolla, 1991). The eigengap method aims to determine the number of eigenvectors, k, to be used for clustering. The first k eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$ approximate zero, while λ_{k+1} is significantly larger, and the gap between λ_k and λ_{k+1} is the largest comparing with the differences between all the other successive eigenvalues. This value k is then used as the estimated number of clusters in the data, and then clustering algorithms can be applied to the first k eigenvectors to define the clusters. The applicable algorithms include K-means clustering, DBSCAN and so on.

2.5.4 Modularity based clustering algorithms

Louvain clustering (Blondel et al., 2008) and Leiden clustering (Traag et al., 2019) are two modularity based clustering methods, which have been widely used by scRNA-seq data analysis tools, e.g. Seurat (Stuart et al., 2019; Hao, Hao, Andersen-Nissen, et al., 2021), Scanpy (Wolf, Angerer, & Theis, 2018b). Both Louvain and Leiden clustering aim to maximize the modularity of a graph, although they differ in their optimization procedures.

For a simple undirected unweigted graph, the **modularity** is defined as (Newman,

2006):

$$Q = \frac{1}{2E}\sum_{ij}(A_{ij} - \frac{d_i d_j}{2E})\frac{s_i s_j + 1}{2} = \frac{1}{2E}\sum_{c}(A_c - \frac{d_c d_c}{2E}) \tag{2.41}$$

where

- A_{ij} is adjacency matrix of the graph,
- E is the number of edges in graph, and $2E = \sum_i d_i = \sum_{ij} A_{ij}$,
- d_i and d_j are the degree of vertices. $d_i = \sum_j A_{ij}$,
- $s_i = 1$ if vertex i belongs to group 1 and $s_i = -1$ if it belongs to group 2,
- $\frac{d_i d_j}{2E}$ is the expected number of edges between vertices i and j if edges are placed at random.

Louvain clustering. The louvain clustering is designed to minimized the modularity of the graph, such that the strongly connected components can be detected. It works as the following steps:

- Initialization. Each node is initialized as a cluster.
- Move nodes and Agglomeration. Then the pair of adjacent clusters by merging which results the maximized increase of modularity is joined into a new cluster.
- Iteration. The second step is repeated until there is no increase of modularity when merging any pair of the adjacent clusters.

The Louvain algorithm is computationally efficient, allowing it to deal with large community networks. However, it suffers the problem of disconnected communities. This problem happens when a node which is the only connection between two subclusters being moved to another cluster. The two subclusters that are bridged by this node are partitioned into one cluster while being disconnected.

Leiden clustering The Leiden clustering is developed to solve the drawbacks of Louvain algorithm, but it allows a user defined resolution parameter to control the size of detected connected components. The modularity of Leiden clustering is similar with

Equation 2.41, but with an extra resolution parameter $\gamma(\gamma > 0)$

$$Q = \frac{1}{2E}\sum_{c}(A_c - \gamma\frac{d_c d_c}{2E}). \tag{2.42}$$

A larger resolution produce more clusters, while smaller resolutions produce fewer clusters. Similar with Louvain algorithm, Leiden algorithm works as follows:

- Initialization. Each node is initialized as a cluster.
- Local move of nodes. Different from Louvain algorithm, Leiden algorithm allows nodes to be merged with neighbouring nodes/clusters which lead to modularity increase instead of largest modularity increase in Louvain algorithm. The neighbouring node/cluster that being merged is randomly selected. The larger the increase in modularity, the more probably to be merged.
- Refine of the clusters. The nodes that on their own can be merged into to clusters until there is no increase of modularity.
- Iteration. The processes above are repeated until no further improvement can be made.

The Leiden algorithm solves the problem of dis-connectivity with the random local move and node refining procedures. The Leiden algorithm is shown to outperform Louvain algorithm both in speed and clustering accuracy(Traag et al., 2019).

2.5.5 Clustering Evaluation metrics

The quality of clustering results can be assessed by many criteria, including intrinsic measures and extrinsic measures. The silhouette score, entropy (Meilă, 2007), Calinski Harabatz score (Caliński & Harabasz, 1974), and Davies-Bouldin score are intrinsic measures. These measures are based on intra- and inter-cluster distances and only require detected clusters and the original matrix as input. On the other hand, extrinsic measures, e.g. Adjusted Rand Index (ARI) (Hubert & Arabie, 1985), calculate the overlap between detected clusters and ground-truth clusters, which requires the ground-truth of clusters as extra input. Here I give a brief introduction of the metrics that have been used in this thesis.

Suppose we have a set of data points $x_1, x_2, \ldots, x_n, x_i \in R^m$ and the data points are clustered into N clusters: $C_1, C_2, \ldots, C_N$ and the number of points in each cluster is denoted as $|C_1|, |C_2|, \ldots, |C_N|$. The centroid of each cluster is $c_1, c_2, \ldots, c_N$. Denote the centroid of all the points as e and number of all points as n. The following lists some of the **intrinsic measures**.

Silhouette score. Suppose sample x_i belongs to cluster C_I. The Silhouette score for a particular sample x_i is calculated as follows:

- Calculate the average distance between the sample x_i and all other samples within the same cluster:

$$a(i) = \frac{1}{|C_I - 1|} \sum_{i,j \in C_I, i \neq j} d(x_i, x_j), \tag{2.43}$$

  where $d(x_i, x_j)$ is the Euclidean distance between points with principal coordinates, standard coordinates or singular vectors.

- For each neighboring cluster ($C_J, J \neq I$), calculate the average distance between the sample x_i and all samples in the cluster C_K. Take the minimum of these average distances across all neighboring clusters, and denote it as $b(i)$:

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{I \in C_J} d(i, j). \tag{2.44}$$

- Compute the Silhouette score $s(i)$ for the sample x_i using the formula:

$$s(i) = \begin{cases} 0, & \text{if } |C_I| = 1, \\ \frac{b(i) - a(i)}{\max a(i), b(i)}, & \text{if } |C_I| > 1. \end{cases} \tag{2.45}$$

Entropy. Assuming points in the data set have the same opportunity to be grouped into each cluster, then the possibility that a point is clustered into cluster C_k is

$$P(k) = \frac{|C_k|}{n}. \tag{2.46}$$

The uncertainty of the grouping of clusters is defined as entropy of the random variable

$P(k)$, that is

$$H(C) = -\sum_{k=1}^{N} P(k) log P(k). \tag{2.47}$$

Since the probability $P(k)$ is no larger than 1, the entropy is always non-negative. The entropy equals 1 when and only when there is no uncertainty in clustering the points, i.e. when there is only one cluster.

Calinski Harabatz score is the ratio between intra-cluster and inter-cluster dispersion. The inter-cluster dispersion is defined as

$$OC = \sum_{i=1}^{N} |C_i|(c_i - e)(c_i - e)^T. \tag{2.48}$$

The intra-cluster dispersion is defined as

$$IC = \sum_{i=1}^{N} \sum_{x \in C_i} (x - c_i)(x - c_i)^T. \tag{2.49}$$

The Calinski Harabatz score is defined as

$$CH = \frac{OC}{IC} * \frac{n - N}{N - 1}. \tag{2.50}$$

Davies-Bouldin score also measures the ratio between within-cluster distance and between-cluster distance. Firstly, the within cluster distance is calculated as

$$S_i = \left(\frac{1}{|C_i|} \sum_{j=1}^{|C_i|} ||x_j - c_i||^q \right)^{\frac{1}{q}}. \tag{2.51}$$

If $q = 1$, S_i turns out to be the average Euclidean distance of vectors in cluster i to the centroid of cluster i. If $q = 2$, S_i is the standard deviation of the distance of samples in a cluster to the respective cluster centroid. The separation between cluster C_i and C_j can be measured as

$$M_{ij} = ||c_i - c_j||_p = \left(\sum_{k=1}^{K} |cc_{ki} - cc_{kj}|^p \right)^{\frac{1}{p}}. \tag{2.52}$$

cc_{ki} is the k-th element of the centroid of cluster c_i and there are K elements/features in each data point. If $p = 2$, M_{ij} measures the Euclidean distance between centroids of clusters.

Then for each pair of clusters, the ratio of within cluster and between cluster distance is calculated as

$$R_{ij} = \frac{S_i + S_j}{M_{ij}}. \tag{2.53}$$

The larger the R_{ij} is, the worse the cluster C_i and C_j are separated. For cluster C_i, the worst separation between it with all the other clusters is written as

$$D_i = \max_{i \neq j} R_{ij}. \tag{2.54}$$

The Davies-Bouldin score is then defined as

$$DB = \frac{1}{N} \sum_{i=1}^{N} D_i. \tag{2.55}$$

The smaller the DB is, the better the clustering result is. For all the usage of Favies-Bouldin score in this study, the values of p and q are all set as 2.

One of the **extrinsic measures** is the Adjusted Rand Index (ARI).

ARI. The Adjusted Rand Index (ARI) (Steinley, 2004) is a metric used to quantify the similarity between two clusterings of data, normally one is the detected clustering, the other one the ground-truth clustering. It is an improvement over the Rand Index, which is a simple measure of similarity between set of clusters but is susceptible to chance. The ARI addresses this limitation by incorporating a correction factor that accounts for the expected similarity due to random chance. By considering this correction, the ARI provides a more reliable measure of the agreement between two set of clusters.

Suppose $U_1, U_2, \ldots, U_I$ and $V_1, V_2, \ldots, V_J$ are two sets of clustering results. Denote the size of $U_i \cap V_j$ is n_{ij} The ARI can be written as

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{|U_i|}{2} \sum_j \binom{|V_j|}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{|U_i|}{2} + \sum_j \binom{|V_j|}{2}\right] - \left[\sum_i \binom{|U_i|}{2} \sum_j \binom{|V_j|}{2}\right] / \binom{n}{2}}. \tag{2.56}$$

2.6 Existing clustering methods in CA space

2.6.1 Hierarchical clustering in full dimensional space

Greenacre proposed a method to partition rows or columns into groups by minimizing with-in group inertia and maximizing between-group inertia (M. Greenacre, 2007). Taking row clustering as an example, the χ^2 distance measure between rows and the centroid of the row group can be written as d_i,

$$d_i = \frac{(\frac{p_{ij}}{r_i} - c_j)^2}{c_j}. \tag{2.57}$$

Combining equations (2.25, 2.57), the total inertia can be written as

$$Inertia = \sum_i r_i d_i^2. \tag{2.58}$$

Suppose the row items can be divided into G groups, we can note the row sets of merged groups as $r_g(g = 1, 2, ..., G)$, row centroid of merged rows in each group as c_g, the χ^2 distance between row groups and group centroid as d_g, the χ^2 distance between row group centroids as d_{ig}, the total inertia then can be rewritten as

$$Inertia = \sum_i r_i d_i^2 = \sum_g r_g d_g^2 + \sum_g \sum_{i \in g} r_i d_{ig}^2, \tag{2.59}$$

which is the addition of intra- and inter-group inertia.

To identify the clusters among row items, the initial step involves treating each item as an individual cluster, where the inter-group inertia is equivalent to the total inertia. The inter-group inertia is maximized at the initialization. Whenever two clusters are merged as a new cluster, the inter-group inertia would be reduced. The next step aims to identify the pair of clusters that, when merged, minimizes the reduction in total inter-group inertia, resulting in a new cluster. This process is repeated iteratively for the updated clusters, merging groups that yield the least reduction in inter-group inertia and eventually leading to the formation of clusters with the lowest intra-group inertia and highest inter-group inertia. Clustering for column items follows a similar approach.

However, this method copes with the entire dimensional frequency table, which could include uninformative noise when dealing with complicated biological data sets. Therefore, clustering in a dimension reduced space is necessary. Hereby, methods have been developed to identify clusters in the dimension reduced space of CA. I will discuss some of the published methods in the following Section 2.6.2 and 2.6.3.

2.6.2 Combined K-means clustering and dimension reduction approaches

Numerous algorithms have been developed to cluster the columns (categories-/conditions) within the dimension reduced CA space. Some of the algorithms tried to do the dimension reduction and clustering simultaneously. For example, Van Buuren and Heiser (Van Buuren & Heiser, 1989) developed a joint dimension reduction and clustering algorithm *GROUPALS*. It clusters the categories of columns with K-means algorithm (MacQueen, 1967) and detected optimal number of low dimensions by minimizing the average within group variances while updating the K-means clusters. Another algorithm that combines dimension reduction and K-means clustering to identify column clusters is the *cluster CA*, which is proposed by M. van de Velden, et al.(Van de Velden, D'Enza, & Palumbo, 2017). Instead of minimizing the within group variances by *GROUPALS*, *cluster CA* is done by maximizing the sum of between group variances. It has been shown to perform comparably with *GROUPALS*. Additionally, MCA K-Means (Hwang, Montréal, Dillon, & Takane, 2006) also combines the dimension reduction and clustering in one goal. It integrates the objective function of dimension reduction and K-means clustering by assigning user defined weights that sum up to 1.

As evaluated by M. van de Velden, et al., all the methods mentioned above outperform K-medoids clustering using Gower distances in the full dimensional space (Van de Velden et al., 2017). Notably, the distance measure used by all the methods mentioned above is the Euclidean distance among standard coordinates of columns. However, there are methods that measure the distance by different ways. I will introduce them in the next subsection 2.6.3.

Another important concern is that all K-means-based algorithms share a common challenge, which is the requirement of a user-defined input for the number of clusters, denoted as k. The chosen value of k influences both the clustering and dimension

reduction processes. However, determining an appropriate number of clusters is often subjective and difficult. Several methods have been proposed to address this issue, such as the “elbow method” 2.2.2 or “silhouette analysis” 2.5.5. It is typically done by varying the value of k and running the clustering algorithms with different values of k. Then the number of clusters that minimizes or maximizes the objective function will be used as the determinant of the value of k. However, these methods might not always yield a definitive answer. A general strategy that helps to optimize the clustering results is still needed.

2.6.3 Graph based clustering in dimension reduced CA space

The dimension reduction and clustering steps can also be performed independently of each other. Community detection methods, such as graph clustering, can be employed to group data points in the dimension-reduced space obtained through CA. An example of this is the application of the walktrap nearest neighbor graph clustering algorithm, as implemented in the *corral* tool (Hsu & Culhane, 2023a), to cluster cells. In the graph built by *corral*, the edges between nodes are determined based on the Euclidean distance calculated from the singular vectors (Hsu & Culhane, 2023a). In this approach, dimension reduction is first conducted using the elbow rule with a scree plot. By selecting the components corresponding to the "elbow" point in the scree plot, the dimensionality of the data is effectively reduced.

After dimension reduction, the authors employ the walktrap nearest neighbor graph clustering algorithm (Pons & Latapy, 2005) to perform clustering in the reduced singular vector space $\mathbf{V}$. This graph-based clustering technique aims to identify distinct cell types from scRNA-seq data by leveraging the connectivity patterns of the data points. By constructing a graph representation where each node represents a cell and edges represent the similarity between gene expression landscape in the cells, walktrap clustering detects densely connected regions within the graph, which correspond to different cell types.

This two-step approach, separating dimension reduction and subsequent graph-based clustering, allows for a comprehensive analysis of the data. It provides a means to reduce the dimensionality of the data while capturing the underlying structure and iden-

tifying distinct cell types. By separating the dimension reduction and clustering processes, researchers have the flexibility to explore different combinations of techniques, adapting to the specific characteristics of their data and the goals of their analysis.

2.7 Biclustering Methods

Biclustering is a method to co-cluster both rows (i.e. features) and columns (i.e. conditions) at the same time. Hartigan (Hartigan, 1972) first introduced biclustering in the 1970s, but it was Cheng and Church (Cheng & Church, 2000) who first applied it to gene expression data analysis. After that, biclustering has been widely used to study the gene expression data, including DNA microarray data, bulk RNA-seq and scRNA-seq data. In this paper, the gene expression matrix always has genes as rows and conditions (for microarray and bulk RNA-seq data) or cells (for scRNA-seq data) as columns. Biclusters are groups with both genes and conditions/cells, where the genes comprise similar expression profile in the co-clustered experimental conditions or subset of cells. The aim of biclustering algorithms is to identify the block structures in data, e.g. the green blocks as shown in the heatmap in Fig. 1.16. In this section, we will explore the characteristics of biclusters and provide a brief overview of some of the existing biclustering algorithms.

2.7.1 Structure of bicluster

Bicluster structures can be categorized based on how rows and columns are grouped. Dependending on the number of genes and conditions in all the biclusters, the bicluster patterns can be categorized as following (Pontes, Giráldez, & Aguilar-Ruiz, 2015):

- **Row exhaustive.** Every gene must belong to at least one bicluster.
- **Column exhaustive.** Every condition must belong to at least one bicluster.
- **Non exhaustive.** The genes and conditions can be not assigned to any bicluster.
- **Row exclusive.** Each gene can be assigned to one bicluster at most.
- **Column exclusive.** Each condition can be assigned to one bicluster at most.

- **Non exclusive.** The biclusters can be fuzzy clusters, i.e., several biclusters can share genes and/or conditions.

2.7.2 Existing Biclustering methods

There are several existing biclustering methods developed for detecting the above mentioned bicluster structures in data. To benchmark performance of the biclustering algorithm developed in this study, I will compare it with nine well-known existing biclustering algorithms: CCA (Cheng & Church, 2000), Plaid (Lazzeroni & Owen, 2002), Xmotifs (Murali & Kasif, 2002), BiMax (Prelić et al., 2006), QUBIC (Li, Ma, Tang, Paterson, & Xu, 2009), s4vd (Sill, Kaiser, Benner, & Kopp-Schneider, 2011), Unibic (Z. Wang, Li, Robinson, & Huang, 2016), BCSpectral (Kluger, Basri, Chang, & Gerstein, 2003) and IRIS-FGM (QUBIC2) (Xie et al., 2020; Chang et al., 2021). These 9 algorithms have been selected because a) they are producible and convenient to use. b) the methods cover different biclustering methodologies. c) most of them are developed for gene expression data analysis or stated the capability to deal with multi-omics data, including DNA microarray data, bulk RNA-seq, and scRNA-seq data.

CCA. The CCA algorithm, introduced by Cheng and Church (Cheng & Church, 2000), was the first biclustering algorithm applied to gene expression data, the microarray data. This algorithm first preprocesses the data matrix by replacing missing values with random numbers. Then it initializes the bicluster as the whole matrix and proportions the matrix into non-overlapping biclusters by assessing the quality of biclusters by thresholding Mean Squared Residue (MSR) which indicates the overall within-cluster similarity of row and column items. Though the preprocessing of missing values increases the possibility of applying this method to sparse scRNA-seq data, the thresholding is data set dependent and MSR is limited to only capture shifting tendencies in the data (Bozdağ, Kumar, & Catalyurek, 2010). Besides, the iterative greedy searching slows down the speed of the algorithm. Moreover, both genes and cells are allowed to be assigned into multiple biclusters, making the interpretation of results challenging.

Plaid. The Plaid models proposed by Lazzeroni and Owen (Lazzeroni & Owen, 2002) are a series of probabilistic models, which aim to simulate expression level in each entry with a probability function. This function calculates the likelihood of the en-

try being observed in all possible biclusters. The probability function incorporates both the background information of each data and the number of biclusters. The models are optimized by minimizing the sum of squared errors between original and approximated expression levels. As a result, the estimated probabilities indicate the membership of each entry in the biclusters. The Plaid models are initially designed for microarray data analysis, they allow one gene to belong to multiple biclusters or none at all.

Xmotifs. A gene's expression level is considered conserved if it remains the same across all samples. A conserved gene expression motif refers to a group of genes that consistently exhibit the same expression pattern within a subset of samples. Xmotifs, developed by Murali et al. (Murali & Kasif, 2002), is a method to identify the conserved gene expression motifs, which are also referred to as biclusters. Xmotifs employs an iterative searching approach where genes and conditions that forms a Xmotif is removed from the data and the remaining undergo the iteraction until all samples are processed. This method is a non-exhaustive biclustering method, a gene or a cell is allowed to not be assigned to any biclusters. This method is also applied to DNA microarray data analysis when it was developed.

Bimax. Prelić et al. (Prelić et al., 2006) proposed the Bimax algorithm, which follows a divide-and-conquer approach and aims to identify the largest possible binary sub-matrix where valuable information is represented by either 1 or 0. This method was also initially developed for microarray data analysis.

QUBIC. QUaliative BIClustering algorithm (QUBIC) (Li et al., 2009) starts by transforming the values in data matrix into integers in either a qualitative or semi-qualitative manner. Subsequently, a weighted graph is constructed based on the transformed matrix, where the gene vertices are connected and weighted according to the number of columns that share the same nonzero integer for each pair of rows (genes). In essence, a bicluster is defined as a group of nodes that form a large, connected subgraph within the graph, exhibiting relatively strong edges on average compared to randomly selected subgraphs that do not intersect with such biclusters. By incorporating information from both the genes and column conditions in the graph (vertices and node weights, respectively), QUBIC is capable of simultaneously partitioning the rows and columns. It is noteworthy that QUBIC can not only detect the positively correlated genes and

conditions, but also negatively related ones. This method was also initially applied to microarray data.

s4vd. Different from the methods mentioned above, s4vd (Singular Value Decomposition-based Biclustering) (Sill et al., 2011) is a singular value decomposition (SVD) based algorithm. This approach involves decomposing the expression matrix with SVD. Subsequently, the left and right singular matrices, which corresponds to the row and column embedding respectively, are partitioned coordinately with a subsampling-based variable selection technique that controls Type I error rates. The original publication of s4vd included a benchmarking study using microarray data, but the algorithm has since been applied to various other datasets as well (Yang & Vingron, 2018). The algorithm allows one gene to be assigned into several biclusters.

Unibic. Unified Biclustering (Unibic) (Z. Wang et al., 2016) is an algorithm for biclustering that operates at the row level. The algorithm starts with generating an index matrix, where the order of genes in each row is encoded. The rows of the index matrix are then divided into subsets. Unibic applies the longest common subsequence framework to these row subsets, identifying and extracting trend-preserving biclusters from the data. The original publication of Unibic also applied the algorithm to microarray data in order to assess its performance.

BCSpectral. Spectral BiClustering algorithm (BCSpectral) is a method that relies on Singular Value Decomposition (SVD). The first step of BCSpectral involves rescaling the genes and conditions independently and in a bistochastic manner. This rescaling process ensures that the SVD of the normalized matrix increases and connects the eigenvectors of the genes and conditions. The matrix can then be partitioned using the eigenvectors corresponding to the largest eigenvalues. Next, the genes and conditions are clustered separately based on the row and column eigenvectors, respectively. Finally, the results of the row and column clustering are merged to form biclusters. Initially, this method was applied to analyze microarray data. However, it has also been adapted for scRNA-seq data analysis (Zhao et al., 2021).

IRISFGM. IRISFGM (Chang et al., 2021) is the package name associated with the biclustering algorithm QUBIC2 (Xie et al., 2020), which is a consecutive work of QUBIC. The development of QUBIC2 was motivated by the limitations of previous

Algorithms	Biclustering for scRNA-seq	One gene is assigned to only one cluster	One cell is assigned to only one cluster	Exhaustive cluster of genes	Exhaustive cluster of cells
CAbiNet	Yes	Yes	Yes	Yes/No	Yes
Bimax	No	No	No	No	No
CCA	No	No	No	No	No
QUBIC	No	No	No	No	No
IRISFGM	Yes	No	No	Yes	Yes
Plaid	No	No	No	No	No
s4vd	No	No	Yes	No	No
Unibic	No	No	No	Yes	Yes
Xmotifs	No	Yes	No	No	No

Table 2.1: **Overview of existing biclustering algorithms and CAbiNet.**

biclustering algorithms in effectively analyzing bulk RNA-seq and scRNA-seq data, where low expression levels and drop-out values are commonly observed. QUBIC2 addresses these challenges by employing a left-truncated mixture Gaussian model to assess the presence of multiple modes in expression data enriched with zero values. Additionally, QUBIC2 incorporates a dropout regression step to account for and mitigate the impact of dropouts. Moreover, it provides a robust statistical test to evaluate the significance of the obtained biclusters.

Besides, there are other algorithms designed for detecting biclusters from microarray and RNA-seq data, e.g., RecBic (Liu, Li, Liu, Su, & Li, 2020). Since most of the biclustering algorithms mentioned above are developed for analysing microarray data, they didn't take the sparsity and low expression level of scRNA-seq data into account consciously. It is critical to know if they can directly applied to scRNA-seq data analysis. In one study (Xie et al., 2020), the authors compared QUBIC2 with 8 previous published algorithms, including QUBIC, Plaid and Bimax, by using four types of data sets, namely synthetic data, microarray, bulk RNA-seq and scRNA-seq data sets. QUBIC, Plaid and Bimax were shown to be applicable to bulk and scRNA-seq data, though performing worse than QUBIC2.

Biclustering algorithms, still encounter several challenges when applied to scRNA-seq data analysis. Firstly, most existing algorithms are unable to effectively handle the issue of dropouts commonly observed in scRNA-seq data. Secondly, these algorithms exhibit limited capacity in interpreting time series data, as highlighted by QUBIC2's in-

ability to distinguish cells collected at different time points. Additionally, the growing volume of scRNA-seq data necessitates the development of more scalable algorithms. Consequently, there is a need for a sensitive and scalable biclustering algorithm to address these challenges. As a result, a novel biclustering algorithm *Correspondence Analysis based biclustering on Networks (CAbiNet)* is developed in this study and will be introduced in Section 4. A brief introduction of the characteristics of CAbiNet can be found in Table 2.1.

2.7.3 Biclustering evaluation criteria

Various metrics have been devised to assess the degree of overlap between ground truth biclusters and detected biclusters. These metrics include the Recovery score (Prelić et al., 2006), relevance score (Prelić et al., 2006), clustering error (Horta & Campello, 2014), Jaccard index (Jaccard, 1912), among others. Let's assume that b_1 and b_2 represent two biclusters.

Jaccard index. The Jaccard index (Jaccard, 1912) is employed to measure the similarity between them, and it can be computed as follows:

$$J(b_1, b_2) = \frac{|b_1 \cap b_2|}{|b_1 \cup b_2|}, \tag{2.60}$$

where $|b_1 \cap b_2|$ is the intersection of biclusters and $|b_1 \cup b_2|$ is the union of biclusters.

Recovery score and relevance score. Suppose G_1 and C_1 are the gene sets and cell sets of a bicluster b_1, G_2 and C_2 are the gene sets and cell sets of bicluster b_2, the recovery score of the similarity between these two sets of biclusters M_1 and M_2 is defined as

$$R(M_1, M_2) = \frac{1}{|M_1|} \Sigma_{b_1 \in M_1} \max_{b_2 \in M_2} ms(b_1, b_2), \tag{2.61}$$

where $ms(b_1, b_2)$ is the match score between two biclusters b_1 and b_2:

$$ms(b_1, b_2) = \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|}, \tag{2.62}$$

which is the Jaccard index between the gene sets of two biclusters. The relevance score is defined as $R(M_2, M_1)$.

As can be seen from the definition of recovery score and relevance score, the similarity of two biclustering sets are determined by the overlapping between gene sets in the biclusters. The information of cell sets is missing from the evaluation. Therefore, I modified the definition of recovery score as the average Jaccard index between two biclusters, that is

$$R(M_1, M_2) = \frac{1}{|M_1|}\Sigma_{b_1 \in M_1} \max_{b_2 \in M_2} J(b_1, b_2). \tag{2.63}$$

The recovery and relevance scores are both values in range between 0 and 1, 0 means there is nothing in common between two bicluster sets, while 1 indicates a perfect match.

Clustering error. The clustering error (CE) (Patrikainen & Meila, 2006) is defined as

$$CE(b_1, b_2) = \frac{d_{max}}{|U|}, \tag{2.64}$$

where $|U| = |b_1 \cup b_2|$ is the union of two biclusters and d_max measures how much the biclusters in two biclustering results intersect:

$$d_{max} = \max_{I_i, J_i} \Sigma_{i=1}^{min(I,J)} |b_{I_i} \cap b_{J_i}|. \tag{2.65}$$

CE ranges from 0 to 1. The larger the CE is, the more similar the detected biclusters and the ground-truth biclusters are.

2.8 Visualization Approaches

2.8.1 Linear Approaches

The most commonly employed technique for visualizing cell-gene relationships is the PCA biplot. In this method, cells are plotted in a two-dimensional space using principal components, while genes are represented by loadings that indicate their contribution to each principal component. The distance between genes or cells in the PCA biplot reflects their similarity. However, it is important to note that the distance between gene and cell points lacks meaningful interpretation since they originate from

two distinct spaces with different bases.

Another type of linear two-way visualization is the CA biplot, which has found extensive application in economic and environmental studies. The CA biplot enables the representation of cells and genes in spaces defined by principal coordinates and standard coordinates (refer to Section 2.4.3 for details on the method). In a CA biplot where cells are represented by principal coordinates and genes by standard coordinates, the Euclidean distance between cells in the lower-dimensional space approximates the χ^2 distances in the original space, similar to the property observed in the PCA biplot. Furthermore, the inner product between cells and genes in this space provides insight into their association. Specifically, a smaller angle between cell points and gene points suggests a higher likelihood of gene-specific expression in the cell. Additionally, the inner product in the lower-dimensional space approximates the Pearson residuals, which indicate the dependencies between a cell and a gene.

When compared to the PCA biplot, the CA biplot provides a more informative representation of the relationship between cells and genes. However, both biplots suffer from certain limitations. Firstly, for a data set like the nowadays scRNA-seq data which is sparse and noisy, the first two dimensions of the biplots only capture a small fraction of the total variance (PCA) or inertia (CA), which is around 10% to 50% (Luecken & Theis, 2019; Northcutt et al., 2019; Slovin et al., 2021). Consequently, the differences between clusters may not be fully visible in the 2D space, and examining variations among clusters in higher dimensions requires adjusting the biplot axes to the relevant dimensions of interest. Secondly, as human spatial perception is limited to 3D space, linear embedding approaches fail to provide a comprehensive understanding of the heterogeneity of data.

2.8.2 Nonlinear Approaches

Based on the limitations of linear visualization approaches mentioned above, a non-linear two-way embedding technique is needed to address these limitations and enhance our comprehension of cell-gene relationships. t-distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten & Hinton, 2008) is a widely used nonlinear dimension reduction technique, often used to visualise high-dimensional data in

two or three dimensions. Uniform Manifold Approximation and Projection (UMAP) (McInnes, Healy, & Melville, 2020) is another nonlinear dimension reduction technique which was first introduced in 2018 as an alternative to t-SNE.

t-SNE constructs a probability distribution over pairs of high-dimensional objects and a similar distribution over pairs of low-dimensional objects. It then defines conditional probabilities that measure the similarity between pairs of points. A random initialization of the low dimensional embedding of points is generated. Furthermore, it minimizes the divergence between the conditional probabilities in high dimensional space and low dimensional space by using a gradient descent method. This procedure ensures the lower dimensional representations approximate the distribution of the higher dimensional data. t-SNE is particularly effective at preserving the local structure of the data, but can be computationally expensive and may require careful tuning of its hyperparameters.

UMAP operates by constructing a graph representation of the high-dimensional data, where each data point is connected to its nearest neighbors. The graph embedding is then optimised using a process called stochastic gradient descent with eigenvectors of the normalized graph Laplacian as initialized embedding. UMAP seeks to minimise a cost function that balances the preservation of both local and global structure in the data. The result is a low-dimensional representation of the data that captures the underlying structure of the original high-dimensional space.

UMAP offers several advantages, including speed. It is capable of efficiently processing large datasets containing millions of data points and high-dimensional feature spaces. UMAP can generate high quality visualisation of a scRNA-seq data set with 20,921 cells in about 100 seconds (McInnes et al., 2020). UMAP is also designed to handle missing data, making it a versatile tool for a variety of data analysis scenarios.

UMAP claims to provide users with valuable features such as the ability to adjust the balance between preserving local and global structure in the data. However, the authors also mentioned some limitations of UMAP. Performance of UMAP can be sensitive to the hyperparameters, especially the number of nearest neighbours for the kNN graph. Besides, UMAP is not as interpretable as linear embedding approaches, such as PCA and CA, there is no specific meaning of the axes, while the first principal

component in PCA retains the most variances. The authors also claimed that UMAP tends to preserve topology instead of preserving the pure metric distances. This makes UMAP perform badly when evaluated by metric measurements, e.g. multi-dimensional scaling (MDS) (Kruskal, 1964)

However, an increasing number of biological studies are making overstated conclusions out of the UMAPs they have generated. In response, a recent paper by (Chari & Pachter, 2023) examined the distance metrics in UMAP 2D embeddings to assess their reliability. This study employed L1 and L2 (Euclidean distance) metrics to investigate distance distortions in UMAP. The findings reveal that UMAP distorts both local and global data structures according to both L1 and L2 metrics. This investigation suggests that some current applications of UMAP in biological data analysis may pose problems. For instance, the trajectory inference of single-cell RNA sequencing data from developmental tissues relies on the layout of 2D UMAP embeddings. Certain tools (Wolf et al., 2019) determine the cellular developmental trajectory based on UMAP visualization with field vectors. However, because the arrangement of cells distorts the data structure in the ambient space, the inferred trajectory may appear discontinuous, and the direction of the vector field could be misinterpreted (Chari & Pachter, 2023). Furthermore, biological conclusions drawn from UMAP embeddings may be unreliable; for example, two cell types appearing close in UMAP are not necessarily similar. Additionally, UMAP is often used to evaluate algorithm performance, such as trajectory inference algorithms (Saelens, Cannoodt, Todorov, & Saeys, 2019) and data integration algorithms (Hao, Hao, Andersen-Nissen, et al., 2021), potentially leading to erroneous conclusions (Chari & Pachter, 2023).

Despite the truth that both t-SNE and UMAP have been applied to challenging problems, including image processing, natural language processing, social science and biological data analysis. They provide a comprehensive representation of the similarities and dissimilarities among samples. Researcher still have to be careful when they draw conclusions from the data, especially the UMAP application scenarios mentioned above.

A common limitation of both t-SNE and UMAP is that neither of them is capable of embedding genes and cells simultaneously in lower dimensional space. Both of them

can only visualize genes or cells separately. A joint embedding of genes and cells is needed.

Single-cell embedding along with features (SIMBA) is developed to visualize both genes and cells in a non-linear embedding space. Moreover, it allows integration of multi-modalities, e.g. scRNA-seq, scATAC-seq, histone modifications and so on. scRNA-seq creates a graph that represents cells and features from multi-modalities as nodes, and their relationships are encoded as edges. The graph is then embedded into a low-dimensional space using a multi-entity graph embedding algorithm, which is similar to techniques used in social networking. scRNA-seq also uses a Softmax-based transformation to help analyze the cells and features based on their distance in this low-dimensional space. This allows for a more comprehensive analysis of the cells and features and their relationships. Even though applying community detection algorithms to the graph constructed by scRNA-seq naturally gives co-clustering of cells from input modalities and features. However, scRNA-seq doesn't provide a function to cluster the cells with features and its performance is not evaluated either. Therefore, an approach that can bicluster and jointly embed the cells and features in a non-linear embedding is still needed. This motivates the creation of CAbiNet.

2.9 Gene Module Detection Methods

Besides the application on classifying the cell types to study the heterogeneity of cells, clustering algorithms are also applied to gene expression data to group genes into co-expression modules using transcriptomic data (Yosef et al., 2013; Jojic et al., 2013; Paul et al., 2015; Alsina et al., 2014; Eren, Deveci, Küçüktunç, & Çatalyürek, 2013; Marbach et al., 2012; Roy et al., 2013; Rotival et al., 2011). The detected gene modules are supposed to be genes co-expressed or share similar expression patterns among samples/cells. Different gene modules usually represent different functional pathways. Knowing the gene modules helps to understand which biological pathways or regulatory networks are activated under the observed experimental conditions or cell types interested in.

There have been many algorithms developed for detecting the gene modules in

microarray data, bulk RNA and scRNA-seq data. The methods can be classified into mainly four categories: clustering methods (Yosef et al., 2013; Jojic et al., 2013; Paul et al., 2015; Alsina et al., 2014), biclustering methods (Eren et al., 2013), graph based methods (Marbach et al., 2012; Roy et al., 2013) and decomposition methods (Rotival et al., 2011).

Instead of clustering the cells, the clustering methods applied to gene module detection work on the genes and group them into clusters. Clustering methods like K-means, hierarchical clustering and density based clustering algorithm (DBSCAN)(Ester, Kriegel, Sander, Xu, et al., 1996) partition the genes by the similarity of genes' expression along samples/cells. The similarity is normally measured by Euclidean distance. These methods are shown to perform well on detecting co-expressed genes in some cases(Yosef et al., 2013; Jojic et al., 2013; Paul et al., 2015; Alsina et al., 2014).

However, the clustering methods suffer from some issues. Firstly, the clustering results may be influenced by noise or outliers in the data. Secondly,the pattern of gene expression may not be shared among all the samples, and some gene modules may be presented in a subset of the data only. Thirdly, one gene can be co-expressed with many genes or more than one gene modules. However, the hard clustering methods can only divide one gene into a single cluster.

To overcome the problem of hard clustering methods, fuzzy clustering methods (Fu & Medico, 2007), like the fuzzy c-means, are applied to allow overlapping between gene modules. Then, one gene can be assigned to multiple modules simultaneously.

To solve the other limitation of clustering methods, decomposition methods have been developed to do dimension reduction to remove noise, e.g. PCA and ICA (Rotival et al., 2011). By decomposition methods, the original count table is transformed into a new space where the components are a linear combination of the unit vector of original space. The first several dimensions in the new space conserve most of the variations in the data. Reducing the space into the first several components, noise in data can be reduced, while still preserving the heterogeneity. Then clustering methods are applied to the dimension reduced space to find the gene modules.

The biclustering algorithms are developed to detect local co-expression patterns of genes. As mentioned in Section 2.7.1, the detected biclusters will not necessarily

cover all the genes or the cells, meaning that the biclustering can define local gene-sample/cell clusters, which naturally solves the issue of clustering methods fail to detect local structures. Some biclustering algorithms also allow for assigning one gene/cell into more than one bicluster, such that the comprehensive gene co-expression patterns can be recovered.

The network based algorithms, for example, the direct Network Inference method (Marbach et al., 2012), are developed to infer gene-gene regulatory relationships.

Among all the methods mentioned above, the decomposition methods are shown to be the best performing gene module detection methods (Saelens, Cannoodt, & Saeys, 2018). Wouter Saelens, et. al. benchmarked some existing gene module detection methods covering all four categories with both synthetic data and experimental data sets. The detected gene modules of each algorithm are evaluated by comparing with the gene regulatory networks. It has been shown that the decomposition methods work best at recovering known modules consistently across data sets. The decomposition methods are also capable of detecting overlapping and local co-expressed gene modules, while the clustering based gene detection methods failed to do that.

3 | Clustering in Correspondence Analysis space

As outlined in the previous section (Section 2.6), CA serves to reduce the dimension of large datasets for noise removal. Subsequently, clustering algorithms can be employed on these dimension reduced spaces to segment the data into clusters. However, there exist three types of dimension reduced CA spaces: those featuring principal coordinates, standard coordinates, and singular vectors (Section 2.4.3). When conducting clustering in a CA space, it becomes crucial to determine which coordinate system is best suited for the clustering process. In this chapter, I will first introduce the datasets that will be used for illustration and testing in Section 3.1. The procedures utilized for data preprocessing will then be elaborated upon in Section 3.2. Following this, I will introduce the application of CA on preprocessed simulated data in Section 3.3. A comprehensive examination of distance measurements in symmetric and asymmetric CA spaces will also be presented in Section 3.3. Finally, I will deliberate on the selection of the appropriate coordinate system for clustering within the dimension reduced space in Section 3.4.

3.1 Data sets

In this study, three simulated scRNA-seq data sets with different sizes and bicluster structures and six experimental scRNA-seq data sets are used to evaluate and benchmark the algorithms.

The simulated scRNA-seq data sets are generated by R package *SPARSim*. The function learns the variation among genes and cells from an experimental single-cell RNA-seq data *PBMC*, and then the parameters are used to generate three simulated data sets. The *PBMC* data that has been used as template is downloaded from R package *ExperimentHub* with *ExperimentHub_ID* as *EH5407*. There are 3312 cells with 22735 genes being detected in this data set.

Three data sets are generated with different parameters. For all the simulated data sets, four cell types with five gene modules are simulated in the simulated data. There are 500 cells in each cell type, 2000 cells in total. Four out of five gene modules contain genes that are highly specifically expressed in four corresponding cell types, while

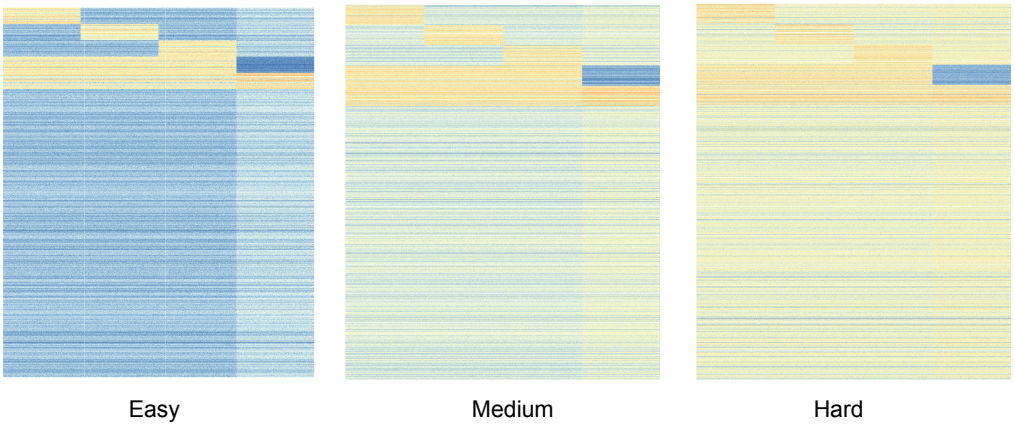

Figure 3.1: **Simulated scRNA-seq data sets.** Three simulated data sets are designed for this study, the data sets are named as "easy", "medium" and "hard" according to the level of noised in data ranging from the least to the most.

genes in the other gene module are lowly expressed in one of the cell types. There are 1000 genes in each of the four gene modules, and 18735 genes in remaining module. For all the simulated data sets, the variance among cell types is the same, while mean value of designed gene modules varies from smallest to largest. The data set with largest mean value is supposed to be the one that biclustering algorithms can easily recognize biclusters in it. Figure 3.1 visualizes the simulated data sets.

The information on the utilized experimental scRNA-seq data sets are listed in Table 3.1. The data sets were downloaded from the citation listed in the table. For the listed data sets, we have the information of the cell types of cells, which are annotated by researchers who published the data. The cell-type annotations will be used as ground truth of the cell clusters and be used for benchmarking the algorithms. All the data sets listed in the table are single-cell transcriptomic data, but are generated with various techniques, including SMART-seq2, 10x, Fluidigm CI, CEL-seq, Stereo-seq. Readouts from different sequencing techniques varies in sequencing depth, gene coverage and dropout rates. Besides, the data sets are selected to have various sizes, with number of genes ranging from 13,000 to 60,000 and number of cells ranging from about 500 to 35,000. The variation of sequencing techniques and data sizes allow us to have a comprehensive evaluation of the algorithms applied.

Short Name	Dataset description	# Cells	# Genes	Protocol	Ref.
Darmanis	Human adult cortical samples	466	22,085	SMART-Seq2	(Darmanis et al., 2015)
FreytagGold	Three human lung adenocarcinoma cell lines, HCC827, H1975 and H2228	925	58,302	10x	(Freytag, Tian, Lönnstedt, Ng, & Bahlo, 2018)
zeisel	Mouse somatosensory cortex and hippocampal CA1 region (ZeiselBrain)	2,874	14,508	Fluidigm C1	(Zeisel et al., 2015)
pbmc3k	Human peripheral blood mononuclear cells	2,700	32,738	10x	(10x Genomics, 2016)
Tirosh	Human melanoma tumor nonmaglignant cells	2,887	23,686	SMART-Seq2	(Tirosh et al., 2016)
PBMC10x	Human peripheral blood mononuclear cells (FACS sorted)	3,362	33,694	10x	(Ding et al., 2020a)
BaronPancreas	Human Pancreas	8,569	20,125	CEL-seq	(Baron et al., 2016)
DemlSpatial	Drosophila melanogaster late stage embryo (14-16h after egg laying)	15,295	13,668	Stereo-seq	(M. Wang et al., 2022)
TabulaSapiens	Human endothelial cells	32,701	58,559	10x	(THE TABULA SAPIENS CONSORTIUM, 2022)
BrainOrganoids	Human cerebral organoids	35,291	33,538	10x	(Rosebrock et al., 2022)

Table 3.1: **Experimental scRNA-seq data sets with expert annotation of cell clusters.** These data sets are used for benchmarking of the biclustering algorithms and illustration of the algorithms developed in this paper.

3.2 Data Preprocessing

Correspondence analysis is sensitive to outliers, so it is important to detect and remove the outliers before applying correspondence analysis to the data (Langovaya, Kuhnt, & Chouikha, 2012). We firstly processed real and simulated scRNA-seq data by removing cells with too low coverage or too few detected genes by sequencing techniques. This step not only speeds up the computation but also allows correspondence analysis to represent data in a more reasonable embedding. Outlier cells were filtered with the functions `perCellQCMetrics` and `perCellQCFilters` from the Bioconductor tool `scuttle` (McCarthy, Campbell, Lun, & Wills, 2017).

We also applied a filtering step of genes to eliminate redundant genes. This step further reduces the size of data and improve downstream analysis efficiency. The selection of highly variable genes was performed by fitting a trend to the variance of log counts for all genes with respect to the mean expression. This process was carried out using the `modelGeneVar` function from the `scran` package. In addition, we removed genes that were expressed in less than 1% of all cells.

Next, the filtered matrix can undergo the application of CA to all remaining genes and cells. Further down-sizing of data can be achieved by retaining only the genes ranked as the top variable genes, the number of the most variable genes varies among 2000, 4000 and 6000, depending on the data and parameter choice. It is important to note that even though the original count matrix is normalized and subsetted to a matrix with most variable genes, it still maintains sparsity. Table 3.2 presents the sparsity of the filtered matrix in the first column, followed by the sparsity of the subset matrices, which include the matrix with the top 2000, 4000 and 6000 most variable genes in the second, third and fourth columns correspondingly. The sparsity is calculated as the ratio between number of zeros divided by the number of elements in the matrix. The sparsity of all the experimental datasets ranges from approximately 55% to 95%, with the exception of the 'FreytagGold' dataset, which exhibits denser characteristics compared to the others.

It has been reported previously that log-transformation can help to reduce overdispersion in RNA-seq data when applying PCA for dimension reduction. It is also re-

Dataset	Sparsity	Sparsity (2000)	Sparsity (4000)	Sparsity (6000)
BaronPancreas_filtered	0.847	0.762	0.834	0.871
Darmanis_filtered	0.758	0.576	0.65	0.7
FreytagGold_filtered	0.575	0.237	0.323	0.461
PBMC_10X_filtered	0.903	0.822	0.879	0.904
Tirosh_nonmaglignant_filtered	0.763	0.636	0.709	0.764
ZeiselBrain_filtered	0.735	0.605	0.679	0.732
brain_organoids_filtered	0.93	0.876	0.916	0.933
tabula_muris_sub	0.829	0.693	0.784	0.833
tabula_sapiens_tissue_filtered	0.822	0.665	0.764	0.815

Table 3.2: **Sparsity of experimental scRNA-seq data sets.**

ported that applying log-transformation prior to applying PCA to count tables in scRNA-seq analysis can help to mitigate the artifacts generated by applying PCA directly to the count table (Nguyen & Holmes, 2019; Hsu & Culhane, 2020, 2023b). Because PCA is most suitable for continuous data that is approximately normally distributed, it may exhibit artifacts when applied to data with gradients or non-continuous count table. However, there is an argument that normalizing with log-transformation distorts the distribution of data. The scRNA-seq gene expression levels are assumed to follow negative binomial distribution or Poisson distribution, the log-transformation makes the data not follow the distributions any more. This will influence the down-stream statistical tests (Lause, Berens, & Kobak, 2021; Townes et al., 2019). Furthermore, opinions differ on whether log-transformation should be applied before conducting CA on the data. One study (Langovaya et al., 2012) suggested that log-transformation can help eliminate the influence of outliers on correspondence analysis, while the other study (Hsu & Culhane, 2023a) recommended against log-transformation prior to applying CA.

Since the preprocessing of data is to prepare it for CAbiNet algorithm to find out the biclusters of cells and genes, the impact of log-transformation will be evaluated based on the biclustering accuracy of CAbiNet. The CAbiNet algorithm will be introduced in the next chapter. Data sets listed in Table 3.1 will be used for testing. These data sets were pre-processed following the methods described in the first two paragraphs of Section 3.2. Then the matrices are subsetted into count matrix with top 2,000, 4,000

and 6,000 most variable genes. Subsequently, subsets of matrices containing either the original counts or log-transformed data were used as inputs for CA. The CA process follows the routine procedures: firstly, Pearson Residuals were calculated. Secondly, the matrix with Pearson Residuals was subjected to singular value decomposition to achieve dimension reduction to a 40, 60 and 80 dimensions, separately. Then CAbiNet was applied to the dimension reduced data to build cell-gene graph with the number of nearest neighbours varies from 30, 60 to 90. Both leiden clustering and spectral clustering are utilized to identify the biclusters. The resolution parameter of leiden clustering are set as 0.1, 0.8 and 1. The number of clusters for spectral clustering was set as the number of ground-truth clusters for each data set.

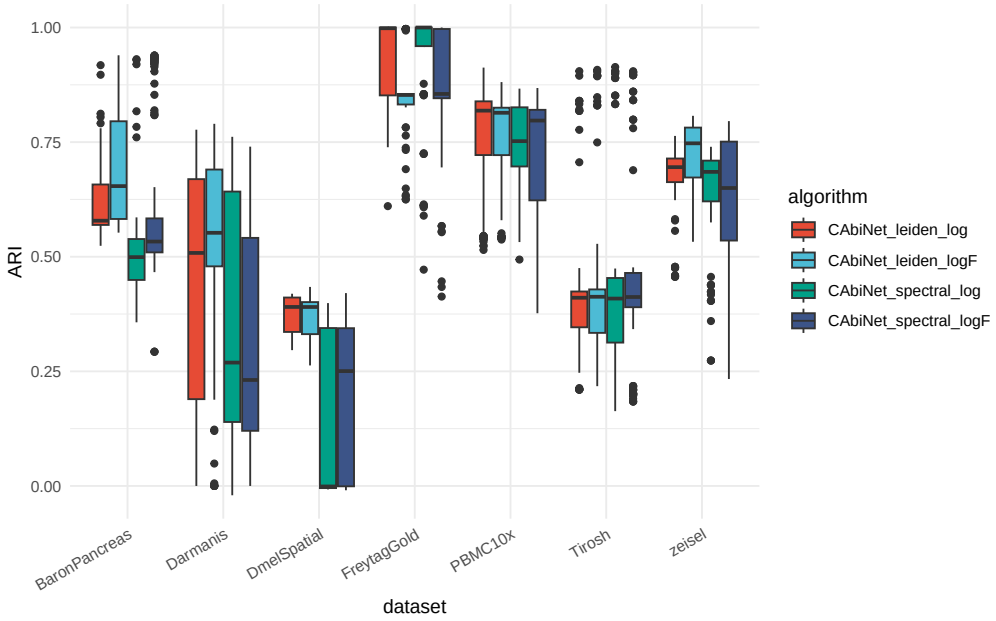

Figure 3.2: **Evaluation of log-transformation effect on clustering of real scRNA-seq data sets with silver standard ground truth.** The boxplot shows the accuracy of clustering results of CAbiNet with leiden algorithm applied on data sets with log-normalization (CAbiNet_leiden_log) and without log-normalization (CAbiNet_leiden_logF). Similarly, the ARI of clusters detected by CAbiNet with spectral clustering on data sets with and without log-normalization are plotted with labels: CAbiNet_spectral_log and CAbiNet_spectral_logF.

The comparison results are depicted in Fig. 3.2. The the accuracy of clustering results ARI are grouped into the following groups: CAbiNet with leiden algorithm applied on data sets with log-normalization (CAbiNet_leiden_log) and without log-

normalization (CAbiNet_leiden_logF), CAbiNet with spectral clustering on data sets with and without log-normalization are plotted with labels: CAbiNet_spectral_log and CAbiNet_spectral_logF. There are 54 runs for each group. The boxplot as shown in Fig. 3.2 summarises the overall distribution of ARIs over all parameter choices. Among the data sets that have been tested, CAbiNet demonstrates superior performance when the original counts are utilized as input for two of the data sets (BaronPancreas and Darmanis). Conversely, for two out of the seven data sets, employing log-transformed counts yields slightly improved biclustering results (FreytagGold and PBMC10x). In the remaining three data sets, the original and log-transformed counts yield similar clustering accuracy.

A recently published paper (Ahlmann-Eltze & Huber, 2023) made a comprehensive evaluation of the effect of normalization methods on scRNA-seq analysis. They benchmark the influence of four types of normalization algorithms including delta method-based variance-stabilizing transformations, residuals-based variance-stabilizing transformations, latent gene expr-ession-based transformations and count-based factor analysis models. They show that the log-normalization methods are the best performing one for scRNA-seq data analysis comparing with the other three types of methods, and it is better than using the raw counts as well.

Based on this study and considering that CA is sensitive to outliers and log-transformation can moderate the influence of outliers and help with the mean-variance dispersion, we employ log-transformed counts as the input for both CA and CAbiNet during the application and benchmarking of CAbiNet.

3.3 Understanding different coordinates

To provide a clearer explanation of correspondence analysis, a demonstration data set is used to showcase the process and biplots. The data set was synthesized with R package *SPARSim*. It consists of 5000 rows representing simulated genes and 2000 columns representing simulated cells. The genes are divided into five groups, with each group containing 1000 genes, and the cells are evenly grouped into four clusters, with 500 cells in each cluster. Additionally, specific dysregulated marker genes, exhibiting

both upregulation and downregulation, have been designed for each of the four cell types. Fig. 3.3A shows the simulated data sets with genes as rows and cells as columns in the heatmap. The brighter the color is, the higher the gene is expressed in the cells.

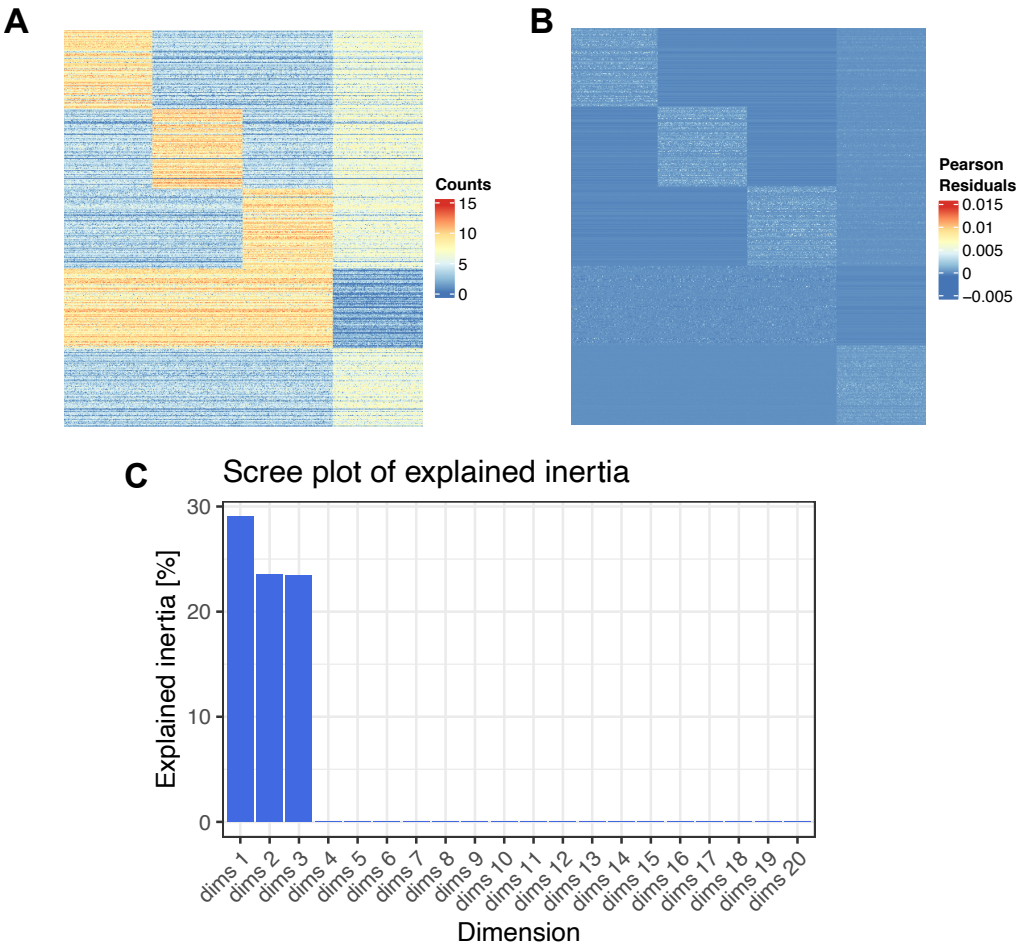

Figure 3.3: **Correspondence Analysis of a simulated scRNA-seq data.** **A**, The log-transformed simulated scRNA-seq count matrix. **B**, Then the correspondence analysis calculates a matrix of Pearson Residuals. **C**, Scree plot visualizes the percentage of inertia per dimension. The first three dimensions contains 76.05% of the total inertia and there is a sharp decrease from the third dimension to fourth dimension suggesting the CA space can be reduced into first three dimensions.

By applying correspondence analysis to the data set which is visualized in Fig. 3.3A, we first transform it into a matrix using Pearson residuals. Then, we perform singular value decomposition (SVD) on the matrix using Equation 2.13. The columns in the matrices $\mathbf{U}$ and $\mathbf{V}$ represent unit bases of new spaces for row and column items, re-

spectively, known as the CA space. Each dimension's inertia corresponds to the square of its singular value. Figure 3.3C illustrates the percentage of inertia for each dimension. The first three dimensions account for more than 76.05% of the total inertia, and there is a sharp decline between the third and fourth dimensions. This suggests that the majority of the information of the data is captured by the first three dimensions, while the remaining dimensions largely contain uninformative noise. By employing the dimension reduction techniques discussed in Section 2.2.2, the dimension of the CA space can be reduced to 3. Hereby, the dimension-reduced space can be used for data visualization and clustering.

The CA biplots of the simulated data are shown as Fig. 3.4, in which the row (genes) are plotted as blue circles and column items (cells) are plotted as red crosses.

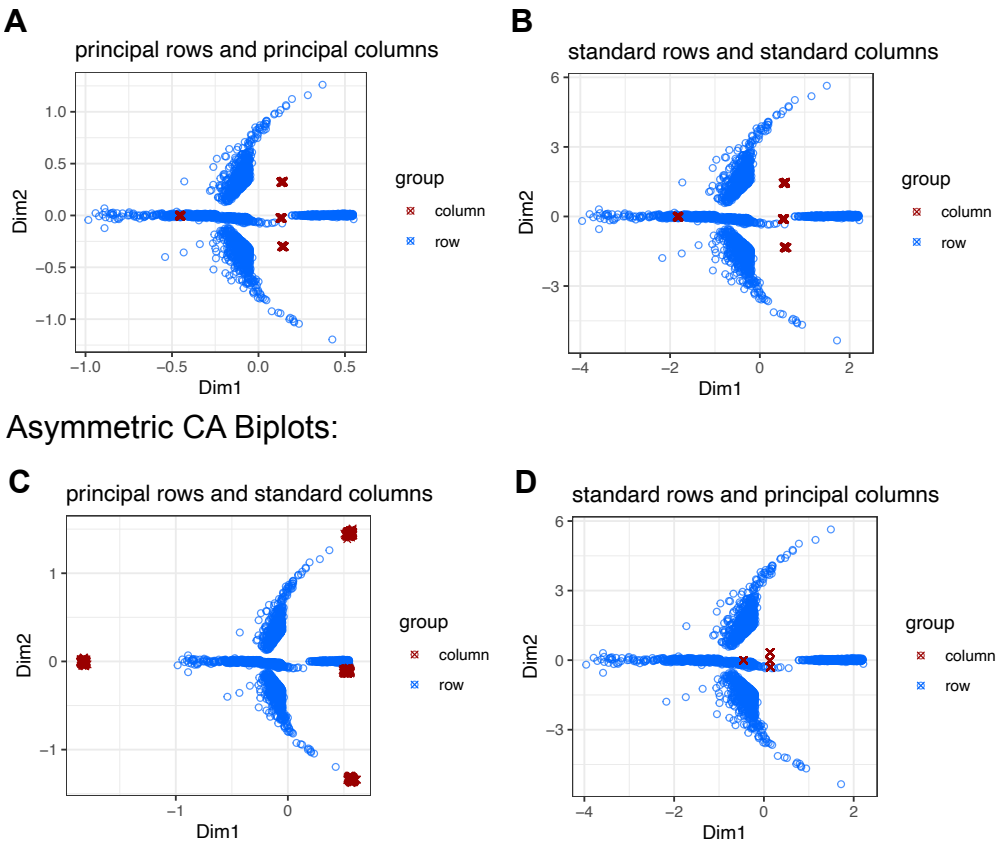

Figure 3.4: **Correspondence Analysis of a simulated scRNA-seq data. A-B**, The symmetric biplots. Both genes and cells are plotted with principal coordinates in panel A, while both of them are drawn with **C-D**, The asymmetric biplots.

In the symmetric biplots, both the genes and cells are plotted with principal coordinates in Fig. 3.4A, while with standard coordinates in Fig. 3.4B. The asymmetric biplots are shown as Fig. 3.4C-D, in which the genes and cells are plotted in different types of coordinates. The genes are in principal coordinates and cells in standard coordinates in Fig. 3.4C, while the other way around in Fig. 3.4D.

Based on Equation 2.17 and 2.16, the standard coordinates can also be interpreted as the principal coordinates divided by their corresponding singular values. Suppose the first k singular values of a matrix happen to be equal to 1, then the re-scaling would make the first k standard coordinates identical to the principal coordinates. In some cases where the singular values of chosen dimension are similar to each other, i.e. the standard coordinates are almost proportionally re-scaled along principal axes, the Euclidean distance between standard coordinates then becomes informative and more applicable than that in normal cases.

However, in reality, these scenarios rarely occur, so the re-scaling will generally cause the standard coordinates to deviate from the principal coordinates. Consequently, calculating cosine or Euclidean distances between items using principal coordinates versus standard coordinates will yield different results.

This study puts effort into understanding the distance measurements in standard and principal coordinated spaces, because it is important for the clustering to use a good distance measure. A proper distance measure is the prerequisite for getting a good clustering result. I will give a more detailed illustration of the distance measured in principal coordinates and in standard coordinates in Section 3.3.1 and 3.3.2.

The biplots also allow an interactive intepretation of the relationship between rows and columns. In symmetric biplots (Fig. 3.4A-B), it may appear that certain gene points are located close to cell points, leading to the misconception that the row features contribute to distinguishing neighboring column categories. However, this interpretation is incorrect because the genes and cells are plotted in two separate spaces with different bases. As a result, the Euclidean distance between gene and cell points lacks a theoretical meaning in this context.

In the asymmetric biplots (Fig. 3.4C-D), genes and cells pointing in the same direction indicate a strong positive association, while those pointing in opposite directions

are negatively associated. The association between a gene and a cell is represented by the angle between the vectors connecting the origin to the gene and cell points in the asymmetric biplot. A smaller angle indicates a stronger association between them. The theoretical explanation can be found in Section 3.3.4.

3.3.1 Distance measured in principal coordinates

Recall the definition of χ^2 distance in Section 2.4.4 between two rows (Equation 2.21) and the reconstitution formula (Equation 2.30 and 2.31), I substitute p_{ij} in the Equation 2.21 by the reconstitution function and prove that the χ^2 distance of two rows $\chi^2(i, i')$ in the original data can be written as the Euclidean distance among them in principal coordinates in the ambient space, $\sum_{k=1}^{K}(f_{ik} - f_{i'k})^2$, where K is the rank of input matrix. That is

$$
\begin{aligned}
\chi^2(i,i') &= \sum_{j}^{J} \frac{(\frac{p_{ij}}{r_i} - \frac{p_{i'j}}{r_{i'}})^2}{c_j} \\
&= \sum_{j}^{J} \frac{(\frac{r_i c_j \left(1+\sum_{k=1}^{K} f_{ik}\gamma_{jk}\right)}{r_i} - \frac{r_i c_j \left(1+\sum_{k=1}^{K} f_{i'k}\gamma_{jk}\right)}{r_{i'}})^2}{c_j} \\
&= \sum_{j}^{J} c_j (\sum_{k=1}^{K} f_{ik}\gamma_{jk} - \sum_{k=1}^{K} f_{i'k}\gamma_{jk})^2 \\
&= \sum_{j}^{J} c_j (\sum_{k=1}^{K} \gamma_{jk}(f_{ik} - f_{i'k}))^2 \\
&= \sum_{j}^{J} c_j (\sum_{k=1}^{K} c_j^{-1} u_{jk}(f_{ik} - f_{i'k}))^2 \\
&= \sum_{k=1}^{K} (f_{ik} - f_{i'k})^2.
\end{aligned}
\tag{3.1}
$$

where f_{ik} is the principal coordinate of entry in row i and column j (see also the Equation 2.16). This formula states that the Euclidean distance between rows in principal coordinates is meaningful, it recovers the χ^2 distance between rows in the frequency table $\mathbf{P}$. This is described by Fig. 3.4A, 3.4C, where the genes are plotted as blue circles with principal coordinates and cells as red crosses.

Suppose K' dimensions are retained in the dimensional reduced space, Equation

3.1 can be written as

$$\chi^2(i, i') = \sum_{k=1}^{K'} (f_{ik} - f_{i'k})^2 + e_{ii'}, \tag{3.2}$$

where $e_{ii'}$ is the error term representing the difference of distance between item i and i' in ambient space and dimensional reduced space.

Similarly, χ^2 distance between two columns (as shown as red crosses in Fig. 3.4A, D) can be approximated by the Euclidean distance between their principal column coordinates in a dimension reduced space, that is

$$\chi^2(j, j') = \sum_{k=1}^{K'} (g_{jk} - g_{j'k})^2 + e_{jj'}.$$

In conclusion, in this section we proved that the Euclidean distances between rows or columns in the dimension reduced CA space using principal coordinates hold meaningful interpretations. These distances reflect the χ^2-distance between rows or columns in the frequency table, bridging the original space and the CA-transformed space.

3.3.2 Distance measured in standard coordinates

As defined in Equation 2.15 and 2.14, the standard coordinates are the singular vectors divided by square root of row/column masses. In this way, the mean of row/column standard coordinates gets normalized to 1. Therefore, comparing with principal coordinates, the standard coordinates lose the explanation of the weights of items. This rescaling changes not only the vector length of items in a biplot, but also the direction of vectors. For example, in Fig. 3.4A v.s. C where the genes are plotted with the same coordinates while cells are in different coordinates. The vectors linking the origin and cell points are in different lengths and directions.

Comparing Fig. 3.4A to Fig. 3.4C, in both of which the genes are plotted with principal coordinates but cells are plotted differently, it is visible that the standard coordinates shift the column (cells) items away from the origin. The dissimilarities among cells are more apparent in standard coordinates comparing with principal coordinates. Therefore, if one is particularly interested in observing the cells, it is preferable to plot them with standard coordinates and the genes with principal coordinates. This arrange-

ment allows for an exaggeration of the differences between cells. Similarly, if the focus is on the variation among genes, an asymmetric biplot can be drawn with the opposite configuration.

3.3.3 Distance measured in singular vectors

In the dimension reduced space, the rows and columns can also be presented with coordinates in singular vectors. Since the singular vectors are orthogonal unit vectors and the singular values of each direction are the inertia preserved by each direction, the inertia of each singular vector is standardized as 1. As mentioned in the previous chapter (Section 2.6.3), the distance of points coordinated in singular vectors is used to measure the dissimilarity between points to build a graph to cluster the points (Hsu & Culhane, 2023a). Different from the distance measured in principal and standard coordinates, points that are close to each other represent categories that have a similar profile in terms of their distribution across the variables analyzed. This implies that these categories have a similar pattern of association with the variables.

3.3.4 Distance in Asymmetric Maps

As illustrated in Section 2.4.3, the asymmetric map refers to a biplot with either principal coordinates of rows and standard coordinates of columns, or principal coordinates of columns and standard coordinates of rows. Figure 3.5 shows the first scenario, where the rows/genes are plotted with blue circles with principal coordinates and columns/cells are plotted as red crosses with standard coordinates. The Euclidean distance between principal coordinates in an asymmetric map still approximates the χ^2-distance between data points in the frequency table, while the Euclidean distance between standard coordinates are meaningless when the corresponding singular values are not equal to 1.

Besides representing the correlation among row or column items, asymmetric map further tells the association between row and column points. The inner product between row item i and column item j in the asymmetric map reconstructs the deviation of the probability p_{ij} from the expectation, which is explained by the reconstruction formula

2.30. That is

$$\frac{p_{ij} - r_i c_j}{r_i c_j} = \sum_{k=1}^{K'} f_{ik}\gamma_{jk} + \epsilon_{ij}. \tag{3.3}$$

The left-hand side of the equation can be interpreted as the extent to which the observed frequency deviates from the expected frequency, denoted as $r_i c_j$. A deviation of 0 signifies that the probability of a gene (row) i expressing in a cell (column) j is equal to the expectation. A higher deviation indicates that the gene's expression level is higher than expected. In this case, the gene is more likely to be considered a marker gene for the cell compared to other genes. **This term on the left-hand side of the reconstitution formula 3.3 is commonly referred to as the association ratio. It gets approximated by the inner product between gene point and cell point in the asymmetric biplot.** The closer the ratios are to 0, the more independent the items are. Looking at Fig. 3.5, the inner product between gene point P1 and cell point C1 is larger than that between gene point P2 and C1, meaning that P1 is more likely to be a marker gene of cell C1, while P2 and C1 are more likely to be independent due to the inner product between them tends to be 0.

Although the row and column points are simultaneously plotted in the biplot, their coordinates are actually based on different basic unit vectors of different spaces. Therefore, the Euclidean distance between a row and a column item in a biplot doesn't make sense. Instead, the association between row and column items should be measured by the inner product between the points.

3.3.5 Numerical experiments on the choice of coordinates and distance measurements

Dimension reduction in CA space usually is followed by clustering of the data, usually clustering of the row items or column items. As mentioned in Section 2.6, the existing clustering algorithms in CA space group clusters based on the similarities measured in principal coordinates, standard coordinates or singular vectors. These studies don't explicitly illustrate the reason of the distance choices. Based on the illustrations in Section 3.3.1-3.3.3, I will discuss about which distance measurement is the best for clustering rows or columns in the dimension reduced CA space.

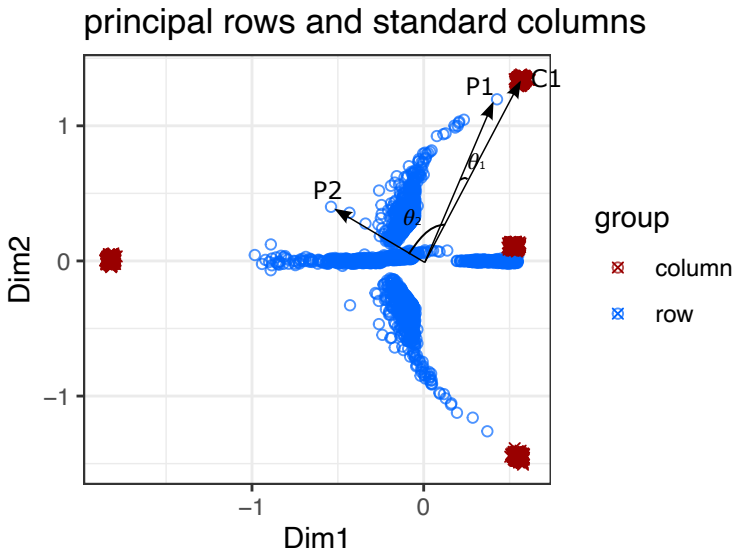

Figure 3.5: **The association between row and column items in a CA asymmetric map.** This is Fig. 3.4C annotated with points and vectors P1, P2 and C1. In this plot, the row points are plotted as blue circles with principal coordinates while columns are plotted as red crosses with standard coordinates. The inner product between row point P1 and cell point C1 in an asymmetric biplot indicates the association between two points. The larger the inner product is, the higher two points are associated. Comparing with row point P2, P1 is more associated with C1.

In the previous sections, I showed that the Euclidean distances between rows or columns in the dimension-reduced CA space using principal coordinates hold meaningful interpretations. These distances reflect the χ^2-distance between rows or columns in the frequency table, bridging the original space and the CA-transformed space. On the other hand, the standard coordinates and singular vectors do not have a clear explanation as the principal coordinates do. **This suggests that the Euclidean distances in the principal coordinates space are preferable for identifying clusters of rows and columns**.

To validate the claim regarding the choice of coordinates, I conducted an experiment using the simulated data set described in Section 3.1. The objective was to compare the efficiency of using three different dimension-reduced spaces (i.e., spaces with principal coordinates, standard coordinates, and singular vectors) in accurately recovering the ground-truth clusters of cells within the data. Evaluation was performed by computing the Silhouette score (Rousseeuw, 1987) for each item. The Silhouette score

measures the similarity among items within the same cluster and dissimilarity between items from different clusters (see Section 2.5.5). The silhouette score is calculated for each data point with range from -1 to 1. A higher Silhouette score indicates that an item is closer to other items in the same cluster than items in other clusters. Conversely, a negative Silhouette score suggests that an item has been assigned to an incorrect cluster.

To mitigate potential bias introduced by clustering algorithms, no clustering was performed in this analysis. Instead, the Silhouette score was directly computed using the known ground-truth clusters of items. Initially, the simulated data set underwent CA, followed by dimension reduction using the scree plot of calculated singular values, resulting in a reduction to 3 dimensions. The Silhouette scores for the ground-truth clusters were then calculated in the dimension-reduced spaces using three types of coordinates: principal coordinates, standard coordinates, and singular vectors, with the Euclidean distance metric. The dimension-reduced space that yielded the most suitable results for clustering would have the highest average Silhouette score across all items. To account for the impact of the number of dimensions on clustering, the average Silhouette score was computed for spaces with varying dimensions, ranging from 1 to 30.

Figure 3.6 demonstrates the testing results. The average silhouette scores within spaces with principal coordinates (blue), standard coordinates (red) and singular vectors (green) are plotted as curve. It shows that in spaces with 1 to 3 dimensions, the silhouette scores for all three types of spaces are relatively similar. However, as the number of dimensions increases, the average silhouette scores in the standard coordinate space and singular vector space decrease significantly, while they decrease more gradually in the principal space. This suggests that if an appropriate number of dimensions is selected for dimension reduction (in this case, 3 according to the scree plot in Fig. 2.6.3), the Euclidean distance within each of the three types of dimension-reduced spaces can effectively differentiate between clusters of items. However, if more noisy dimensions are included, clustering with Euclidean distance in the spaces with standard coordinates and singular vectors becomes problematic.

This is because singular vectors are orthonormal vector that only provide direction information without indicating the variation of inertia along the direction, lacking the

information which is crucial for distinguishing clusters of data points. Similarly, the standard coordinates rule out row or column masses, such that the mean of each row or column is standardized. This also makes the standard coordinates fail for clustering. For the simulated data set, the first three singular values (the square root of inertia) are close to 1. Therefore, whether or not the three types of coordinates scaling by the singular value is applied does not make a significant difference, resulting in roughly similar silhouette scores for the three types of dimension-reduced spaces within the first three dimensions.

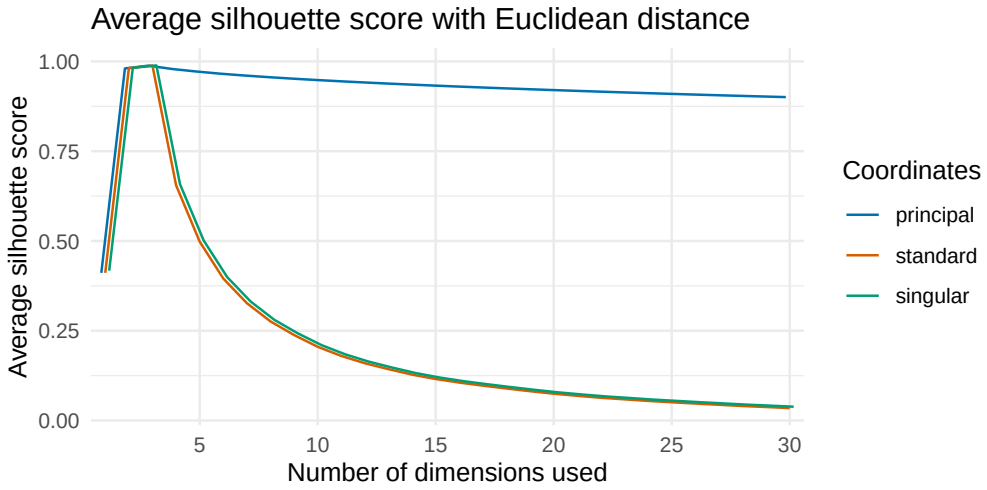

Figure 3.6: **Evaluation on simulated data set with gold standard ground truth of clusters.** The average silhouette score with the Euclidean distance between principal coordinates is robust and much higher than using the standard and singular coordinates when including more dimensions, indicating using the principal coordinates gives the best recovery of clustering ground-truth.

To test if the claim still holds true for complex experimental data, I conducted a similar evaluation on six scRNA-seq datasets with expert-annotated cell types to determine the most suitable coordinate type. The scRNA-seq datasets utilized in this study are listed in Table 3.1, they are data sets BaronPancreas, Darmanis, FreytagGold, PBMC, Tirosh and ZeiselBrain (refer to Table 3.1). The evaluation procedures employed were the same as those used for the simulated data.

The results are shown in Fig. 3.7, with the sub-figures representing results from six data sets respectively. The average silhouette score calculated from spaces with principal coordinates are shown as the blue curves, while for standard coordinates as

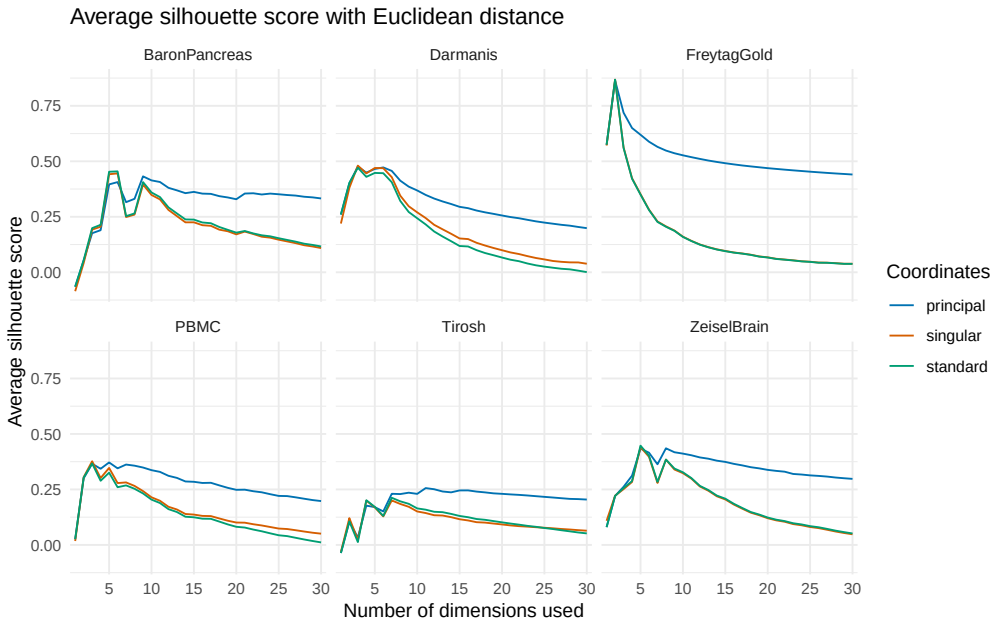

Figure 3.7: **Evaluation on real scRNA-seq data sets with silver standard ground truth of clusters.** The Euclidean distance among the principal coordinates gives the best recovery of clustering ground-truth. The higher the silhouette score is, the better the result is.

green curves and singular vectors as red curves. The results show similar patterns with the results for simulated data. The average silhouette score, calculated based on the Euclidean distance between the principal coordinates in the dimension-reduced space, remains the most robust and highest compared with the other two coordinate spaces (see Fig. 3.7). The principal coordinates are less affected by the increasing number of dimension, i.e. increasing extent of noise, indicating that Euclidean distance in the space with principal coordinates is the most suitable for subsequent clustering analyses.

3.4 Choice of coordinates for clustering in CA space

Most of the existing clustering algorithms can be applied to the dimension-reduced CA space to cluster data points within the matrix. As described in Chapter 2, methods such as hierarchical clustering, K-means, and graph-based community detection methods like spectral clustering and Leiden can all be applied to clustering in the CA space. When employing these clustering algorithms, an essential consideration is the selection

of a suitable distance measure. As discussed in previous sections, when using methods like hierarchical and K-means clustering, similarity among items should be calculated based on the Euclidean distance between principal coordinates. Similarly, for graph-based algorithms, the similarity between vertices should also be determined using the Euclidean distance among the principal coordinates. Despite the distance choice, the number of dimensions can also influence clustering results.

To investigate the impact of coordinate choice and the number of dimensions on clustering, four scRNA-seq datasets: Darmanis, FreytagGold, PBMC, and ZeiselBrain (refer to Table 3.1) were used for the evaluation. Prior to clustering, the data sets were preprocessed according to the method described in Section 3.2, with the number of top variable genes varying between 2000, 4000, and 6000. Selecting different numbers of top variable genes helps to assess the susceptibility of different coordinates to noise and evaluate their robustness. I conducted K-means clustering on dimension-reduced spaces using principal coordinates, standard coordinates, and singular vectors separately. The dimension of the spaces ranged from 1 to 100. For the K-means method, the heuristic input of the number of clusters was determined by the ground-truth number of clusters in simulated data and the number of expert-annotated cell types in scRNA-seq data. Since K-means clustering relies on a random initialization, the random seed of initialization was set as 66 for all runs.

The quality of K-means clustering results was assessed using four criteria: silhouette score, entropy, Calinski Harabatz score (Caliński & Harabasz, 1974), and Adjusted Rand Index (ARI) (Hubert & Arabie, 1985) (refer to Section 2.5.5). Higher values for these indices indicate better clustering performance. The results are presented in Fig. 3.8.

Among the five data sets analyzed in Fig. 3.8, the optimal number of dimensions corresponds to the dimensions with the highest indices, as shown in Figure 3.7. Tthe optimal number of dimensions for all the data sets is below 25. Preserving more dimensions beyond the optimal number introduces more noise to the dimension reduced space, which poses a challenge for the clustering algorithm in distinguishing true signals from the noise.

As the number of dimensions increases, the evaluation indices for K-means clus-

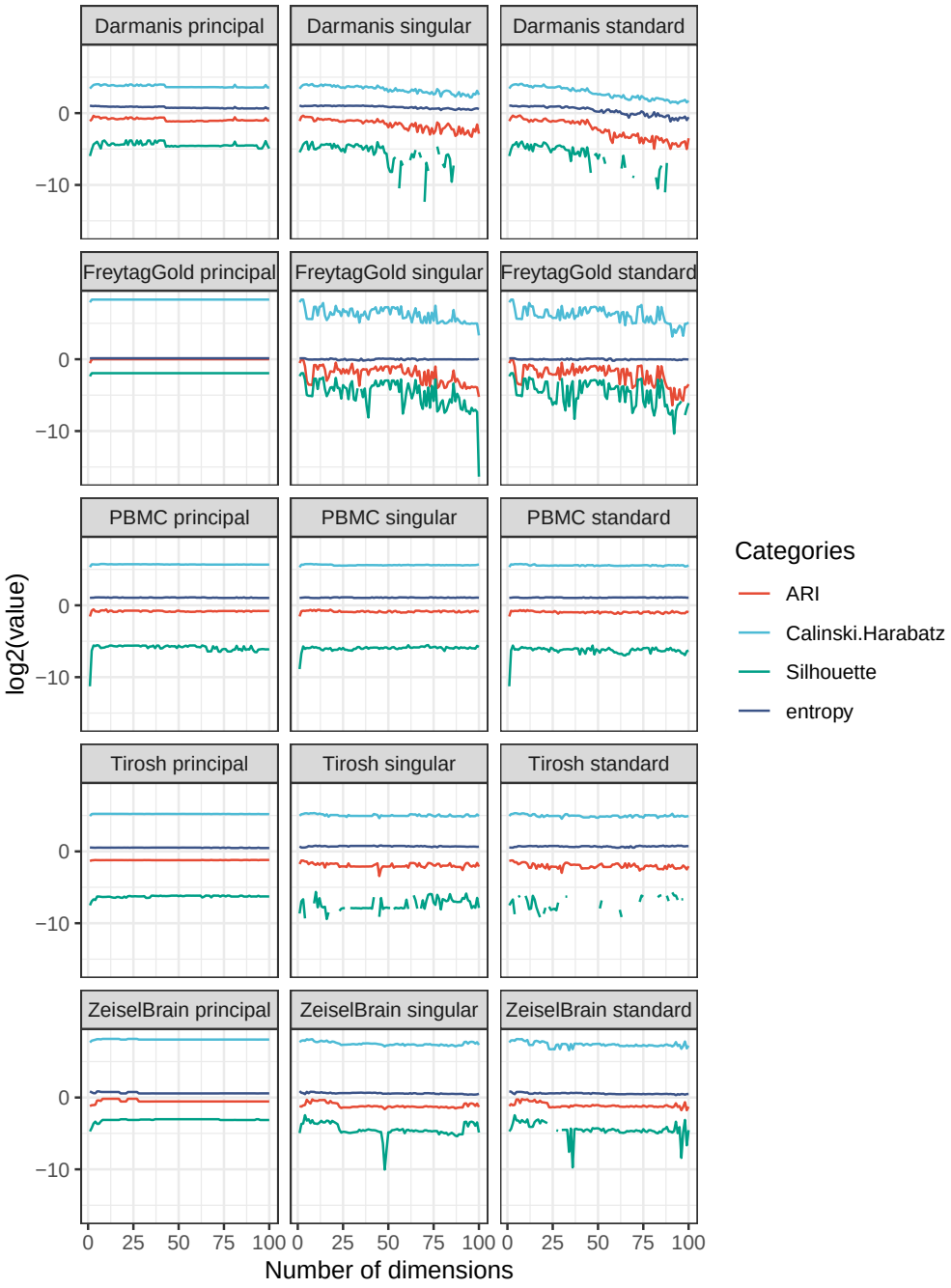

Figure 3.8: **Influence of coordinate choice on clustering in CA space.** The first column of figure shows the evaluation results on principal coordinates over five scRNA-seq data sets (Darmanis, FreytagGold, PBMC, Tirosh and ZeiselBrain). The second and third columns represent the evaluation results on singular vectors and standard coordinates respectively.

tering in spaces with standard coordinates and singular vectors either decrease or exhibit drastic fluctuations. However, the evaluation indices for the space with principal coordinates remain the most robust for four out of five data sets (excluding the PBMC data set). The explanation for this observation is that principal coordinates project the original data into a space where the first several dimensions preserve most of the meaningful signals by assigning them higher weights, while assigning smaller weights to higher dimensions. Therefore, it is more robust than the other two types of coordinates.

In contrast, singular vectors and standard coordinates do not take the contribution of items to the inertia into account, so they will result in skewed downstream clustering when more dimensions are included. This finding confirms that the re-scaling approach used by principal coordinates is the best among the three coordinates.

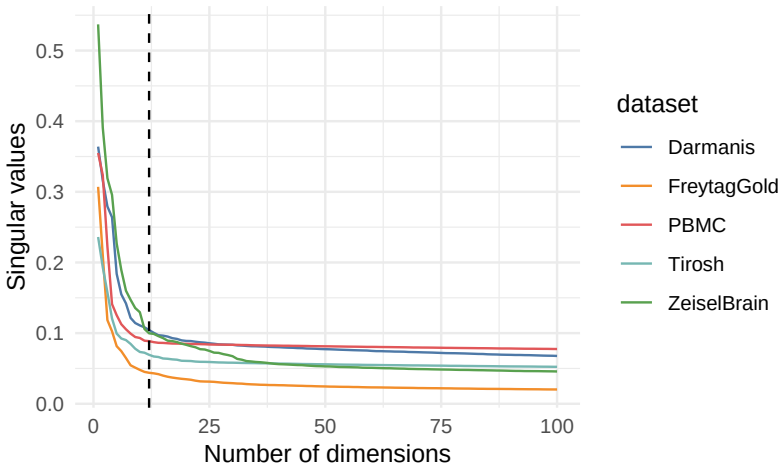

Figure 3.9: **Inertia contained in each dimension after singular decomposition in correspondence analysis.** The x-axis shows the number of dimensions, the y-axis shows the singular values of each dimension which is the square root of inertia. The singular values of different data sets are colored differently.

An interesting observation from Fig. 3.8 is that the FreytagGold data set exhibits the most unstable evaluation indices in spaces with singular vectors and standard coordinates compared to the evaluation values of other data sets. To gain insights into why the standard coordinates and singular vectors are more sensitive for this particular data set, the first 100 singular values were plotted in Fig. 3.9 for all the data sets.

From Fig. 3.9, we can see that FreytagGold data has the lowest singular values in dimension 12 to 100. Since the standard coordinates are scaled by singular values,

the decrease of singular values reduces the contribution of higher dimensions (12 to 100). This allows the K-means algorithm to be less influenced by the additional noises present in the higher dimensions. This finding suggests that the effectiveness of principal coordinates stems from the singular values, as neither the standard coordinates nor the singular vectors exhibit stability for clustering in this context.

In conclusion, in addition to the theoretical evidence supporting the approximation of Euclidean distance between principal coordinates and the χ^2 distance between data points in the original frequency table, the aforementioned results indicate that principal coordinates are more effective in preserving the meaningful signal in the scRNA-seq data. Moreover, they consistently yield more robust K-means clustering results compared to the other two types of coordinates.

4 | Correspondence Analysis based biclustering on Networks (CAbiNet)

In the last chapter, two properties of CA were discussed. One property is that the Euclidean distance measured in principal coordinates leads to a better clustering compared with standard coordinates and singular vectors. The other property of correspondence analysis is that the association between a gene and a cell can be approximated by the inner product of gene and cell vectors in the dimensional reduced asymmetric map.

These geometric properties allow us to establish a connection between similar or associated genes and cells through a single graph, denoted as cell-gene graph. In this graph, cells are not only connected to other cells that exhibit similar gene expression profiles but also to genes that tend to have a high expression level specifically in those cells. By performing clustering on the cell-gene graph, biclusters can be generated consisting of both cells and genes. These biclusters not only distinguish different cell types but also identify gene modules that are specific to each cell type. Leveraging the cell-gene graph constructed from the CA space, we designed a novel biclustering algorithm called *Correspondence Analysis based biclustering on Networks (CAbiNet)*, which facilitates the co-clustering of row and column items. The details of the algorithm will be discussed in Section 4.1 - 4.3. In section 4.4, I will talk about some strategies have been used to accelerate our R package *CAbiNet*.

4.1 Dimension reduction with Correspondence Analysis

Consider a gene expression matrix obtained from scRNA-seq, where the rows represent genes and the columns represent cells. To start, the matrix undergoes preprocessing steps outlined in Section 3.2, which involve removing unwanted cells and genes. The processed matrix is then subjected to CA to calculate the Pearson Residuals, do Singular Value Decomposition and then reduce the dimension of the data set with scree plot (Fig. 4.1). The columns of the matrix $\mathbf{U}$ in Fig. 4.1 are the left singular vectors, and the columns of the matrix V are the right singular vectors. The principal coordinates and standard coordinates can be calculated with formula 2.14, 2.15, 2.16,

2.17. These coordinates can be used to draw CA biplots to visualize the data in lower dimensional space.

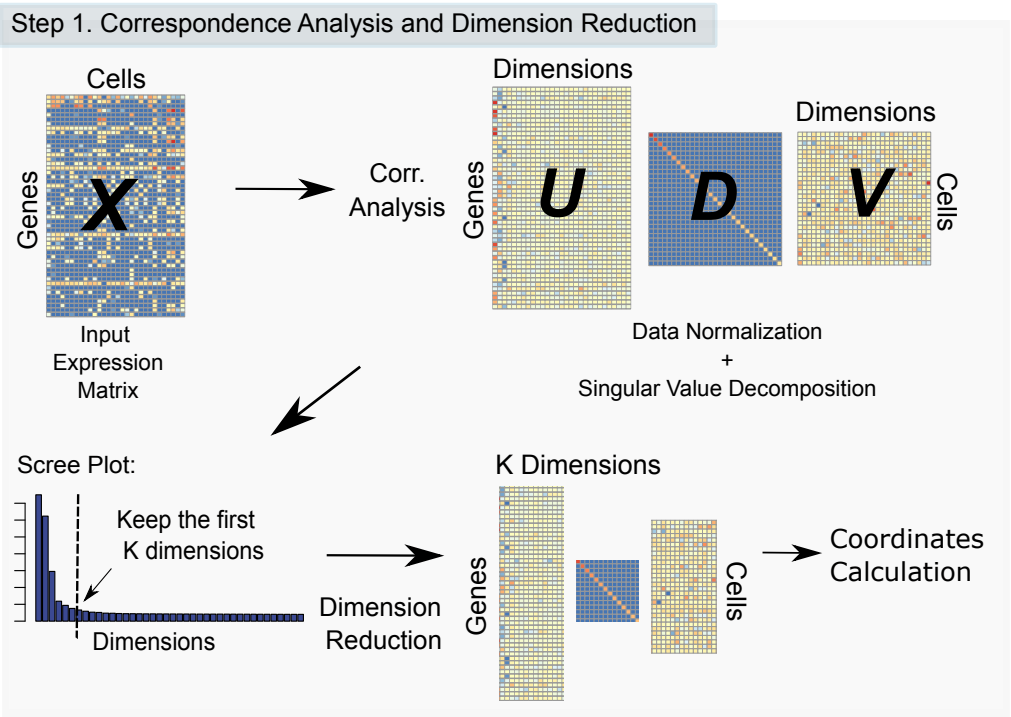

Figure 4.1: **CAbiNet step 1: Apply correspondence Analysis to the matrix.** Data matrix is firstly transformed into Pearson Residuals by Correspondence Analysis, then the residual matrix is decomposed by singular value decomposition to get the unit basis of CA space. Based on the scree plot of explained inertia, that is the eigen values, dimension of the new space can be reduced to K. Scaling the singular vectors, the principal coordinates and standard coordinates of rows and columns can be calculated.

4.2 Build up a gene-cell graph based on Correspondence Analysis

Based on the dimension-reduced CA space, the cell-gene graph is constructed using three main steps. Firstly, k-Nearest Neighbour graph (kNN) cell-gene graph is generated, where each cell and gene is connected to its k closest neighbors. Subsequently, the kNN graph is transformed into a Shared Nearest Neighbour graph (SNN) graph by considering the common neighbors of each node. Finally, an optional step involves trimming out isolated genes to refine the cell-gene graph. The detailed procedures are

as follows:

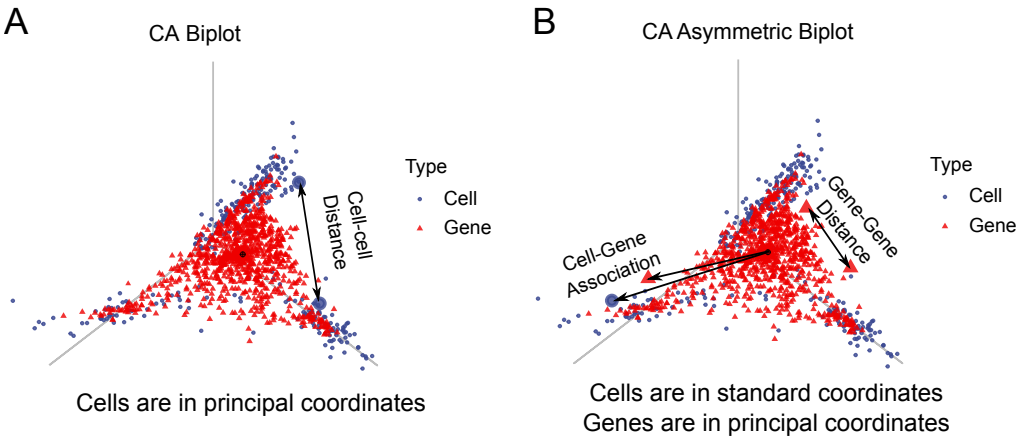

Figure 4.2: **Distance measure in CAbiNet.** To build up a cell-gene SNN graph, the distance between cells are measured in principal coordinates, distance between genes are in principal coordinates as well, while the distance between cells and genes are measured by the inner product between the vectors pointing to cells and genes in a CA asymmetric biplot.

- **Cell-gene kNN graph**. Firstly, a **cell graph** is built up. In the cell graph, each cell is linked to its k-nearest neighboring cells based on the Euclidean distance measured in their **principal coordinates** as illustrated in Fig. 4.2A. This is an unweighted graph with 1 representing a connection and 0 for no connection. Similarly, a gene graph is built, in which each gene is connected to its k-nearest neighboring genes based on the Euclidean distance measured in their principal coordinates in the dimensional reduced space as well (Fig. 4.2B).

  Moreover, a **cell-gene bipartite graph** is constructed. The connections between genes and cells are determined by their **association ratio**, which is calculated as the inner product between the cells and genes in the dimensional reduced asymmetric biplot (see Section 3.3.4 and Fig. 4.2). Specifically, the cell nodes are connected to the top k most associated gene nodes, with the edge direction from the cell to the gene. The edges from genes to cells can be either the same as from cells to genes, resulting in an undirected cell-gene graph (default behavior of package *CAbiNet*), or determined by the top k highly associated cells of the genes, creating a directed cell-gene graph.

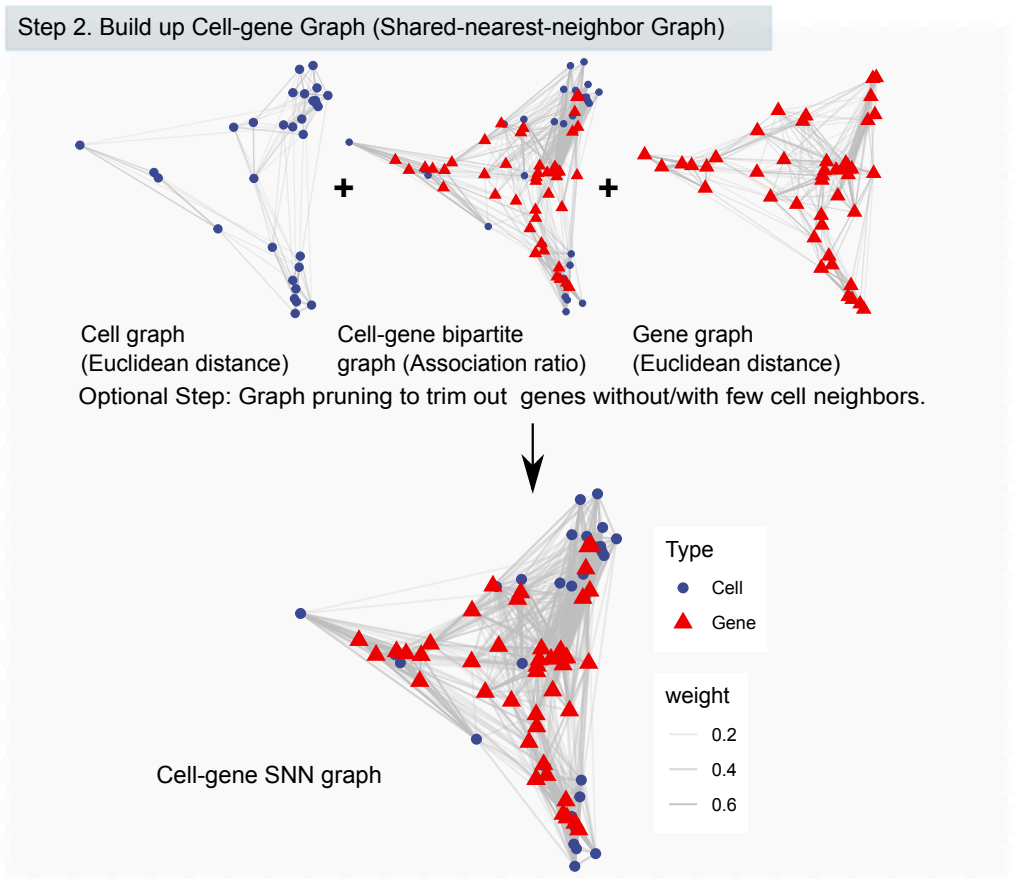

Figure 4.3: **CAbiNet step 2: Build up a cell-gene SNN graph.** The cell-gene graph is composed of three sub-graphs, they are the cell-cell graph built with Euclidean distance between cells measured in principal coordinates, the gene-gene graph built with Euclidean distance between genes measured in principal coordinates and the cell-gene bipartite graph built with the cell-gene association ratio.

Merging all the sub-graphs, we get the cell-gene graph with cells connected with similar cells and highly associated genes, genes connected with similar genes (see Fig. 4.3).

In CAbiNet, users have the flexibility to choose the number (k) of gene-gene neighbors, cell-cell neighbors, cell-gene neighbors, and gene-cell neighbors separately, allowing customization of the graph structure.

- **Transform kNN graph to SNN graph.** For each pair of nodes, suppose the sets of their neighbouring nodes are b_1 and b_2. The overlap between neighbourhoods of each pair of nodes then can be calculated by the Jaccard index in Equation

2.60. The Jaccard scores are then utilized as weights for the edges between pairs of nodes in the gene-cell graph. This transforms the binary adjacency matrix of gene-cell graph into a weighted adjacency matrix with Jaccard scores, representing the strength of connection between genes and cells.

- **Graph pruning**. In practice, even though two nodes are connected in the graph, the neighbouring nodes of these two nodes can be totally different, i.e. the Jaccard index of this pair of nodes are as small as 0, that is $|b_1 \cap b_2| = 0$ such that

$$J(b_1, b_2) = \frac{|b_1 \cap b_2|}{|b_1 \cup b_2|} = 0. \tag{4.1}$$

When the intersection $|b_1 \cap b_2|$ is small, it means these two nodes don't have many neighbouring nodes in common, thus the connection between them is too weak to be informative. Therefore, we set a threshold of the Jaccard index to trim off this type of edges. The threshold is set as $1/15$ by default. It can be adjusted upon the size of neighbourhood.

If the specified number of neighbors, denoted as k, is too small when constructing the cell-gene graph, there is a possibility that certain genes will not be connected to any cells (although they may still be connected to other gene nodes), particularly after transforming the kNN graph into the SNN graph. These unconnected genes indicate a lower degree of association with any specific cells, suggesting that they may not serve as marker genes for any particular cell types. These genes can be removed in the context of identifying cell types and their corresponding marker genes. To address this, CAbiNet provides an option to remove these genes from the graph. After removing these genes, the cells are supposed to be more associated with remaining genes, which are potentially the marker genes of cells.

The cell neighbours of genes are further check. The overlapping of cell neighbours of each pair of genes are calculated by Jaccard index as well. There could be pair of genes do not share sufficient cell neighbours. This can happen when these genes are associated with more than one cell clusters, such that one of the genes is connected with cells from one of the cell clusters, while the other gene

having more connection with cells from another cluster, making the overlapping of the neighbourhoods of two genes small. Therefore, in the case of detecting marker genes of cell clusters, the connection between this kind of genes can be eliminated to give a clear cut between biclusters. CAbiNet allows to set a threshold on the overlap of the cell neighbours of each pair of genes to remove redundant edges between genes.

It is important to note that graph pruning is not mandatory and its application depends on the specific research objectives. If the aim is to identify cell clusters and the corresponding marker genes, it is recommended to trim the genes. However, if the objective extends beyond identifying cell clusters to include analyzing gene expression patterns across the cell population, the graph pruning can be disabled.

4.3 Detection of biclusters

The cell-gene graph combines three sub-graphs: the cell graph, gene graph, and cell-gene bipartite graph. This integrated graph facilitates the reconstruction of not only the correlation between cells and genes but also the similarities among genes or cells. The cell-gene graph can be utilized with various state-of-the-art community detection methods to identify modules, resulting in clusters that may consist of cells, genes, or both (referred to as biclusters).

In CAbiNet, we have incorporated two graph clustering algorithms, namely leiden and spectral clustering. These algorithms assist in uncovering meaningful patterns and structures within the graph, facilitating the identification of distinct clusters.

The Leiden algorithm is the default clustering algorithm integrated into CAbiNet. As explained in Section 2.5.4, the Leiden algorithm is designed to minimize the modularity of the graph, thereby reducing the deviation of node degrees within clusters from the expected values. A crucial parameter for the Leiden algorithm is the resolution, which determines the number of clusters to be identified. Increasing the resolution value leads to the detection of a greater number of clusters.

Another implemented community detection method is spectral clustering, which has been introduced in Section 2.5.3. The adjacency matrix of the cell-gene graph is

first transformed into a graph Laplacian (see Section 2.5.3) by subtracting the degree matrix (refer to Equation 2.37). The graph Laplacian is then normalized by the square roots of the node degree matrix (see Equation 2.38). This normalized graph Laplacian is decomposed into a diagonal matrix containing eigenvalues and a matrix consisting of orthogonal eigenvectors. The eigenvalues are also known as the spectrum of the matrix. The **eigengap** criterion, discussed in Section 2.5.3, can be applied to automatically determine the number of clusters. CAbiNet allows the automatic detection of number of clusters by eigengap when applying spectral clustering. If the eigengap determines the number of clusters as N, then the first N eigenvectors will be utilized for clustering using either K-means or spherical K-means. Therefore, CAbiNet also allows the user to specify the number of clusters by providing an integer input *nclust* when using spectral clustering to detect the biclusters alternatively.

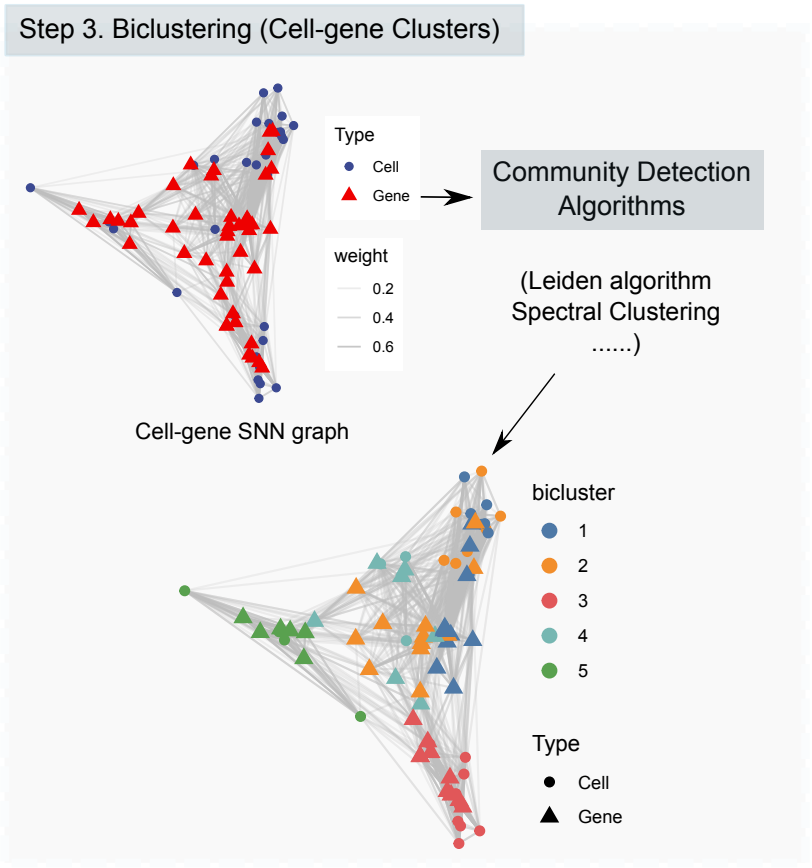

Figure 4.4: **CAbiNet step 3: Detection of biclusters.** Community detection methods, like leiden algorithm and spectral clustering, can be applied to the cell-gene SNN graph to detect the biclusters.

Due to the random initialization of cluster centroids and the local optimization nature of K-means and spherical K-means clustering, running these algorithms with different random seeds can yield different locally optimized results. Since the locally optimized clustering results may be inaccurate, CAbiNet provides the option to run K-means or spherical K-means multiple times. The final clustering result is determined by selecting the run with the least intra-cluster standard deviation. Although the generated clustering is still locally optimized, this approach helps improve the performance of K-means and spherical K-means by selecting the most consistent clustering outcome.

The outputs of Leiden clustering and spectral clustering consist of nodes from the cell-gene graph, without distinguishing between genes and cells. As a result, the clusters can include either genes, cells, or both. Clusters that exclusively consist of genes are referred to as **gene modules**, which represent distinct gene expression patterns across cells and identify co-expressed genes. Clusters comprising only cells aid in differentiating cell types. Biclusters, which include both genes and cells, are assumed to contain genes that are highly expressed specifically in the grouped cells. Some of these genes may be known marker genes that assist in annotating cell types.

CAbiNet provides non-exhaustive biclustering results, as described in the bicluster structure introduced in Section 2.7.2. This means that some genes and columns may not be assigned to any bicluster. However, unlike traditional biclustering algorithms, CAbiNet not only outputs biclusters with genes and cells but also allows for the output of clusters containing only genes or cells. Genes (or cells) are not forced into biclusters if there is no correlation with cells (or genes).

4.4 R package `CAbiNet`

During the initialization process of our R package, `CAbiNet` which is developed by Clemens Kohl and me, we observed that it was relatively slow when dealing with large data sets containing numerous cells and genes. To enhance the scalability of `CAbiNet`, we implemented various optimizations to accelerate the package. Here, I highlight three key modifications that were made by me to improve the code's speed.

Firstly, the original SVD function was designed to decompose the entire matrix,

which becomes time-consuming when dealing with large data sets containing a substantial number of genes and cells. Additionally, for downstream analysis, typically only the first 30-100 dimensions are utilized, rendering the calculation of all dimensions unnecessary.

The efficiency of the calculation of SVD is also influenced by the matrix storage method. In our package, CAbiNet, the singular value decomposition (SVD) is performed twice. The first SVD is conducted by Correspondence Analysis on the Pearson residual matrix, which is usually a dense matrix. The second SVD is carried out on the adjacency matrix of the cell-gene graph when applying spectral clustering to the cell-gene graph in CAbiNet. When a small value of number of nearest neighbours, k, is set for a large data set, the adjacency matrix can become highly sparse. In such cases, converting the adjacency matrix into a sparse matrix representation can save a lot computing memory.

There are many existing packages can be used to calculate SVD and truncated SVD for dense and sparse matrice, including full SVD functions for dense matrix: *svd* function from R base, *svd* function from python package *torch*, *svd* function from python package *scipy*; and also truncated SVD functions for both dense and sparse matrices: *irlba* function from R package *irlba* and *svds* function from *scipy*. To test which algorithm is the fastest in doing SVD for sparse and dense matrices, I simulated 20 random square matrices with 10 in dense matrix format and the other 10 in sparse matrix format. The sparsity of sparse matrices is 90%. The dimensions of matrices range from 100 to 1000. The running time evaluation of each algorithm on each data set is run for 10 times to get an overall evaluation. The partial SVDs are run to calculate the first 10 singular vectors.

The evaluation results are shown in Fig. 4.5, in which the figures on the first row show the results of full SVD of dense matrices (titled as 'Full_svd_dense') and partial SVD of dense matrices ('Partial_svd_dense') and the remaining two figures refer to results of full and partial SVD of dense and sparse matrices ('Full_svd_sparse' and 'Partial_svd_sparse'). The dimension of data sets is shown on the x-axis, and running time with millisecond unit in log10 scale is shown on y-axis. The boxplots summarize the running time of each algorithm on each data among the 10-time trials. For the

dense matrices, the *svd* function from python package *torch* ('torch_svd') is the fastest approach for calculating full SVD and *irlba* is the slowest. For a data with fewer than 500 dimensions, *irlba* is the fastest to get partial SVDs while *svds* function from *scipy* package is the fastest for matrices larger than 500 dimensions. For sparse matrices, 'torch_svd' is also the fastest for calculating full SVDs while *irlba* is the fastest in calculating partial SVDs.

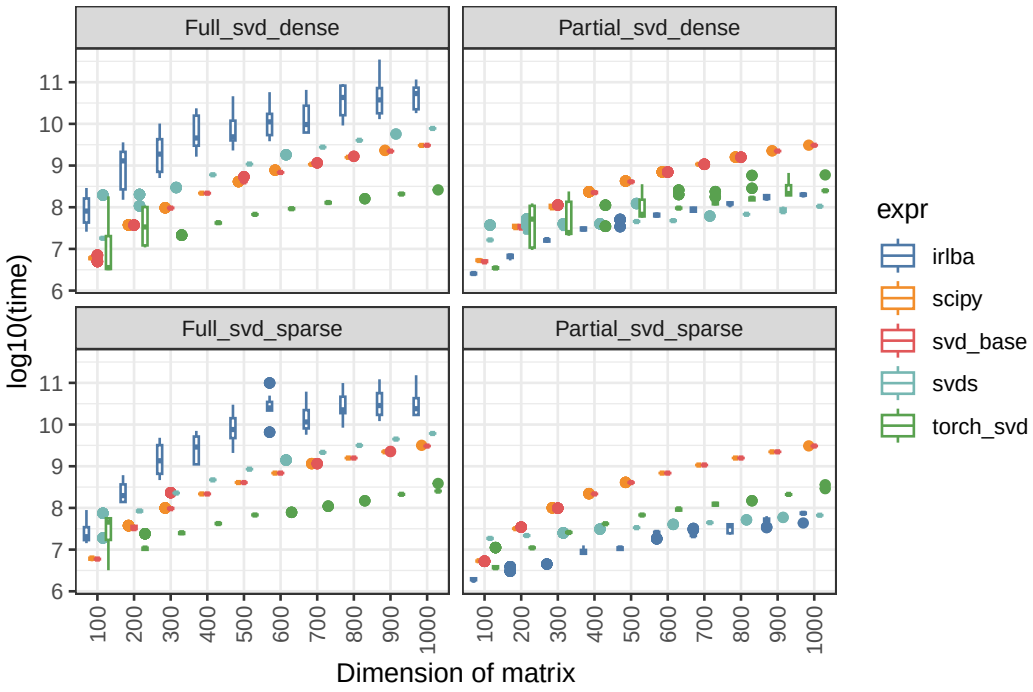

Figure 4.5: **Running time evaluation of full and partial SVD functions on dense and sparse matrices.** The figures from left to right, from up to bottom are the running time of algorithms doing a full SVD of dense matrices ('Full_svd_dense'), partial SVD of dense matrices ('Partial_svd_dense'), full SVD of sparse matrices ('Full_svd_sparse') and partial SVD of sparse matrices ('Partial_svd_sparse'). The running time is in millisecond unit and shown in log10 scale. The x-axis shows the dimension of each data set. Results of different algorithms are in different colors and the boxes summarise running times amog 10 trials.

I further tested influence of number of truncated singular vectors on the computing speed of partial SVDs. For each sparse data set, I range the number of truncated singular vectors from 10 to 90. Each algorithm on each data set with each number of truncated singular vectors is run repeatably for 10 times. The overall evaluation on running time is shown in Fig. 4.6. Each sub-figure in Fig. 4.6 represents the running time of a data set,

with the number of dimension of each data is shown as the title. The x-axis shows the number of truncated singular vectors to calculate, while y-axis shows the running time in log10 scale. As shown, for small data sets, e.g. data with 100 dimensions, calculating the first 10-20 singular vectors with *irlba* is the fastest, whereas *torch_svd* becomes the fastest when more singular vectors should be calculated. However, as shown by the height of the box, the running time of *torch_svd* varies in a wide range. For large data sets, e.g. data with 800 and 900 dimensions, *irlba* is still the robustest and fastest when only calculating the first 10-90 dimensions. However, *irlba* gets slower than *svd_base* when more singular vectors should be calculated. Notably, the running time evaluated for *svd_base*, *scipy* and *torch_svd* is the time of running full SVD, while the other two algorithms for running truncated SVD. Therefore, truncated SVD could be more expensive than calculating the full SVD when too many dimensions are reuired.

Based on the discussion on Fig. 4.5, we make use of *torch_svd* to calculate the full SVD of pearson residuals in CA. We also offer the option of calculating partial SVD with *irlba* in the `cacomp` function from the `APL` package when a number of dimensions to keep is given by the user. For the SVD of graph Laplacian in spectral clustering, we use `irlba` in our package to calculate the truncated SVD. Since the single cell data sets are usually large and `irlba` is more efficient in calculating truncated SVD on large sparse data sets. By adopting the partial SVD approach (for sparse matrix), we have greatly improved the speed of the SVD calculation in our package.

Secondly, I implemented the transformation from a kNN graph into a SNN graph by using C++ implementations, which greatly improves the calculation speed. The source code can be found from https://github.com/VingronLab/CAbiNet/blob/main/src/ComputeSNNasym.cpp.

Thirdly, in the graph pruning step, it is necessary to determine the degree of overlap between neighboring cell nodes for each pair of gene nodes. Previously, this calculation was quite computationally intensive when using R functions. To improve efficiency, we replaced it with a C++ function. The source code for this function can be accessed at https://github.com/VingronLab/CAbiNet/blob/main/src/calc_overlap.cpp.

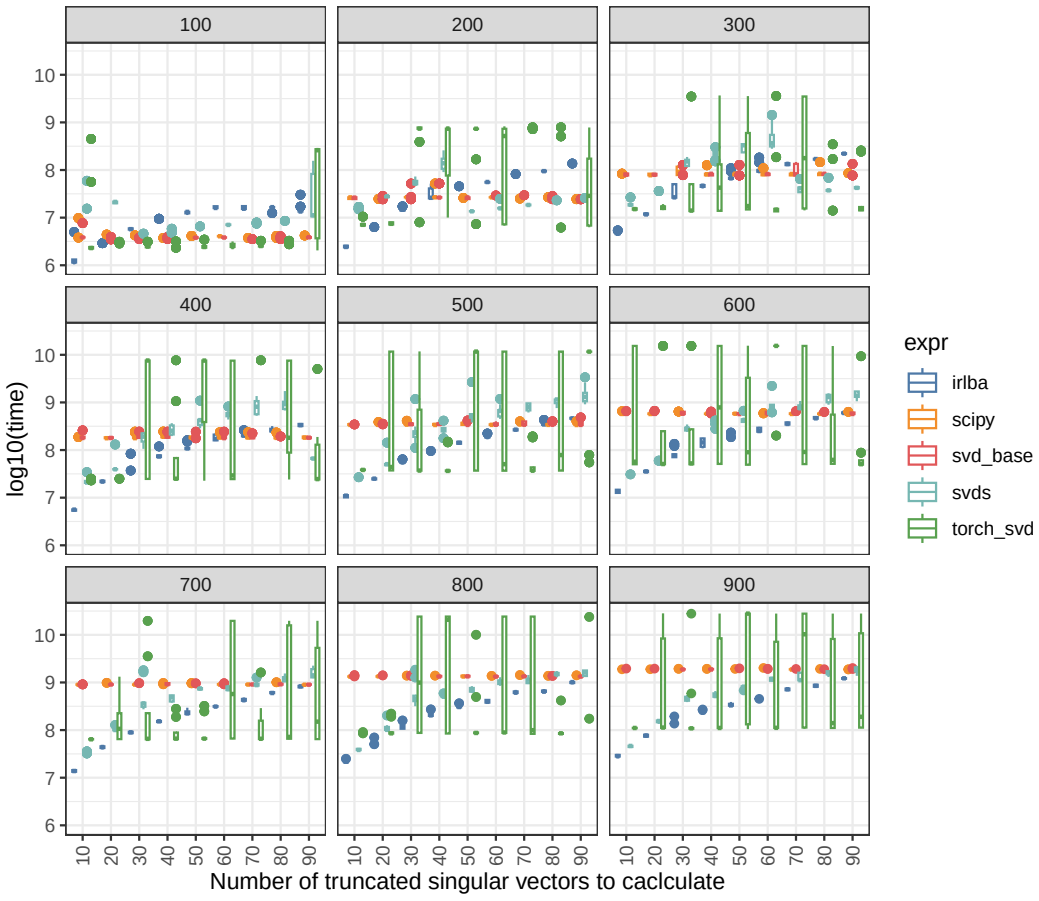

Figure 4.6: **Running time evaluation of partial SVD functions on sparse matrices.** Each sub-panel shows the running time of each data set, with the size of data sets shown in the title of each panel. The running time is in millisecond unit and shown in log10 scale as the y-axis. The x-axis shows the number of truncated singular vectors that has been calculated. Results of different algorithms are in different colors and the boxes summarise running times among 10 trials.

5 | Visualization of biclustering results

PCA and CA biplot provide visualizations of genes and cells in a maximum of three dimensions, limiting the ability to visualize the heterogeneity of data across higher dimensions. Heatmaps, on the other hand, can be difficult to read and extract information from when dealing with large data sets. In order to address these limitations, we aim to develop a non-linear embedding technique for the cell-gene biclusters. This will allow for the visualization in a two-dimensional plane, facilitating a more comprehensive understanding of the data. I will introduce biMAP (Section 5.1) and cabiMAP (Section 5.2) which are created to visualize both cells and genes in a 2D planar in a non-linear manner in this chapter.

5.1 BiMAP with the cell-gene SNN graph

As discussed in Section 2.8.2, UMAP is a technique that reduces the dimension of data by constructing a nearest neighbor graph and optimizing it based on a cost function that preserves both local and global structure in the data. In the context of CAbiNet, a cell-gene SNN graph has already been constructed, so we utilize it as input for UMAP. This allows the lower-dimensional embedding to capture the balance between the local and global structure present in the cell-gene SNN graph. The resulting embedding, referred to as biMAP, positions cells and cell-type-specific marker genes in proximity to each other in a two-dimensional space.

The implementation of biMAP in the `CAbiNet` package involves calling the *umap* function from the R package *umap*. Since the SNN graph is used as input, the function doesn't require the number of neighbors of kNN graph as an input any more, since the neighbourhoods have already been decided in the SNN cell-gene graph. The SNN graph will be directly used for nonlinear embedding. An important parameter for the *umap* function is *n_component*, which is typically set to 2 to generate a two-dimensional embedding, serving as the default parameter.

In the `CAbiNet` package, an interactive output of biMAP is provided, allowing the names of cells and genes to be displayed on the screen and observed by hovering the mouse cursor over the corresponding points in the biMAP. This functionality enables

users to explore the detected marker genes of cell clusters in a more convenient and intuitive manner. Examples of biMAP and interactive biMAP can be seen in Fig. 5.1. The points with black boundaries in the biMAP represent genes, while the others are cells. In a biMAP, the points can be colored by the biclusters they belong to, thus the grouping of cells and corresponding marker genes can be easily observed. With labeling the genes with their names on biMAP or moving mouse over points on an interactive biMAP to print the information of genes onto screen, the marker genes can be easily read and used to annotate cell clusters. This function of biMAP helps scientist with the cell annotation process.

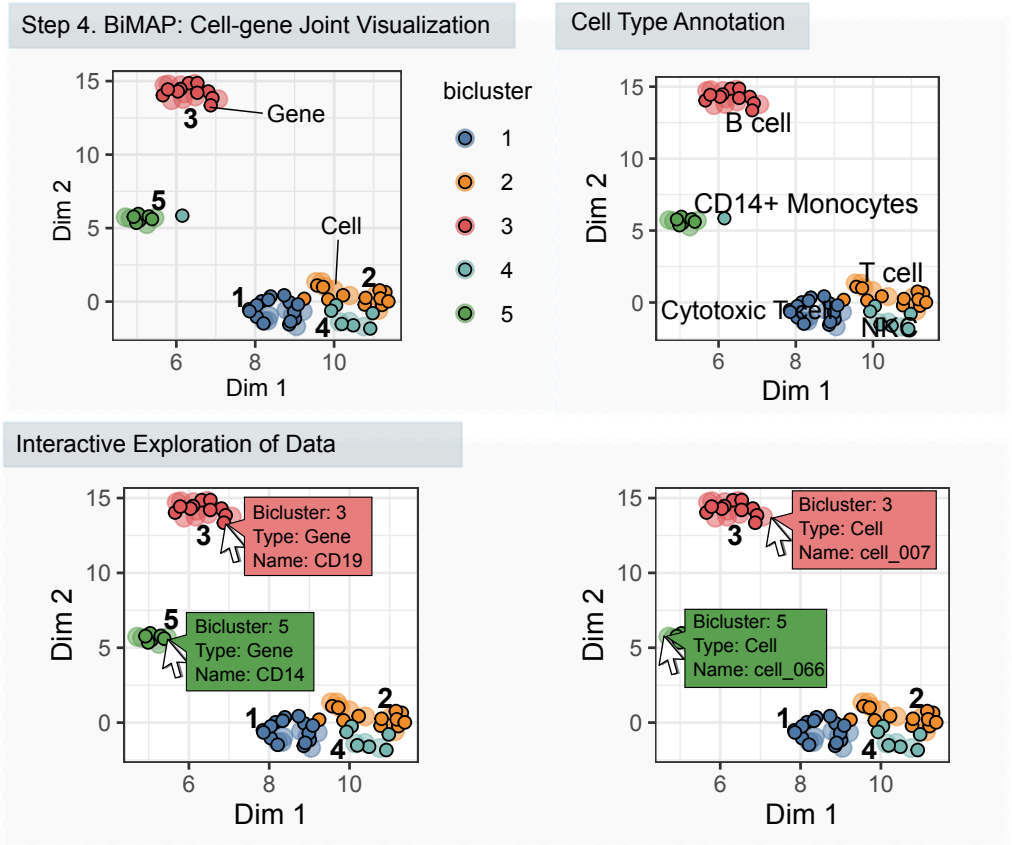

Figure 5.1: **CAbiNet step 4: Visualization of biclusters.** CAbiNet offer a function to generate a simultaneous embedding of cells and genes in a biMAP. The cells and genes can either be color-coded by the biclustering results or the annotation of cell types. The points with black boundaries in the biMAP represent genes, while the other points cells. CAbiNet also allows an interactive exploration of the biMAP. The name of genes and cells can be printed onto the screen with hovering mouse cursor on the data points on the biMAP.

5.2 CabiMAP

The biMAP embedding heavily relies on the structure of the SNN graph, which itself is determined by the parameters used during the construction of the cell-gene graph. One of the key parameters that significantly influences the results is the number of neighbors in the SNN cell-gene graph, denoted as k. Choosing a value of k that is either too large or too small may adversely affect the representation of cell-gene relationships.

Furthermore, the choice of the parameter k significantly impacts the spatial arrangement of genes in the SNN graph-based biMAP. When a specific value of k is chosen, the biMAP may position a subset of marker genes in close proximity to the corresponding cells, while other genes that exhibit similar expression patterns to the marker genes could end up being placed far away from their associated cells. This phenomenon arises from the inherent characteristics of the cell-gene graph construction process. If an inappropriate value of k is used for determining the number of neighbors, certain genes might lose their connections with the cells they should be associated with in the cell-gene graph. Consequently, these genes would be excluded from the biclusters they should belong to. Furthermore, missing of edges/connections between nodes in the graph will lead to segmentation of clusters in the biMAP visualization.

To address these issues, I propose an improvement to the embedding process by generating an embedding directly from the CA projections of cells and genes. This alternative approach aims to mitigate the limitations associated with the SNN graph-based biMAP and provide a more accurate representation of the relationships between genes.

Since the goal is to embed both genes and cells together, the first step is to merge the gene and cell coordinates into a single matrix. This is achieved by row-wise concatenating the cell's principal coordinates $\mathbf{F}$ (Equation 2.17) with the gene's principal coordinates $\mathbf{G}$ (Equation 2.16). Let's assume the concatenated matrix as $\mathbf{A}$, which can be represented as follows:

$$\mathbf{A} = \begin{pmatrix} \mathbf{F} \\ \mathbf{G} \end{pmatrix}.$$

The next step is to calculate a distance matrix between the features (including both genes and cells) in the matrix $\mathbf{A}$. Since the direction of gene and cell vectors matters for gene-cell association interpretation, using cosine distance to calculate the similarity of items (both genes and cells) in matrix $\mathbf{A}$ would be an alternative to using Euclidean distance. The matrix of cosine distance measure can be written as

$$\mathbf{D} = \frac{\mathbf{A}\mathbf{A}^T}{|\mathbf{A}\mathbf{A}^T|} = \begin{pmatrix} \mathbf{F}\mathbf{F}^\mathbf{T} & \mathbf{F}\mathbf{G}^\mathbf{T} \\ \mathbf{G}\mathbf{F}^\mathbf{T} & \mathbf{G}\mathbf{G}^\mathbf{T} \end{pmatrix} / |\mathbf{A}\mathbf{A}^T|.$$

Next, the cosine distance matrix $\mathbf{D}$ is utilized as input for UMAP to generate lower-dimensional embedding, the ***correspondence analysis factor based MAPping of biclusters (cabiMAP)***.

The cabiMAP is applied to one of the simulated data sets mentioned in Section 3.1 and two experimental scRNA-seq data sets. The results and discussion can be found from Section 8.1, 8.2 and 8.3.

6 | Benchmarking

In this chapter, I will evaluate the performance of CAbiNet by comparing it with the existing algorithms with both simulated and experimental scRNA-seq data sets. The evaluation strategies will be introduced in Section 6.1. The evaluation of CAbiNet on simulated data sets will be demonstrated in Section 6.2 and evaluation on experimental data sets will be introduced in Section 6.3. The performance of CAbiNet on detecting the gene modules will be illustrated in Section 6.4.

6.1 Evaluation strategies

CAbiNet is developed for dealing with scRNA-seq data. Its objective is to identify cell clusters characterized by cell-cluster-specific expressed genes, known as cell-gene biclusters. To assess its performance, a comparison is conducted against eight existing biclustering algorithms. Among these algorithms, seven are primarily designed for microarray and bulk RNA-seq data analysis, while the remaining algorithm is specifically designed for scRNA-seq data. The existing biclustering algorithms considered in the evaluation are *Xmotifs, Unibic, s4vd, Plaid, IRISFGM, QUBIC, CCA, and Bimax* (see brief introduction of these algorithms in Section 2.7).

The effectiveness of biclustering algorithms can be assessed by comparing the identified biclusters with known ground truth biclusters. However, in real experimental data, there is no available ground truth for biclusters. To address this, three simulated data sets with varying levels of variation in the designed block biclusters were generated following the procedures introduced in Section 3.1. The effectiveness of the biclustering algorithms on experimental data is evaluated by utilizing the data sets presented in Table 3.1 for the analysis.

The simulated and experimental datasets undergo a preprocessing step outlined in Section 3.2. Specifically, the rows of all data sets are subsetted to include the top 2000, 4000, and 6000 variable genes. These sub-matrices containing log-transformed counts are then used as input for the biclustering algorithms.

Each algorithm has several adjustable parameters, and different parameter combinations can lead to different clustering results. To ensure fairness, each algorithm is

run 108 times with different parameter combinations for each matrix. The parameter values are either suggested by the algorithms or are set close to their default values. The combinations of parameters allows a fair comparison of the algorithms, eliminating of the influence of parameter choices.

For the simulated data sets, where the ground-truth biclusters are available, the agreement between the detected biclusters and the known ground-truth biclusters is assessed using clustering error, recovery, and relevance scores (see Section 2.7.3). Higher scores indicate a greater consistency between the detected biclusters and the ground-truth biclusters. The overlapping between detected and ground-truth cell clusters and gene clusters are evaluated by ARI (see Section 2.7.3). The detailed evaluation results can be found in Section 6.2.

As for the experimental scRNA-seq data sets, since the ground-truth information of cell-gene biclusters is unavailable, the biclustering results are evaluated based on the clustering of cells and genes separately. ARI score is used in this case to evaluate the overlapping between detected cell clusters and the ground-truth clusters. The results will be introduced in Section 6.3.

In addition to evaluating the clustering results, the scalability of the algorithms is assessed by recording their running times. The results can be found in Section 6.3 as well.

6.2 Benchmarking of biclustering algorithm on simulated data sets

Figure 6.1 presents the evaluation results of biclustering algorithms on simulated data sets. The data sets are categorized based on the level of noise present in the data, ranging from the least noisy to the most noisy, and are named accordingly as *easy*, *medium*, and *hard*. Each algorithm is run 108 times with different parameter combinations for each data set, and the evaluation results are displayed as boxplots in the figure. Each boxplot displays a unique category of the clustering result. Figure 6.1 displays the ARI scores for cell clustering, gene clustering, the recovery score of biclustering, and the relevance score of biclustering, represented as subfigures a-d, respectively.

Figure 6.1A illustrates the ARI scores of cell clustering results. It shows that CAbiNet with Leiden and spectral clustering achieves the highest ARI scores, indicating that CAbiNet provides the most accurate cell clustering results. Plaid also performs well and is ranked second in terms of cell clustering. Notably, CAbiNet demonstrates robustness to noise in the data, as its ARI scores remain consistently high regardless of the amount of noise present.

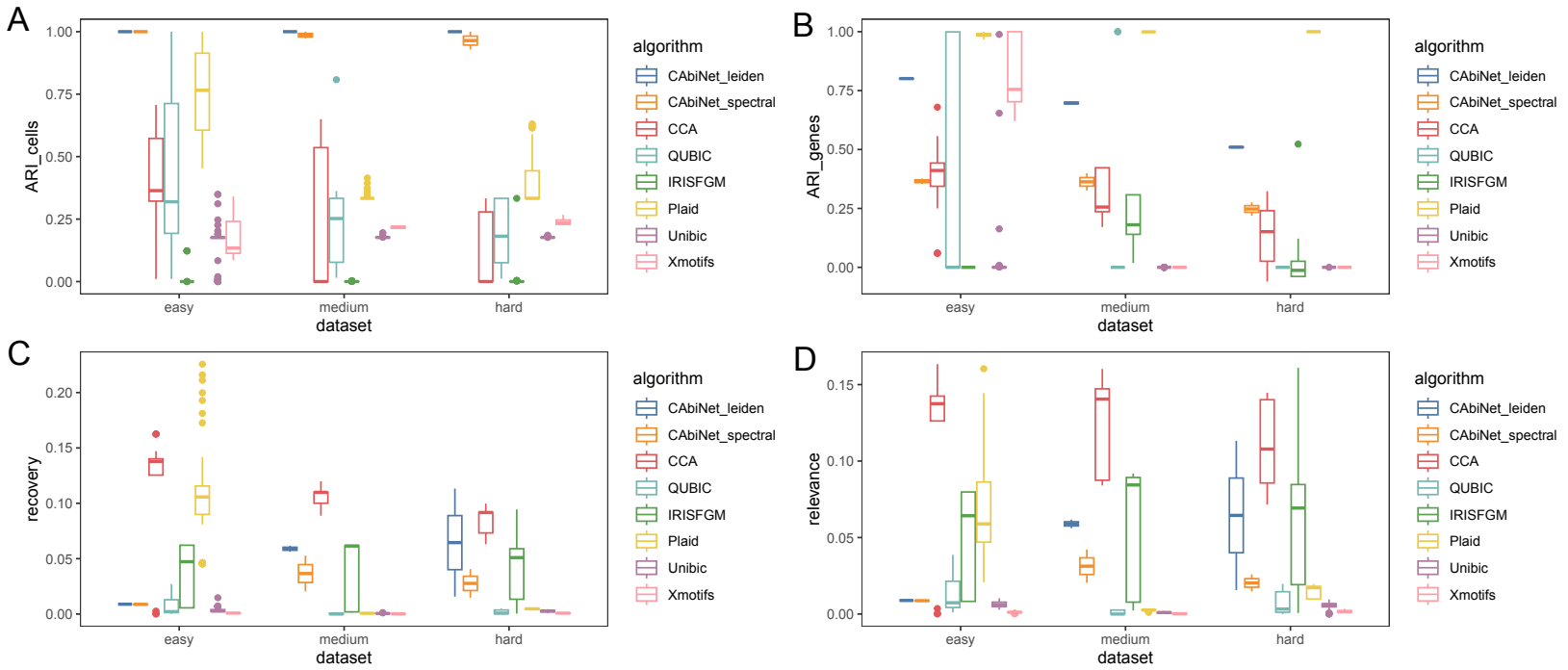

Figure 6.1: **Benchmarking of CAbiNet biclustering against other biclustering algorithms with simulated data sets.** **A**, The ARI of cell clustering in the biclustering results gotten with all parameter choices for each algorithm on simulated scRNA-seq data sets are displayed as boxplot. **B**, The ARI of gene clusters in the biclusters detected by CAbiNet. **C** shows the recovery score of biclusters, it tells the overlapping between detected and ground-truth biclusters. **D** shows the relevance scores of biclusters.

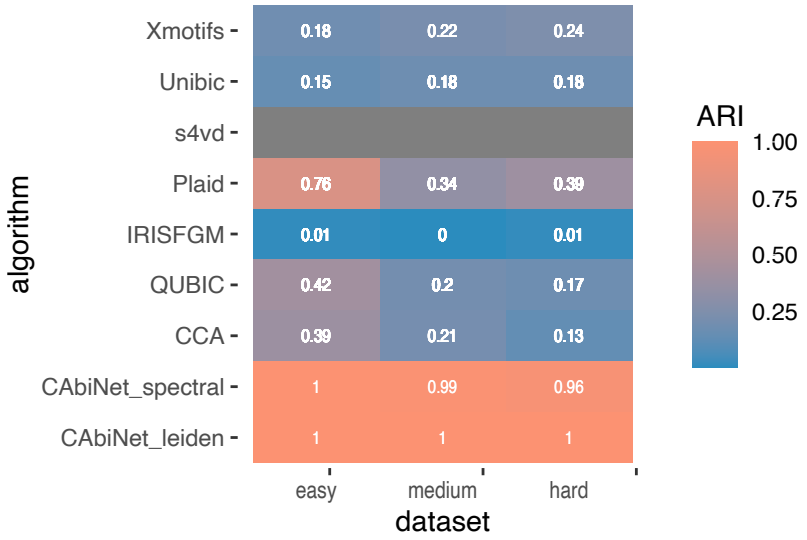

Figure 6.2: **Benchmarking of CAbiNet biclustering against other biclustering algorithms on simulated data sets.** The mean ARI of cell clusters in the (bi-)clustering results over all parameter choices for each algorithm on both real and simulated scRNA-seq data sets are shown on figure. The values are visualized as color-codes and values. The entries colored as grey indicate failure occurs while running the respective algorithm on a certain data set. This can be due to either a few runs failing on a data set or the algorithm only detecting a single bicluster.

In terms of gene cluster detection, CAbiNet demonstrates the second-best performance among the algorithms (Fig. 6.1). It is important to note that although Plaid may appear superior to CAbiNet in terms of cell clustering based on the boxplot, it does not consistently perform well in detecting biclusters under certain parameter combinations (Fig. 6.2 and 6.3). Figure 6.2 represents the mean ARI values for cell clustering, while Fig. 6.3 for gene clustering. Brighter grids indicate higher values and better clustering results. Gray blocks indicate cases where the ARI values of some algorithm runs are not applicable (NA), resulting in NA mean ARI values.

In terms of evaluating biclusters using recovery and relevance scores, CCA emerges as the top-performing algorithm (Fig. 6.1C-D), with CAbiNet achieving the second-best performance.

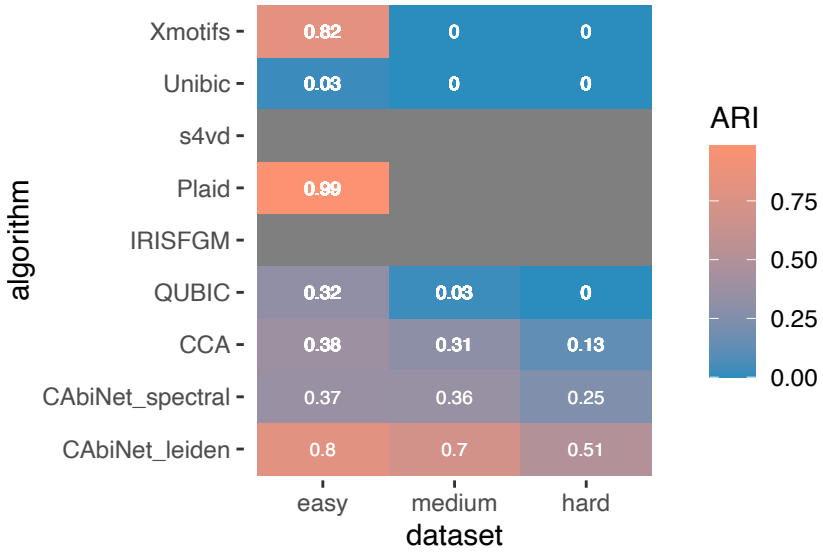

Figure 6.3: **Benchmarking of CAbiNet biclustering against other biclustering algorithms on simulated data sets. A** displays the mean ARI of gene clustering in the (bi-)clustering results over all parameter choices for each algorithm on both real and simulated scRNA-seq data sets. The values are visualized as color-codes and values. The entries colored as grey indicate failure occurs while running the respective algorithm on a certain data set. This can be due to either a few runs failing on a data set or the algorithm only detecting a single bicluster.

6.3 Benchmarking of biclustering algorithm on experimental data sets

For the experimental data sets, the top five algorithms in defining cell clusters are CAbiNet with the leiden algorithm (*CAbiNet_leiden*), CAbiNet with spectral clustering algorithm (*CAbiNet_spectral*), Seurat, Monocle3, and Plaid (see Fig. 6.4). Among the eight other biclustering algorithms, CAbiNet with leiden performs the best on nine out of ten data sets, while Plaid performs the best on the *Tirosh* data set. Surprisingly, the scRNA-seq-oriented biclustering algorithm IRISFGM does not perform as well as expected. When comparing with cell clustering algorithms such as Seurat and Monocle3, CAbiNet with leiden outperforms them on eight out of ten data sets, performs similarly to Monocle3 on the *FreytagGold* data set, and performs worse than Monocle3 on the *Tirosh* data set. It is worth noting that when observing the x-axis of Fig. 6.4, which represents the data sets ordered from smallest to largest sizes, CAbiNet is successful

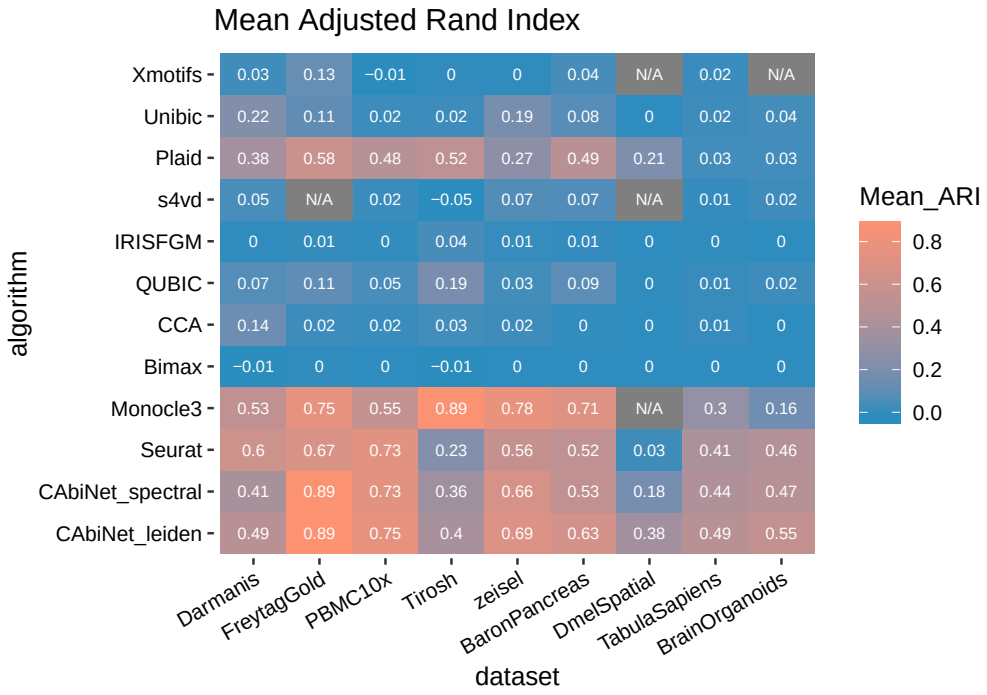

Figure 6.4: **Benchmarking of CAbiNet biclustering against other biclustering algorithms and scRNA-seq analysis toolkits on experimental scRNA-seq data sets.** Mean Adjusted Rand Index of (bi-)clustering results over all parameter choices for each algorithm on both real and simulated scRNA-seq data sets. The entries colored as grey indicate a failure of the respective algorithm on a certain data set, the values are N/As (Not applicable). This can be due to either all runs failing on a data set or the algorithm only detecting a single bicluster.

on all data sets, while Monocle3, Xmotifs, and s4vd failed to handle large data sets (indicated by gray blocks in the figure). It is important to note that the gray blocks in the Fig. 6.4 indicate a complete failure of the algorithms rather than the occurrence of failures during their execution.

It is important for a good biclustering algorithm to not only generate accurate biclusters but also run efficiently on large data sets. To assess the scalability of algorithms, the running time of each algorithm on each experimental data set was recorded and summarized in Fig. 6.5. Among the most accurate algorithms, including CAbiNet with leiden (denoted as *CAbiNet_leiden*) and spectral clustering (*CAbiNet_spectral*), Seurat, Monocle3, and Plaid, the running time of CAbiNet is comparable to the other three algorithms on small data sets. However, on larger data sets, Plaid runs slightly faster than the other four algorithms (for data set sizes, refer to Table 3.1). It is worth

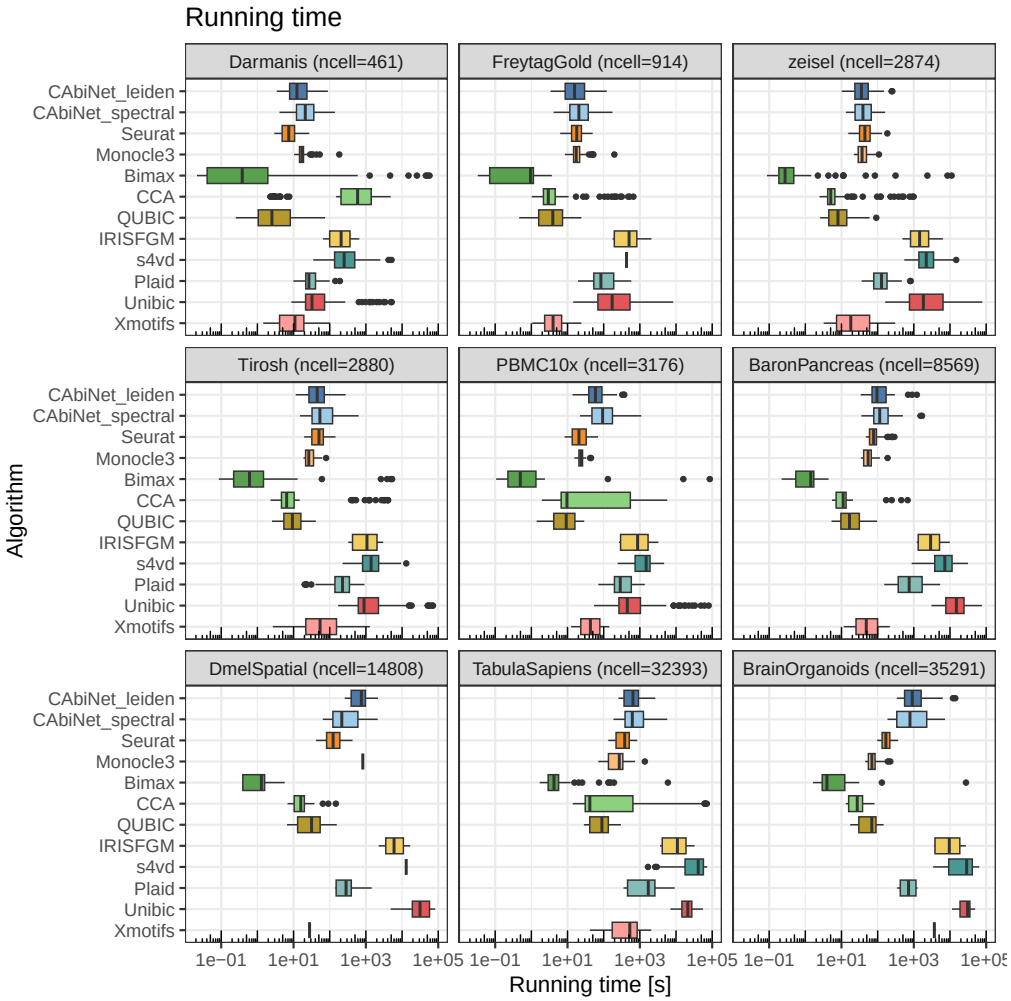

Figure 6.5: **Running time of algorithms on experimental data.** The boxplots demonstrate the running time of each algorithm on each data set over all the 108 runs. The x-axis shows the running time in second.

noting that Bimax has the fastest running time on all tested data sets, but its ARI scores are significantly lower, indicating that it can hardly detect meaningful cell clusters. Furthermore, when compared to IRISFGM, which is specifically designed to handle high drop-outs in scRNA-seq data, CAbiNet is faster on all tested data sets.

6.4 Gene module detection and evaluation

The biclusters produced by biclustering algorithms naturally partition the genes into gene clusters, also known as gene modules. These gene clusters consist of genes

that exhibit similar expression patterns across the cells. Such genes are considered to be co-expressed, meaning they share common expression landscapes. Co-expressed genes often participate in specific biological pathways, working collectively to regulate various cellular processes. In this section, I will evaluate the gene modules detected by CAbiNet and other clustering algorithms and benchmark the algorithms to investigate their performance on identifying biological meaning gene clusters.

Notably, the gene node pruning procedure in CAbiNet (see Section 4.2) is designed to preserve genes that are more likely to serve as marker genes for cell clusters while discarding those that do not exhibit similar cell neighbors among their gene neighbors. However, setting a threshold on graph pruning may result in the removal of a significant number of genes, thereby hindering the detection of gene modules. To utilize CAbiNet as a gene module detection method, it is necessary to deactivate the gene selection function.

6.4.1 Evaluation criteria of gene modules

CAbiNet and the biclustering algorithms discussed in Section 2.7.2 were employed to analyze the experimental data sets provided in Table 3.1. Each algorithm was executed using 108 different parameter combinations on each data set, as previously mentioned. It should be noted that certain methods, including CAbiNet, can produce clusters consisting of cells, genes, or both. In this context, gene modules refer to the gene partition within all the generated (bi-)clusters containing genes.

To assess the gene modules in the biclustering results, gene enrichment analysis was conducted on the gene clusters identified by each biclustering algorithm. The gene enrichment analysis was done with the R package *clusterProfiler*. Three gene annotation databases, namely Gene Ontology (GO), Kyoto Encyclopedia of Genes and Genome (KEGG), and Reactome, were employed for gene module enrichment. A meaningful biclustering algorithm should partition the genes into modules that exhibit significant enrichment in biological processes or regulatory networks.

Enrichment analysis was conducted on the biclusters identified by each algorithm. Each bicluster may exhibit enrichment in multiple pathways, although the significance of these enriched pathways may vary. As a result, we obtain a range of pathways associ-

ated with each bicluster across all the algorithms. This diversity of pathways makes the comparison between biclusters challenging. To simplify the comparison, the enriched pathway with the lowest p-value for each bicluster was selected as a representative. For the biclusters identified by each algorithm in each run, the most significantly enriched pathway was retained for the gene modules in each cluster. Subsequently, the p-values for the pathways were compared across different algorithms. The best-performing algorithm is expected to exhibit the lowest p-values for all the biclusters obtained from the 108 runs with different parameter combinations. The evaluation results can be found in Section 6.4.2.

6.4.2 Benchmarking of gene modules

The evaluation of gene clustering results involves assessing the enrichment of pathways associated with each gene cluster obtained from nine biclustering algorithms on five experimental data sets. The p-values corresponding to the most significant pathway for each cluster are collected and compared across different algorithms. The distribution of these p-values is presented as boxplots in Fig. 6.6. The results indicate that CAbiNet achieves the most significant p-values for enriched pathways in two out of the five data sets (Darmanis and FreytagGold), while obtaining the second most significant p-values in one data set (PBMC_10X) and the third most significant in two data sets (BaronPancreas, Tirosh_nonmaglignant). Bimax performs the best on two data sets (BaronPancreas, PBMC_10X) and the second best on three data sets. This suggests that CAbiNet exhibits comparable performance to Bimax in detecting biologically meaningful co-expressed gene modules.

Furthermore, the percentage of significantly enriched pathways for each gene module is calculated, and the mean percentages for each algorithm on each data set are shown in Fig. 6.7. Bimax is demonstrated to be the best-performing algorithm, as the gene modules detected by Bimax exhibit greater significance compared to other algorithms. S4vd shows the second best performance, while CAbiNet, Plaid, and QUBIC demonstrate similar effectiveness in this regard.

To sum up, CAbiNet is comparable with existing gene module detection biclustering algorithms that have been compared with.

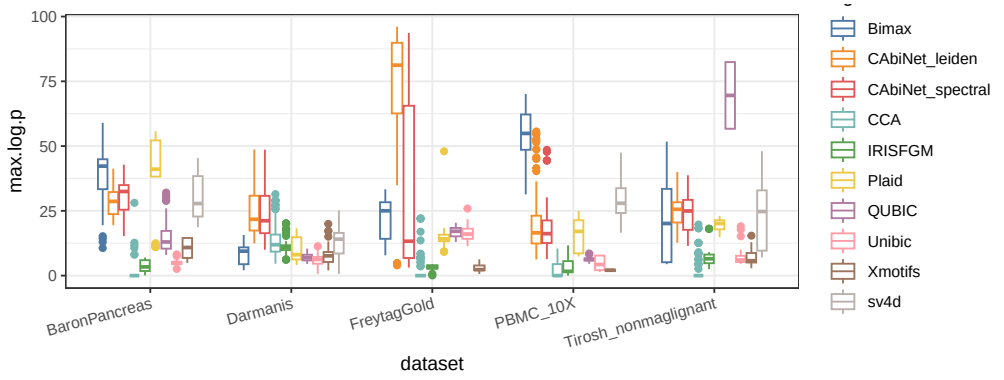

Figure 6.6: **Evaluation of gene modules detected by algorithms on experimental data.** The gene modules detected by algorithms are enriched by gene enrichment analysis, and the minimized p-values of pathways enriched by each gene module for each data set and each algorithm are visualized by boxplots.

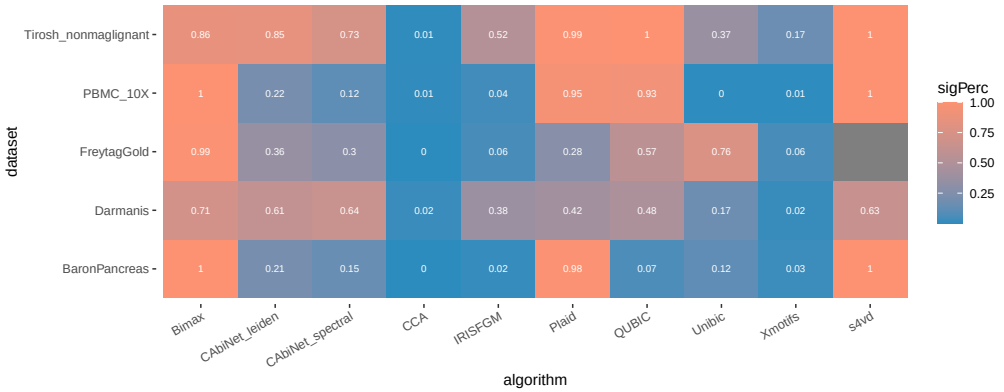

Figure 6.7: **Percentage of pathways with significant p-values enriched by gene modules that generated by biclustering algorithms on experimental data.**

7 | Finding optimal parameter settings for clustering

Determining the optimal clustering results for a data set with unknown cell types can be challenging. In current scRNA-seq data analysis, researchers often adjust parameters multiple times to obtain clustering results. They then identify marker genes for each cluster using statistical tests and compare them to prior knowledge to assess the quality of the results. This iterative process can be tedious, especially for algorithms with numerous parameters. Therefore, an algorithm which can automatically determines the best clustering results within several trials is necessary. Adjusted rand index (ARI, see Section 2.5.5) is used to determine the overlapping between detected clusters and the ground-truth clusters, which tells the goodness of the clustering result. Parameters that optimizing ARI will generate the best clustering result. This initialized my idea to create a model to predict the ARIs of data without ground-truth of clusters. The created random foreset model will be introduced in this chapter.

As mentioned earlier (see Section 2.5.5), the Silhouette score, Calinski-Harabasz index, Davies-Bouldin score and entropy are commonly used to assess the quality of clustering results. Firstly, people will try different parameter combinations of clustering algorithms and then calculate the scores for the clustering results of each run. These metrics theoretically indicate the effectiveness of the clustering outcome. A higher Silhouette score, Calinski-Harabasz score, and a lower Davies-Bouldin score, lower entropy generally suggest better clustering results. The larger the ARI is, the better the clustering quality is. However, in practical scenarios, the relationship between these metrics and the quality of clustering results is not always consistent.

Figure 7.1 depicts the relationship between the evaluation metrics and the clustering quality, which is measured by Adjusted Rand Index (ARI) to determine the overlapping between detected clusters and ground-truth clusters. The clustering was performed on eight experimental scRNA-seq datasets (as listed in Table 3.1) using CAbiNet with both spectral clustering and Leiden clustering. For each data set, the algorithms were run with 360 parameter combinations by varying number of top variable genes, the dimensions of CA space, the number of nearest neighbours. The clustering results were then evaluated using the aforementioned intrinsic metrics and extrinsic measure ARI. The pairwise correlation between ARI, Silhouette score, Calinski-Harabasz score,

Davies-Bouldin score, entropy and number of detected clusters ("Nrcluster") is illustrated in Fig. 7.1. In this figure, the scatter plots show the correlation between each pair of scores with each point representing a clustering trial. The points are colored by the data sets the clustering was tested on. The first row of the figure shows the correlation between ARI and the other four types of metrics. It can be observed that neither of these metrics is linearly correlated with ARI. Moreover, a higher silhouette score does not necessarily coincide with a high ARI, which means a higher or lower value of these metrics can not indicate accurately the clustering is good or bad. Therefore, relying solely on one of these intrinsic score may mislead the interpretation of quality of clustering. Hence, there is a need for a novel measure that accurately reflects the quality of clustering.

Based on the insights gained from Fig. 7.1, which suggests that a single metric can not effectively indicate the true quality of clustering results in most cases, I propose an approach to combine these metrics to predict the clustering quality, i.e. to approximate the ARI. This is achieved by employing a Random Forest regression model, where the Silhouette score, Calinski-Harabasz index, Davies-Bouldin score, entropy and 'Nrcluster' serve as input features, while the ARI is used as the output value.

Since the data sets are sequenced by different techniques and by different labs, the distributions of each score in different data sets can be different. This can also be observed from the density plots on the diagonal of Fig. 7.1. The mean and variance of metrics vary along data sets. To eliminate the batch effect on model training, I trained several random forest regression models separately on each data set. Besides, in order to make the model applicable to new data sets, the eight data sets are divided to two groups: a training group with 7 data sets and a validation group with one data set. Then the model is trained on the training group and tested on the test group. This strategy avoids the model from over-fitting.

The model was constructed with the following steps:

- Select 7 out of 8 data sets as training sets and the remaining one as testing set. The input features are Silhouette score, Calinski-Harabasz index, Davies-Bouldin score, entropy and 'Nrcluster', the predictable value is ARI.

- Train random forest regression models on each training set separately.
- Predict the ARI score of the testing set by using all the trained models, and calculate the average predicted ARI score and output it as the predicted ARI value.

Once the user has a clustering result, they can compute the mentioned metrics and use them as input values for the well-trained Random Forest model. By utilizing this model, the ARI of the clustering results can be predicted. These indices are assigned different weights in the RF regression model to achieve accurate ARI prediction for the clustering results.

This process was repeated for 8 times, with each time leaving out one of the data sets. The predicted ARI scores on the leave-out data set are shown in Fig. 7.2. It is shown that this model can predict ARI of four out of eight data sets quite well, including PBMC_10X, ZeiselBrain, BaronPancreas and brain_organoids data. The model has difficulties in predicting the ARI of FreytagGold and Tirosh data. For some runs the predicted ARI values are not as high as expected. However, this model fails for Darmanis and tabula_sapiens data ARI prediction, where no correlation can be observed between the predicted and true ARI values.

Potential reasons for the failure of model on some data sets could be

- Sequencing techniques. Darmanis and Tirosh data sets are sequenced by SMART-Seq, while most of the other training sets are sequenced by 10X. Readouts of different techniques influence the metrics values.
- Sparsity. Failure on FreytagGold data may be due to the data sparsity. According to Table 3.2, this data set is denser than other ones. Therefore, the intrinsic values of this data set are smaller than that of other data sets, making the prediction inaccurate.
- Insufficient training data. The model is only trained on seven data sets now, and the data sets only cover limited sequencing techniques, data sparsity and data resources (tissue types, species etc).

Considering the potential reasons listed above, this model can further be improved by involving more data sets covering different sequencing techniques and more various

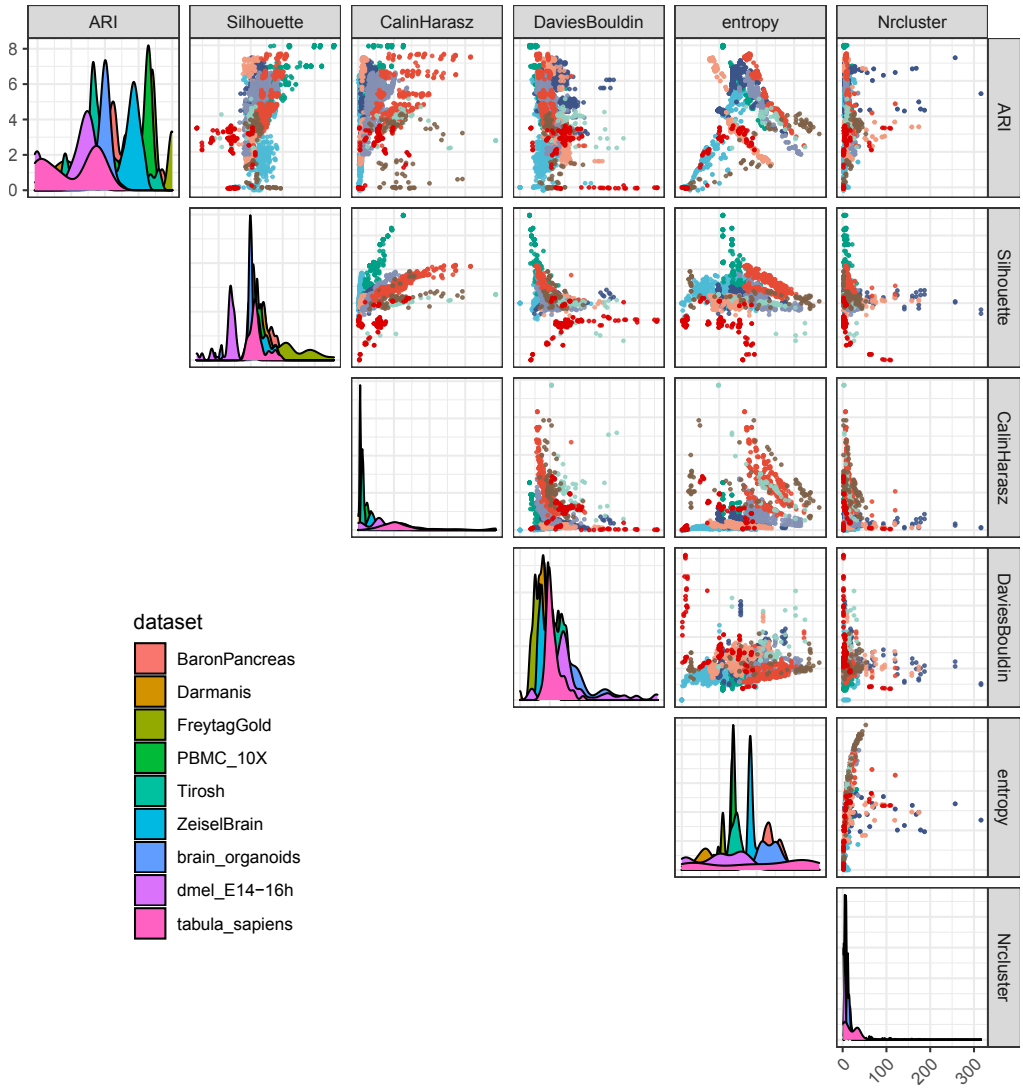

Figure 7.1: **Correlation between evaluation indices of clustering results.** For each combination of parameter choices, a biclustering result is defined by CAbiNet. For each clustering, the ARI, Silhouette score, Calin Harasz, Davies Bouldin score, entropy and number of detected clusters ("Nrcluster") are calculated to evaluate the clustering quality. The correlation between each pair of the metrics are visualized by the scatter plots. Points of different data sets are colored differently.

data characteristics. This model is particularly valuable when working with data sets that lack a ground truth for evaluating clusters. The predicted ARI can serve as an indicator for parameter tuning and assessing the quality of the clustering. In an ideal scenario where an exhaustive parameter search is conducted, the parameter combination that yields the highest predicted ARI would be considered the optimal choice, leading to

the best clustering outcomes. However, performing a comprehensive parameter search is often impractical in real-world situations. Instead, a local grid search of parameters is commonly employed to achieve locally optimized clustering results. The predicted ARI can then serve as a criterion for determining when the clustering is locally optimized.

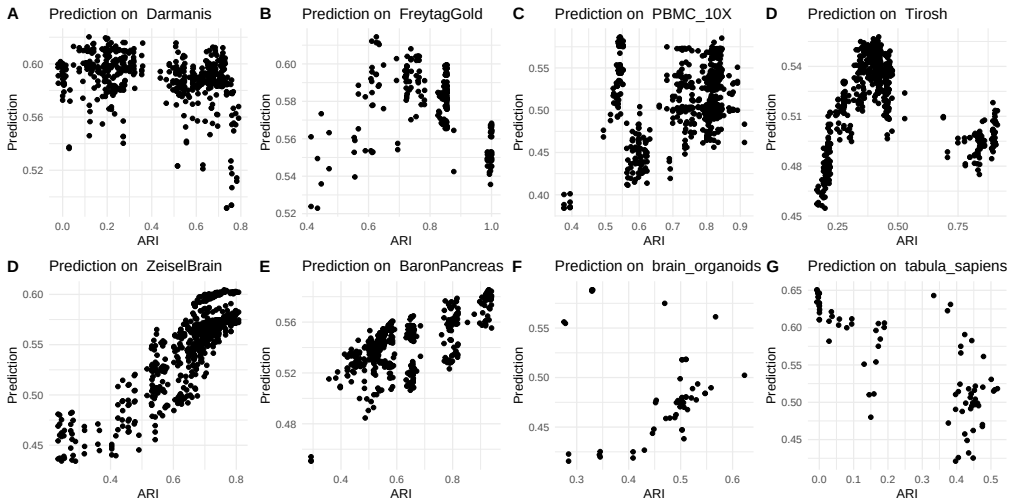

Figure 7.2: **Predictive performance of random forest.** The scatter plot depicts the relationship between the ground-truth ARI, represented on the x-axis, and the predicted ARI, shown on the y-axis. A linear regression model is applied to fit the scatter plot, and the fitted equation along with the residuals between the fitted and original values are presented on the figure.

8 | Application to particular data sets

8.1 Applying CAbiNet to synthetic data

To provide a comprehensive demonstration of the data analysis capabilities of CAbiNet, I initially applied CAbiNet to the simulated scRNA-seq data set mentioned in Section 3.1. The simulated data set with name *easy* was firstly pre-processed following the steps mentioned in Section 3.2. Then the most variable 6,000 genes were retained. The cutoff 6,000 was chosen to cover all five gene modules that have been designed. CA was done on the truncated data set with R package `APL`. The cell-gene graph then was built up by CAbiNet with $k = 100$.

CAbiNet successfully divides the cells and genes into six biclusters. The ARI of the cell clustering result is 0.948, and the ARI of the gene clustering result is 0.947, indicating that CAbiNet accurately identifies the biclusters.

However, the performance of the biMAP, which embeds the SNN cell-gene graph, is not satisfactory. Figure 8.1 illustrates the embedding of the SNN graph constructed from a simulated dataset without gene pruning (Fig. 8.1). biMAP in Fig. 8.1A shows the cells colored by ground truth cell clusters. The cells supposed to be from the same cluster are split into two clouds (e.g. cells labeled as 'cell_B'). Figure 8.1B shows that the marker genes are distant from their corresponding cell types, indicating that the biMAP fails to capture the relationships in this simulated data set. This issue may be attributed to the construction of the SNN graph, where the connections between cell and gene nodes are too weak due to parameter choices, causing the link between them to be lost in the biMAPs. This observation triggers the development of cabiMAP.

Figure 8.2 presents the cabiMAP embedding of the same simulated data set, with the panel on the top showing the embedding of cells colored by ground-truth cell clusters. The panel on the bottom shows the cabiMAP embedding of both cells and genes colored by the biclustering detected by CAbiNet. The points with black borders represent genes while the others representing cells.

Comparing with Fig. 8.1, the cabiMAP in Fig. 8.2 improves the embedding of cells and genes. CabiMAP positions cells with their marker genes closer to each other. The cell clusters and corresponding co-clustered genes can be easily recognized from

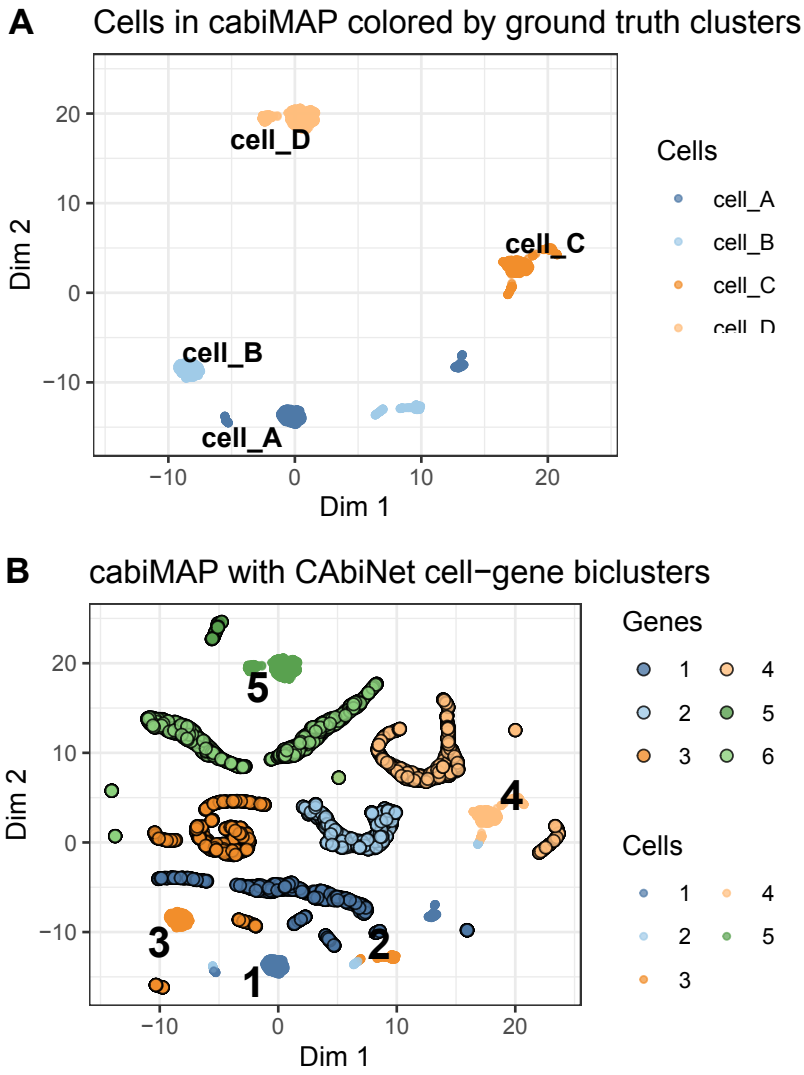

Figure 8.1: **BiMAPs of simulated data set. A**, biMAP only showing cell clusters and cells are colored by the ground-truth clusters. **B**, biMAP with cells and genes colored by biclusters detected by CAbiNet.

the figure.

The underlying reason could be concerned with UMAP algorithm. UMAP tends to distort the actual distance between points when calculating the non-linear embedding. biMAP uses different distance measures for cell-cell/gene-gene and cell-gene subgraphs and the number of nearest neighbours on each graph can also be different, while cabiMAP uses cosine distance to measure similarity between all of the points, it is more likely for UMAP to place the points in proper embeddings in cabiMAP.

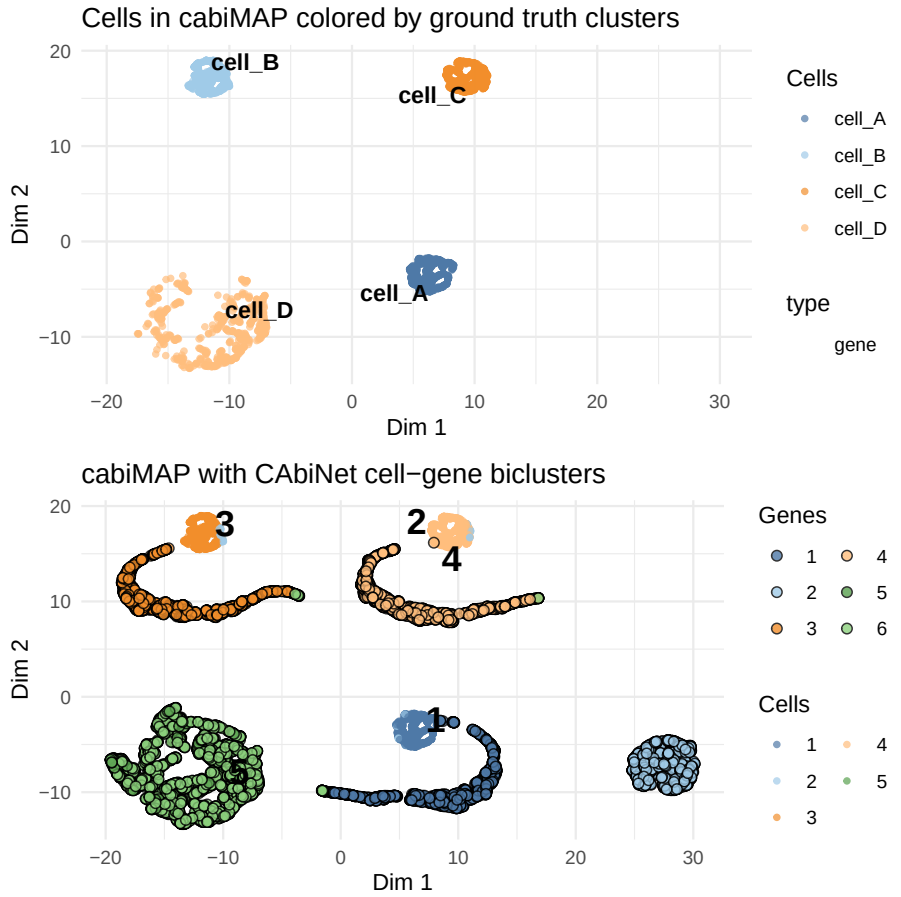

Figure 8.2: **The cabiMAP of simulated data.** In the top panel, the cells are color-coded by ground-truth cell types. In the bottom panel, the cells and genes are colored according to the biclustering results obtained from CAbiNet. In this panel, the points with black borders represent genes, while the remaining points represent cells.

8.2 Analysing scRNA-seq data with CAbiNet: PBM-C10x data

To showcase the fundamental capabilities of CAbiNet, we utilize a single-cell PBMC10x RNA-seq data set (Ding et al., 2020b) as an example. This data set consists of 3,176 cells, which have been categorized into nine distinct cell types. The annotation of cells has been performed by experts using fluorescence-activated cell sorting (FACS) methodology. Notably, the annotated cell types encompass B cells, $CD14^+$ monocytes, and natural killer cells, among others. The data set encompasses a total of 11,881 expressed genes.

The PBMC10x data set (as listed in Table 3.1) was pre-processed according to pre-

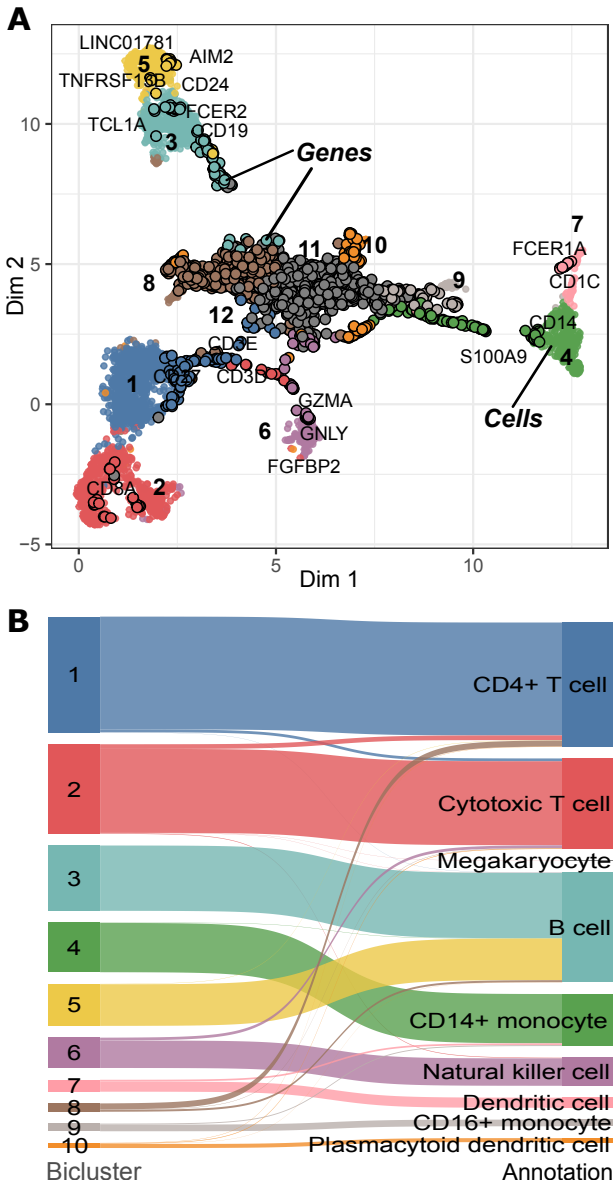

Figure 8.3: **Application of CAbiNet on PBMC10x data. A**, Joint biMAP visualization of the cell-gene biclustering results by CAbiNet, with genes and cells from the same bicluster colored identically. Genes are black circles filled in with the color of the associated cell cluster and cells are smaller dots. Some known marker genes have been labeled manually. An interactive version of this figure can be found in the Supplementary Data. **B**, The agreement between the expert annotation and CAbiNet biclustering results is shown in the Sankey plot.

processing procedures illustrated in Section 3.2 and the top 2,000 most variable genes were retained. This data matrix was subjected to CA and 80 dimensions were kept using the Bioconductor package `APL`. The cell-gene graph was built up with CAbiNet with

$k_c = 20$, $k_g = 20$, $k_{cg} = 10$ and $k_{gc} = 50$. Then Leiden clustering was applied to the graph to find biclusters. Getting the biclustering results from the function `caclust` in our package, we removed those clusters which contain fewer than 10 genes. The biMAP coordinates were calculated with the function `biMAP` with $k = 10$ and plotted with the function `plot_biMAP`. The feature biMAPs in Fig. 8.4 and 8.5 were drawn using `plot_feature_biMAP`.

As mentioned earlier, CAbiNet encompasses dimensionality reduction, clustering, and visualization steps. Following standard pre-processing procedures (Section 3.2), CAbiNet calculates the CA and applies it to project the PBMC10x data into a lower-dimensional space. Additionally, CAbiNet constructs the SNN graph. The algorithm proceeds by identifying biclusters and visualizing the outcomes using a biMAP (Fig. 8.3A), achieved by applying the Uniform Manifold Approximation and Projection (UMAP) technique on our cell-gene SNN graph. The number of genes and cells in each cluster is shown in Table 8.1.

cluster	ncells	ngenes
1	868	98
2	666	30
3	488	100
4	372	124
5	312	23
6	227	59
7	90	13
8	67	553
9	54	54
10	31	55
11	0	837
12	0	14

Table 8.1: **Number of cells and genes in each bicluster.**

There are 12 clusters in total, including 10 biclusters which contain both genes and cell, and two mono-clusters which only have genes. The clustering quality of cell clusters is measured by the ARI. CAbiNet achieves an ARI of 0.79 on this data set, indicating a good agreement between the CAbiNet clustering and the expert annotation. Figure 8.3B shows a Sankey plot illustrating the correspondence between annotation and computed clusters. The large agreement allows us to compare our results with the

expert annotations.

The biMAP visualization in Fig. 8.3A displays the cell and gene clusters. The genes are represented by black circles filled with the color corresponding to their associated cell cluster. Within the biMAP, there are clusters located in the center (clusters 11 and 12) that solely consist of genes. These gene clusters are not specific to any particular cell cluster. In Fig. 8.4, the location of cells and genes are determined by the biMAP, while the color of cell points is determined by the expression level of the gene which is highlighted as a red dot in the plot. The higher the expression level of the gene is, the brighter is the color. It is demonstrated that these genes exhibit ubiquitous expression across various cell types. Consequently, they do not provide meaningful information for differentiating the clusters. Therefore, it is reasonable for CAbiNet to place these genes in the cloud in the center.

The biMAP representation effectively positions cell-type specific genes in close proximity to their corresponding cell clusters. To facilitate interpretation and validation, we manually labeled known marker genes for each cell type on the biMAP. For instance, the genes *S100A9* and *CD14* are located near cluster 4 in Fig. 8.3A, and these genes are established markers for $CD14^+$ Monocytes. This proximity strongly suggests that cluster 4 corresponds to $CD14^+$ Monocytes. The feature plots in Fig. 8.5 confirm the high expression of these two marker genes within cluster 4. The Sankey plot in Fig. 8.3B further supports the identification of cluster 4 as $CD14^+$ Monocytes.

Similarly, the marker genes *FGFBP2* and *GNLY* for natural killer cells are situated near cluster 6, indicating the identity of this cell cluster. The expression pattern of these marker genes in the feature plots (Fig. 8.5) and their alignment with the expert annotation in the Sankey plot (Fig. 8.3B) also confirm the assignment of cluster 6 as natural killer cells.

Interestingly, the biMAP reveals a distinction among the expert-annotated B cells, represented by two clusters labeled 3 and 5 in Fig. 8.3. Each subgroup exhibits its own unique set of marker genes, indicated by the colors that correspond to the cells in the same cluster. Cluster 3, depicted in cyan-blue, appears to consist of naive B cells based on its proximity to the marker genes *FCER2* and *TCL1A* (Fig. 8.3A) (Ramesh et al., 2020). On the other hand, cluster 5 (light-yellow color) includes genes *AIM2* and

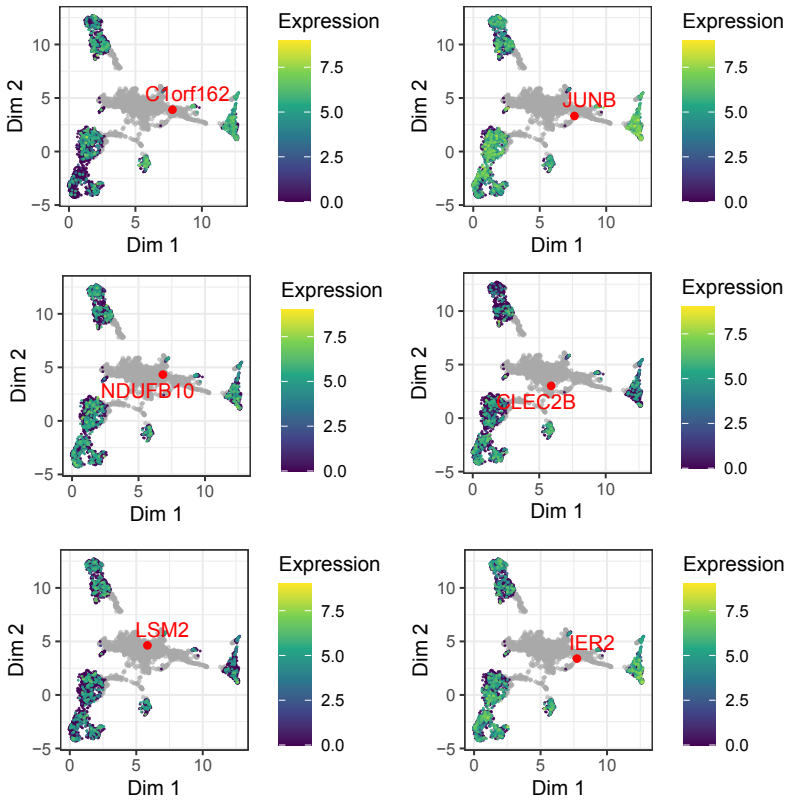

Figure 8.4: **Feature biMAPs for PBMC10x data.** In these biMAPs, cell points are colored by the expression level of the gene that has been highlighted and labeled in red and gene points are colored in gray. The highlighted genes are located at the center of biMAP, without being close to any cell clusters, they have roughly even express levels in the cell clusters which is consistent with the biclustering result that these genes are not specifically expressed in any cell cluster.

TNFRSF13B, which are associated with memory B cells (Ramesh et al., 2020; Franzén, Gan, & Björkegren, 2019), suggesting the identity of cluster 5 as memory B cells in Fig. 8.3A. The expression levels of these genes, as depicted in the feature plot in Fig. 8.5, provide further support for this interpretation.

Moreover, we draw association plots (APLs, see Section 2.4.8) for cluster 3 and 5 to see if the detected potential marker genes are reasonable. Firstly, the centroid of each cluster is calculated and all the cells and genes are projected to the direction of centroid and the orthogonal direction of the centroid. The projection are visualized as the APL as shown in Fig. 8.6, where Fig. 8.6A shows the APL of cluster 3 and Fig. 8.6B shows the APL of cluster 5. The red crosses in Fig. 8.6A and B represents cells belonging

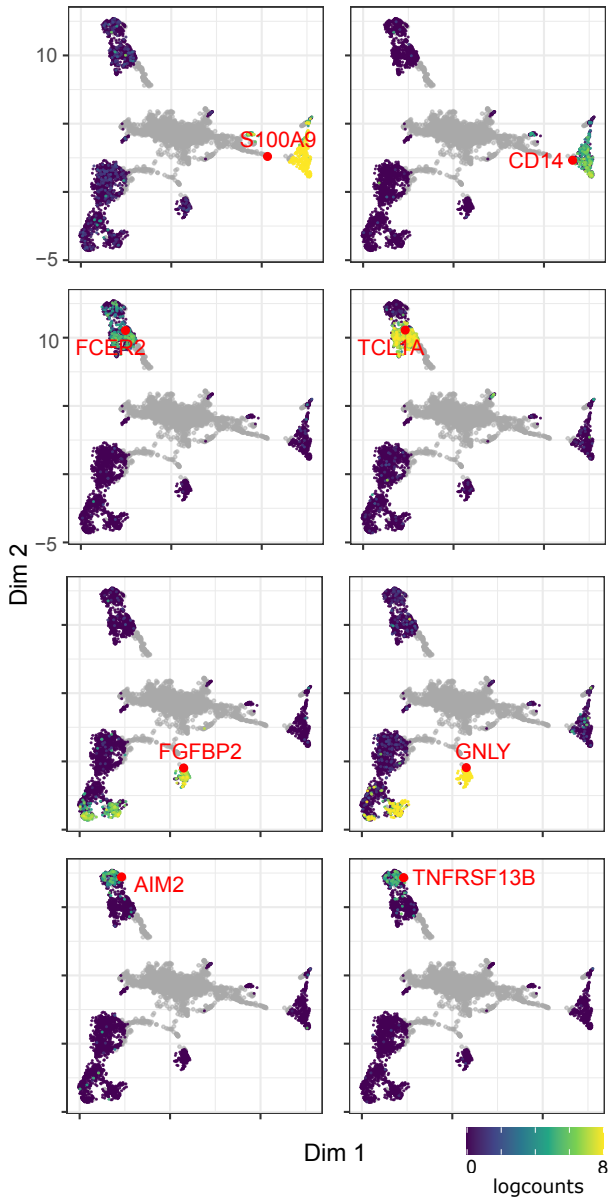

Figure 8.5: **Feature biMAPs with CAbiNet on PBMC10x data.** The expression levels and position of selected marker genes are shown on the biMAP. The grey points are genes and cells are colored by the $\log_2$-expression levels of genes highlighted in red. $CD14^+$ monocytes marker genes *S100A9* and *CD14* in bicluster 4 are highly expressed in cells that co-clustered with them. The natural killer cells marker genes *FGFBP2* and *GNLY* are highly expressed in the co-clustered cells in bicluster 6. *FCER2* and *TCL1A* are highly expressed in bicluster 3, while *AIM2* and *TNFRSF13B* are highly expressed in bicluster 5, indicating that cells in these two clusters are different B cell subtypes.

to cluster 3 and 5 correspondingly. The red circles in plots represent the co-clustered potential marker genes for each cluster. The remaining blue points represent genes from

other clusters and remaining dark red crosses represent cell from other clusters. Most of the detected marker genes are located at the same quadrants with the cells and have large positive values in x-axis, meaning that the genes are positively associated with the co-clustered cells, i.e. the genes are potential marker genes of the cells. The genes that are further to the right and closer to x-axis, the more likely these genes are marker genes of the cells in the observed cluster.

We identified six known marker genes associated with B cells in clusters 3 and 5 of the APLs (Fig. 8.6A and B). In these plots, marker genes are denoted by red circles. Notably, in cluster 3, *FCER2* and *TCL1A* appear as the rightmost genes along the x-axis, while *AIM2* and *TNFRSF13B* which are co=clustered with cluster 5 are positioned closer to the origin. The B cell common marker, *CD19*, falls between these two sets of marker genes. Conversely, the arrangement is reversed in Fig. 8.6B. These APLs illustrate that clusters 3 and 5 possess distinct sets of cluster-specific marker genes, suggesting that cells within these clusters represent subtypes of B cells.

Notably, even though the detected marker genes are positively associated with the co-clustered cells in Fig. 8.6, there are still some genes that are close to zero, meaning that the association between these genes and the co-clustered cell are weak. To test if the detected marker genes are significant or not, we calculated the S_α-score (Gralinska & Vingron, 2023) of the genes in cluster 5. As shown in Table 8.2, 21 out of 23 detected marker genes have positive S_α-scores (see Section 2.4.8), while the other two having negative values. The larger the S_α score of a gene is, the more likely the gene is a marker gene of the observed cluster. This indicates that most of the co-clustered genes are the potential marker genes, but there is still a chance to have false positive predicted marker genes in the co-clusters. Therefore, it is recommended to apply APL and S_α-score to the detected bicluster to double check the significance of the co-clustered genes. The genes that with positive S_α-scores are the ones that are true positive marker genes.

The biMAP depicted in Fig. 8.3A provides a visual representation of cells and their associated marker genes, highlighting both the similarities and differences among cells. However, it may introduce some distortions in capturing the homogeneity and heterogeneity among genes due to the construction of the cell-gene graph. To address this limitation, we applied cabiMAPs to the PBMC data, resulting in different embeddings

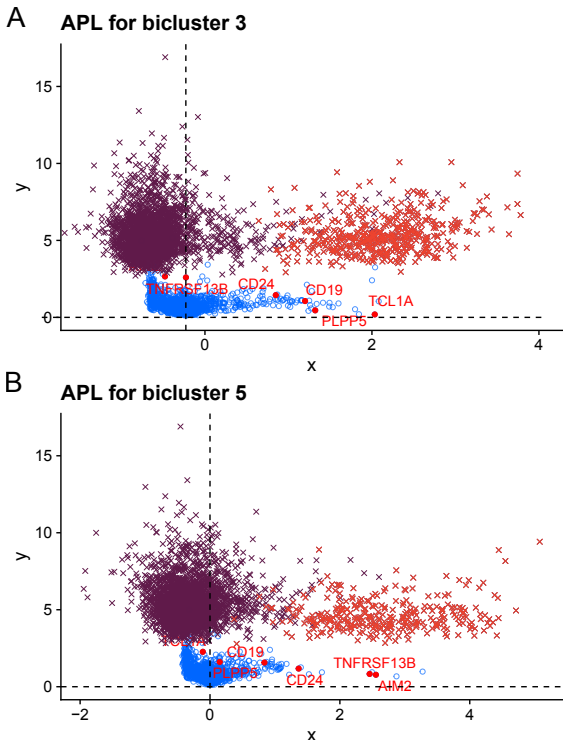

Figure 8.6: **Association plots for bicluster 3 and bicluster 5 in which some marker genes are highlighted.** A, Association plot for bicluster 3. The genes in bicluster 3 are points in red, while the other genes are in blue. The cells in bicluster 3 are crosses in red, while the other are crosses in dark red. B, Association plot for bicluster 5. The genes in bicluster 5 are points in red, while the other genes are in blue. The cells in bicluster 5 are crosses in red, while the other are crosses in dark red. The more a gene is to the right of x-axis, the more likely this gene is a marker gene of the cells that are highlighted in red. The known marker genes are located at the positive x-axis, showing high association with the corresponding cell clusters.

for cells and genes compared to the biMAP with the SNN graph. These alternative embeddings aim to provide a clearer representation of the homogeneity and heterogeneity among genes in the data set. This is shown in Fig. 8.7.

Figure 8.7 shows cabiMAP which was built on the cosine distance between principal row and column coordinates (see Equation 5.2). Figure 8.7A shows the caniMAP embedding of cells and the cells are colored by the expert annotated cell types. Figure 8.7B shows the cabiMAP of cell-gene biclusters, with genes being plotted as points with black boundaries and remaining points as cells. Both cells and genes are colored by CAbiNet detected biclusters.

The marker genes that have been shown in Fig. 8.3 are also labeled in Fig. 8.7B.

Genes	Alpha Score	Rank
ENSG00000253701	2.82	1
IGHA1	2.56	2
AIM2	2.19	3
TNFRSF13B	2.07	4
LINC01781	2.06	5
POU2AF1	1.29	6
BLK	1.15	7
LYPLAL1	0.86	8
CD24	0.82	9
JCHAIN	0.82	10
PDLIM1	0.54	11
GNG7	0.53	12
RALGPS2	0.52	14
ARHGAP24	0.5	16
SPIB	0.5	17
PPP1R14A	0.49	18
BASP1	0.48	19
SP140	0.41	21
PNOC	0.41	23
DDAH2	0.35	26
CHCHD10	0.13	52
ZCWPW1	-0.08	202
DUS2	-0.34	1147

Table 8.2: S_α **scores and ranking of the detected marker genes.** The S_α scores are calculated by APL (see Section 2.4.8). The larger the alpha score of a gene is, the more likely the gene is a marker gene of the observed cluster.

Similar with Fig. 8.3A, the marker genes are locating close to their corresponding cell cluster. For example, gene AIM2 and TNFRSF13B is overlaying with cells in cluster 5 which is annotated as memory B cells. Gene PLPP5 and TCL1A from bicluster 3 are with the cells which are naive B cells. The B cell marker CD19 is locating in the middle of cluster 3 and 5 which are all B cells. The placement of gene CD19 in this cabiMAP is more reasonable than that in the biMAP (Fig. 8.3), since CD19 is evenly expressed in these two B cell subtypes and it does not have a preference over these two cell types. The expression level of CD19 can be observed in Fig. 8.8, in which the expression levels of the genes on x-axis are visualized as points and each cell is colored by the bicluster it belongs to. CD19 shows equally high expression level in cells in cluster 3 and 5.

For the genes in cluster 11 which is a monocluster with only genes, biMAP tend to

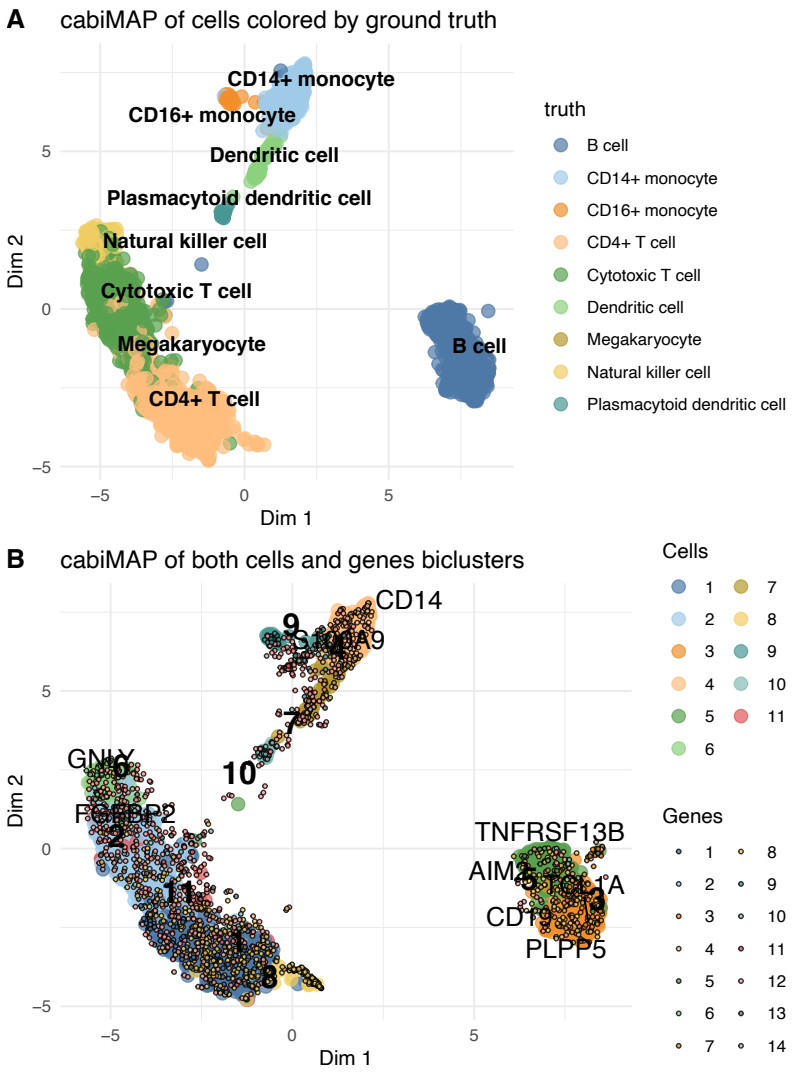

Figure 8.7: **The cabiMAPs for PBMC data. A**, The cabiMAP with only cells and the cells are colored by the expert annotated cell types. **B**, CabiMAP with both cells and genes. The points with black boundaries represent genes, while the remaining points represent cells. Both cell and gene points are colored upon the biclusters detected by CAbiNet.

place them in the middle of the plot separating with cell clusters (as shown as points in the middle of Fig. 8.3A), while cabiMAP overlaying with the cell clusters. As shown in Fig. 8.9, I randomly selected 6 genes in bicluster 11 and labeled them with text. These genes are close to cell clusters that they are not co-clustered with. For example, FTH1 is close to cell cluster 9. To see if the positioning of these genes is meaningful or not, the expression levels of these genes in all the cells are visualized in Fig. 8.9.

FTH1 seems to have a higher expression level in cell cluster 9 (blue dots in Fig. 8.9B) comparing with other cell clusters. Therefore, the closeness of FTH1 to cell cluster 9 in the cabiMAP makes sense.

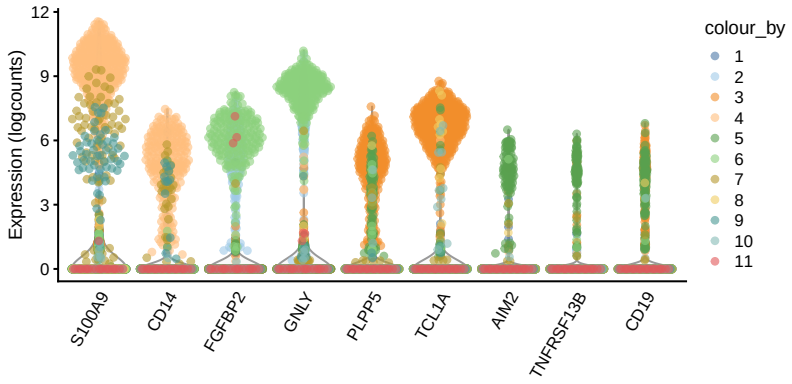

Figure 8.8: **The expression level of genes that not co-clustered with cells by CAbiNet in PBMC data.** Expression levels of genes in each cell cluster are colored differently.

This demonstrates that the cabiMAPs maintain a closer correspondence to the underlying data compared to the biMAP with the cell-gene SNN graph. The cabiMAP allows to visualize the cell clusters with more differentially expressed genes. However, the biMAP with the cell-gene SNN graph performs better in terms of grouping cell-type specific marker genes together with their respective clusters and placing house-keeping genes in the center of the graph. It gives a more concrete visualization of the most specific marker genes.

8.3 Application to Spatial transcriptomic data

Analyzing spatial transcriptomic data poses a unique challenge due to the presence of a large number of drop-outs, where gene expression measurements are missing (M. Wang et al., 2022; Chen et al., 2022b). To investigate the performance of CAbiNet on such sparse data, we applied it to spatial transcriptomic data obtained from late-stage *Drosophila melanogaster* embryos (14-16 hours after egg laying, E14-16h) (M. Wang et al., 2022). The data was generated using Stereo-seq, a technique that resolves the gene expression profile into 14,808 pseudo-cells (corresponding to bins of pixels on a chip) with 7,178 genes (Chen et al., 2022b). In the original publication, 10 cell types

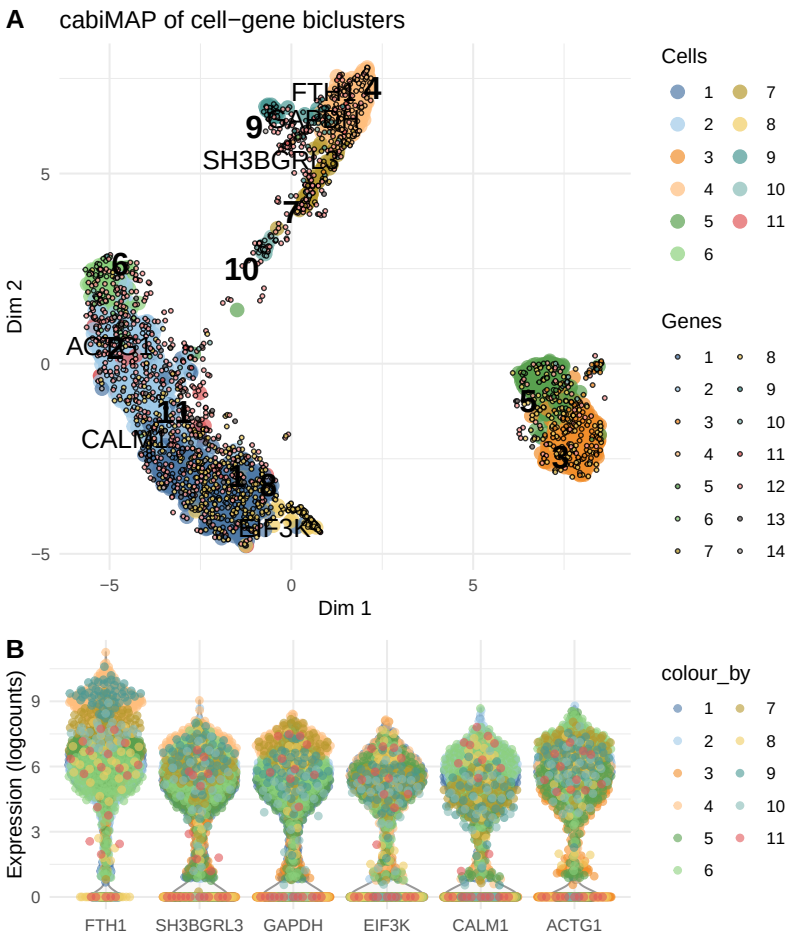

Figure 8.9: **The expression level of genes in PBMC data.** Expression levels of genes in each cell cluster are colored differently.

were annotated based on unsupervised clustering. However, when visualized using the standard UMAP projection in Fig. 8.10A, the boundaries between cell types are poorly defined, making it challenging to distinguish cell types and identify marker genes associated with each cell type. This highlights the difficulty in analyzing and interpreting cell types in spatial scRNA-seq data.

The E14-16h *Drosophila melanogaster* embryo scRNA-seq data by Wang et. al (M. Wang et al., 2022) was pre-processed as described in Section 3.2 and batch effects between spatial slices were removed with the ComBat (Johnson, Li, & Rabinovic, 2007) function from the `sva` package sva. The data was then reduced to 150 dimensions by CA, and the cell-gene kNN graph was built up by using $k_c = 60$ for the cell-cell subgraph and $k = 10$ for gene-gene/cell-gene subgraphs. The gene-cell graph was set

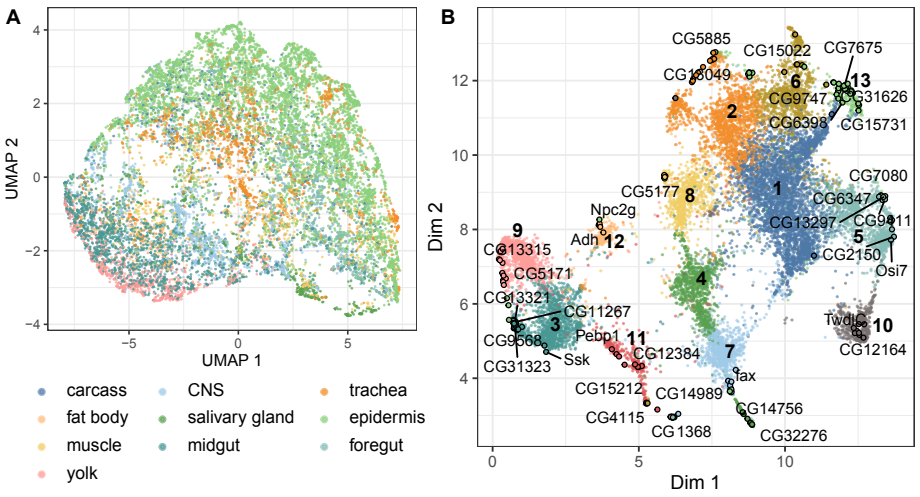

Figure 8.10: **Spatial *Drosophila melanogaster* Stereo-seq data. A**, UMAP embedding of expert annotated cell types. **B**, biMAP embedding of cell-gene biclusters. The genes and cells from the same biclusters are colored identically. Genes are filled circles with a black outline and the cells are the smaller dots. Selected marker genes are labeled in the biMAP.

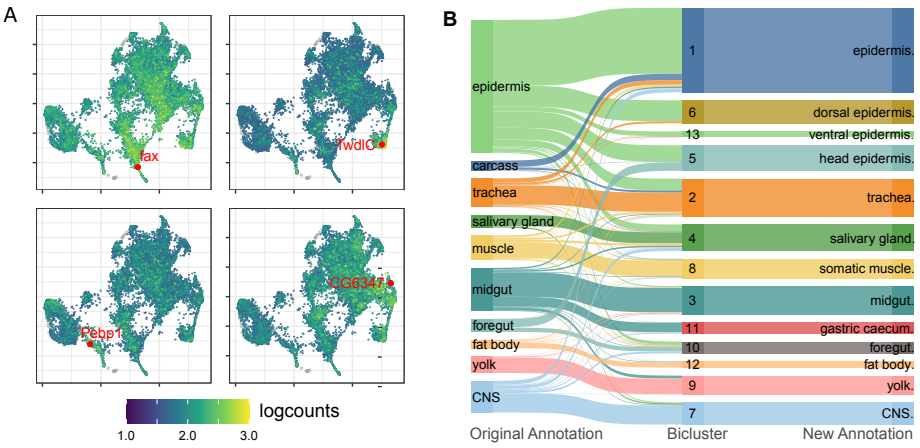

Figure 8.11: **Spatial *Drosophila melanogaster* Stereo-seq data.** **A**, The feature-biMAPs show the expression levels of known marker genes (*fax* (CNS), *TwldC* (foregut), *CG6347* (head epidermis) and *Pebp1* (gastric caecum)) in the cells. The cells are colored by the $\log_2$-expression levels of the highlighted genes. **B**, The Sankey plot shows the consistency among the expert annotation, the biclustering results from CAbiNet and the revised CAbiNet-based annotations.

to the transpose of the cell-gene graph and genes were trimmed by graph pruning with $overlap = 0.1$. The resolution of `Leiden` was 1.2. For a clearer visualization of results, we trimmed out clusters with only genes or cells when plotting Fig. 8.10 and 8.11.

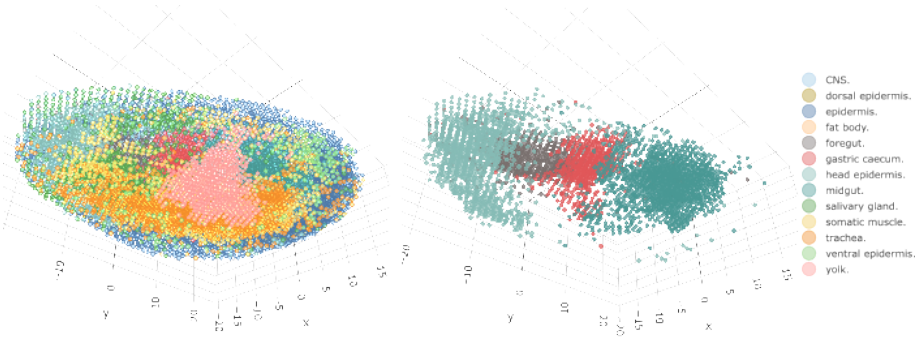

Figure 8.12: **Spatial *Drosophila melanogaster* Stereo-seq data.** Spatial distribution of the cells. The left panel is the 3D visualization of the embryo with cells colored by the biclustering. The right panel shows four cell types out of the left panel. From head to tail they are head epidermis, foregut, gastric caecum and midgut. Interactive versions of panel b and e can be found in the supplementary materials.

CAbiNet successfully identifies 13 biclusters from the spatial transcriptomic data (Fig. 8.10B), each containing co-clustered genes that are biologically meaningful. Notably, cluster 7 consists of 8 out of 14 genes (*fax*, *CG14989*, *Cam*, *Gbeta13F*, *Obp44a*, *ctp*, *fabp*) known to be marker genes for the central nervous system (CNS). Similarly, cluster 10 contains 6 out of 11 genes (*TwdlC*, *CG12164*, *Cpr50Cb*, *Cpr56F*, *Cpr65Av*, *Cpr66D*) known to be foregut marker genes. The expression levels of *fax* and *TwdlC*, as shown in Fig. 8.11A, indicate that these genes are highly expressed specifically in the co-clustered cells, further validating their role as marker genes.

CAbiNet effectively captures the intricate cluster structure present in the spatial transcriptomic data and provides a clear visualization of biclusters, enabling intuitive cell type annotation. Our analysis reveals that the cells originally annotated as midgut in the original publication can be further subdivided into two distinct cell types, as indicated by their assignment to clusters 3 and 11 in the biMAP (Fig. 8.10, 8.11B). By examining the marker genes detected in cluster 11, such as *Pebp1* and *Acbp4*, which are known markers of gastric Caecum, we find that these genes exhibit higher expression levels in cluster 11 compared to other clusters (Fig. 8.11A). This suggests that cluster 11 represents gastric caecum, a sub-structure of the midgut that was not previously identified in the original analysis. Similarly, we identify cluster 5 as head epidermis, a specific subtype of epidermis, supported by the expression level of the head epidermis

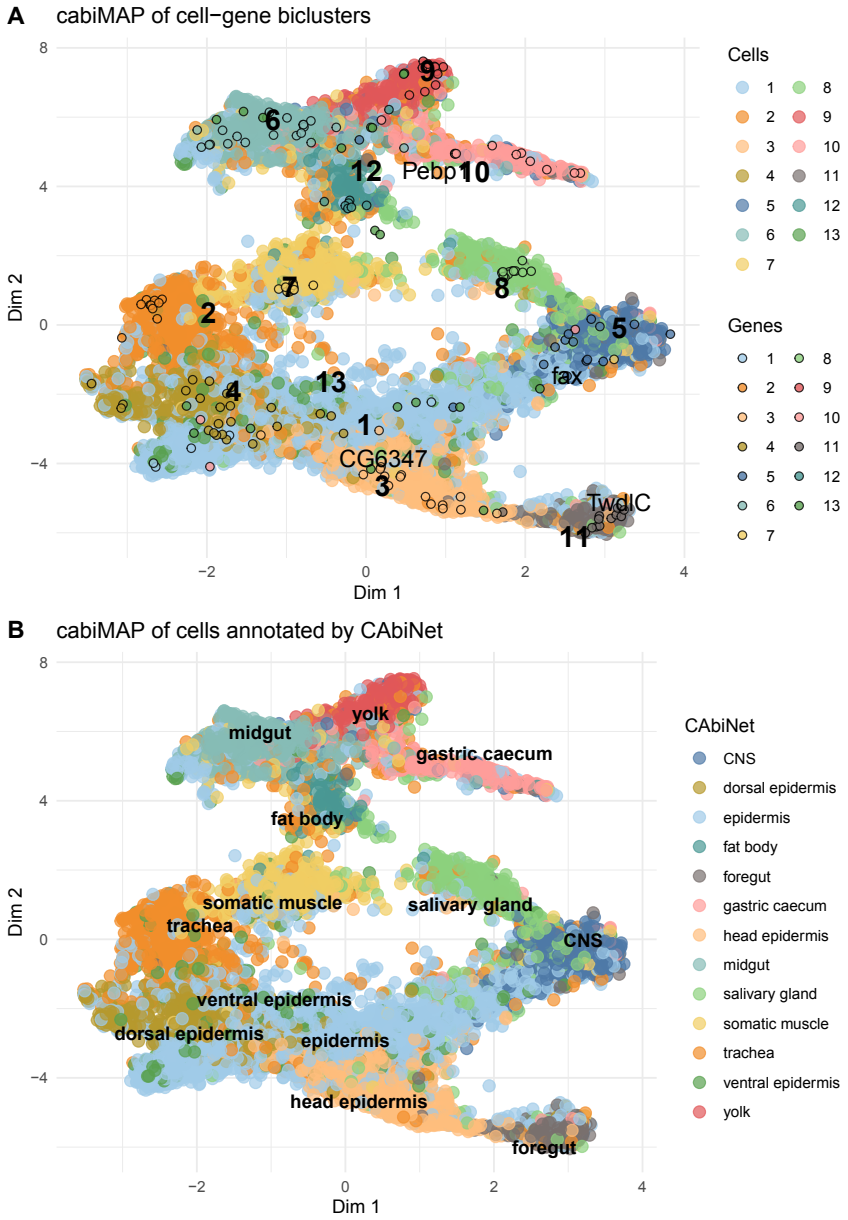

Figure 8.13: **cabiMAP of spatial *Drosophila melanogaster* Stereo-seq data.** **A**, cabiMAPs of both cells and gens colored by biclustering result of CAbiNet. Four known marker genes are labeled with text. **B**, cabiMAPs of cells colored by new annotation of cell types.

marker gene *CG6347* shown in Fig. 8.11A.

Using the biclustering results obtained from CAbiNet, we performed new annotations of the cell clusters, and the resulting annotated cell types are presented in Fig. 8.11B. To visualize the spatial distribution of these annotated cell types, Fig. 8.12

color-codes the cells in the embryo according to the enhanced annotations. Notably, the spatial arrangement of the annotated head epidermis, foregut, gastric caecum, and midgut cells (shown in the right panel of Fig. 8.12) aligns with the actual embryonic anatomy, displaying a sequential arrangement from head to tail.

Figure 8.13 illustrates the visualization of cabiMAP. In Fig. 8.13A, both genes and cells are colored by the biclusters detected by CAbiNet, the genes are the points with black boundaries and cells represented as points without boundaries. Four known marker genes *Pebp, fax, CG6347* and *TwdlC* are labeled with text on the plot. Figure 8.13B shows the cabiMAP of cells which are colored by our new annotation of cell types based on CAbiNet biclusters. Similarly with the biMAP mentioned above, the marker genes again are locating close to their corresponding cell types. For example, gene *fax* is overlaying with cell cluster 5 which is annotated as CNS.

Different from biMAP in Fig. 8.10B that it tends to force the genes to the boundary of cell clusters, the cabiMAP positions the genes more evenly in the embedding. This may be influenced by how the cell-gene graphs are built, and how the UMAP embeds the graph. The embedding of cells are similar in these two types of plots. Both of them allow to distinguish the cell cluster differences. However, since both of the methods employ UMAP to calculate the low dimensional embeddings of the cell-gene graph and the distance in the 2D UMAP is distorted. Therefore, any interpretation of the distance in the biMAP or cabiMAP should be carefully made.

To sum up, CAbiNet offers a more informative and comprehensive integration of genes and cells in its joint embedding compared to the visualization shown in Fig. 8.10A. Additionally, CAbiNet produces detailed biclustering results and facilitates cell type annotation for spatial transcriptomic data with both biMAP and cabiMAP.

9 | Discussion

Correspondence analysis (CA) provides an interpretable approach to understanding the relationship between rows (features) and columns (conditions). This inspired the development of CAbiNet, a method that constructs a graph connecting cells with their highly associated genes.

In this study, I firstly demonstrate that the principal coordinates are the most suitable choice for clustering in the Correspondence Analysis dimension reduced space. The Euclidean distance between principal coordinates approximates the χ^2 distance between the original values in this space. Furthermore, the principal coordinates exhibit greater robustness compared to standard coordinates and singular vectors, particularly in handling data noise. This conclusion is supported by analyses conducted on both simulated and experimental single-cell RNA sequencing (scRNA-seq) datasets.

Based on the distance measured by Euclidean distance between principal coordinates of cells and genes, as well as the cell-gene association measured by the association ratio, a cell-gene graph is constructed. By applying community detection algorithms to the cell-gene graph, CAbiNet naturally identifies biclusters of cells and genes. Our study demonstrates that CAbiNet outperforms existing biclustering algorithms, as evaluated on both simulated and experimental scRNA-seq datasets, in terms of bicluster detection. Furthermore, CAbiNet exhibits comparable performance to established scRNA-seq analysis pipelines, such as Seurat and Monocle3, in accurately distinguishing cell clusters. CAbiNet also demonstrates superior computing speed compared to certain existing biclustering algorithms.

In scenarios where CAbiNet is run without gene pruning, gene modules are recognized naturally by the biclusters. The significance of the gene modules detected by CAbiNet is compared to other biclustering algorithms, the results indicate that CAbiNet can identify biologically meaningful gene modules with greater significance, except when compared to CCA.

Like other existing biclustering algorithms, determining the optimal parameter choices for CAbiNet can be challenging. To address this issue, we employed a random forest regression model to assist in finding locally optimized clustering results. The random forest model takes six scores which measure the clustering quality as input

and generates predicted ARI scores for the clustering results. The ARI scores indicate how well the clustering approximates the "ground-truth" clustering results. Importantly, we observed a strong alignment between the predicted ARI scores and the actual ARI values, demonstrating the accuracy of our model in assessing clustering performance.

We also introduced a visualization technique called biMAP, which allows for the simultaneous embedding of cells and genes in a two-dimensional space. In the biMAP visualization, cells and their associated marker genes are positioned in close proximity, facilitating the visual identification of cell clusters and their corresponding marker genes. This direct spatial relationship between cells and genes eliminates the need for additional statistical tests when annotating the detected cell clusters. Additionally, our CAbiNet package includes an interactive biMAP function, enabling users to hover their mouse cursor over points and retrieve information, thereby simplifying and enhancing the process of cell cluster annotation compared to traditional methods.

In order to address limitations observed in certain scenarios where biMAP was not performing well, such as simulated data sets with simple structures or when improper parameters were used leading to distorted connections between cells and genes, I introduced an improved version of biMAP called cabiMAP. This new version of the visualization technique utilizes the factors derived from CA. cabiMAP can overlay the cells and genes properly in a 2D embedding not only for a simulated data with simple structure, but also for experimental scRNA-seq data with complicated structures. Additionally, cabiMAP not only provides meaningful embedding of cells but also facilitates the identification of gene modules. To cater to different research needs, I have developed four distinct types of cabiMAPs, each reflecting different aspects of gene-cell and cell-gene relationships. Users have the flexibility to choose the most suitable cabiMAP variant based on their specific research objectives.

CAbiNet has demonstrated its effectiveness in detecting cell clusters and identifying marker genes from scRNA-seq data, as illustrated in the PBMC data set and spatial transcriptomic data from Drosophila melanogaster embryos. The utilization of biMAPs and cabiMAPs has provided a more intuitive approach to annotating cell types, even enabling the distinction of sub-cell types within the data. However, it is important to note that CAbiNet currently focuses solely on up-regulated genes, thereby disregarding

the valuable information from down-regulated genes. This limitation restricts the application of CAbiNet to scenarios involving drug-treated samples and CRISPR-screening libraries, where down-regulated genes play a significant role. To overcome this limitation, a potential enhancement for CAbiNet would involve modifying the process of constructing the cell-gene graph by incorporating a signed graph approach that considers both up-regulated and down-regulated genes. By doing so, CAbiNet would be able to capture a more comprehensive view of gene expression patterns and enhance its applicability to a broader range of experimental conditions.

Another concern regarding CAbiNet is its utilization of Leiden clustering and k-means based spectral clustering, which produce non-overlapping biclusters. This means that each cell and gene can only be assigned to a single (bi)cluster. However, certain genes may act as common markers for multiple cell types. For example, CD19 is a marker gene for various B cell types, including Naive B cells and memory B cells. In such cases, CD19 could potentially be assigned to both sub-cell clusters, whereas CAbiNet currently assigns it to only one of the sub-cell clusters. To address this issue, we aim to enhance CAbiNet by implementing fuzzy clustering algorithms, allowing genes to be assigned to multiple cell clusters. However, it is worth noting that fuzzy clustering may also blur the boundaries between cell clusters. Assigning the same cell to multiple cell clusters can make downstream analyses, such as differential gene expression analysis, more challenging compared to having a definitive clustering result. Therefore, there is a need to develop a biclustering approach that provides solid clustering for cells and fuzzy clustering for genes, striking a balance between accuracy and interpretability.

Bibliography

10x Genomics. (2016). *Pbmc3k - datasets - single cell gene expression.* https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/pbmc3k?

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, *2*(4), 433–459.

Ahlmann-Eltze, C., & Huber, W. (2023). Comparison of transformations for single-cell rna-seq data. *Nature Methods*, 1–8.

Alsina, L., Israelsson, E., Altman, M. C., Dang, K. K., Ghandil, P., Israel, L., ... others (2014). A narrow repertoire of transcriptional modules responsive to pyogenic bacteria is impaired in patients carrying loss-of-function mutations in myd88 or irak4. *Nature immunology*, *15*(12), 1134–1142.

Alter, O., Brown, P. O., & Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences*, *97*(18), 10101-10106. Retrieved from https://www.pnas.org/doi/abs/10.1073/pnas.97.18.10101 doi: 10.1073/pnas.97.18.10101

Baron, M., Veres, A., Wolock, S. L., Faust, A. L., Gaujoux, R., Vetere, A., ... Yanai, I. (2016, October). A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure. *Cell Systems*, *3*(4), 346–360.e4. doi: 10.1016/j.cels.2016.08.011

Beh, E. J., & Lombardo, R. (n.d.). Correspondence analysis using the cressie–read family of divergence statistics. *International Statistical Review*, *n/a*(n/a). Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/insr.12541 doi: https://doi.org/10.1111/insr.12541

Bendixen, M. (1996). A practical guide to the use of correspondence analysis in marketing research. *Marketing Research On-Line*, *1*(1), 16–36.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, *2008*(10), P10008.

Bolla, M. (1991). *Relations between spectral and classification properties of multi-*

graphs. DIMACS, Center for Discrete Mathematics and Theoretical Computer Science.

Boyer, C. B., & Merzbach, U. C. (2011). *A history of mathematics*. John Wiley & Sons.

Bozdağ, D., Kumar, A. S., & Catalyurek, U. V. (2010). Comparative analysis of biclustering algorithms. In *Proceedings of the first acm international conference on bioinformatics and computational biology* (pp. 265–274).

Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y., & Greenleaf, W. J. (2013). Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, dna-binding proteins and nucleosome position. *Nature methods*, *10*(12), 1213–1218.

Caliński, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, *3*(1), 1–27.

Cao, J., Spielmann, M., Qiu, X., Huang, X., Ibrahim, D. M., Hill, A. J., ... Shendure, J. (2019, February). The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, *566*(7745), 496–502. doi: 10.1038/s41586-019-0969-x

Cauchy, A.-L. (1829). Sur l'équationa l'aide de laquelle on détermine les inégalités séculaires des mouvements des planetes. *Oeuvres Completes (IIeme Série)*, *9*.

Cazes, P., Chouakria, A., Diday, E., & Schektman, Y. (1997). Extension de l'analyse en composantes principales à des données de type intervalle. *Revue de Statistique appliquée*, *45*(3), 5–24.

Chang, Y., Allen, C., Wan, C., Chung, D., Zhang, C., Li, Z., & Ma, Q. (2021). Irisfgm: an integrative single-cell rna-seq interpretation system for functional gene module analysis. *Bioinformatics*, *37*(18), 3045–3047.

Chari, T., & Pachter, L. (2023, 08). The specious art of single-cell genomics. *PLOS Computational Biology*, *19*(8), 1-20. Retrieved from https://doi.org/10.1371/journal.pcbi.1011288 doi: 10.1371/journal.pcbi.1011288

Chen, A., Liao, S., Cheng, M., Ma, K., Wu, L., Lai, Y., ... others (2022a). Spatiotemporal transcriptomic atlas of mouse organogenesis using dna nanoball-patterned arrays. *Cell*, *185*(10), 1777–1792.

Chen, A., Liao, S., Cheng, M., Ma, K., Wu, L., Lai, Y., ... Wang, J.

(2022b). Spatiotemporal transcriptomic atlas of mouse organogenesis using dna nanoball-patterned arrays. *Cell*, *185*(10), 1777-1792.e21. Retrieved from https://www.sciencedirect.com/science/article/pii/S0092867422003993 doi: https://doi.org/10.1016/j.cell.2022.04.003

Cheng, Y., & Church, G. M. (2000). *Biclustering of expression data. in intelligent systems for molecular biology.* Menlo Park: AAAI Press.

Churchill, G. A. (2002). Fundamentals of experimental design for cdna microarrays. *Nature genetics*, *32*(4), 490–495.

Clausen, S.-E. (1998). *Applied correspondence analysis: An introduction* (Vol. 121). Sage.

Darmanis, S., Sloan, S. A., Zhang, Y., Enge, M., Caneda, C., Shuer, L. M., ... Quake, S. R. (2015, June). A survey of human brain transcriptome diversity at the single cell level. *Proceedings of the National Academy of Sciences of the United States of America*, *112*(23), 7285–7290. doi: 10.1073/pnas.1507125112

de Haan, J. R., Wehrens, R., Bauerschmidt, S., Piek, E., Schaik, R. v., & Buydens, L. M. (2007). Interpretation of anova models for microarray data using pca. *Bioinformatics*, *23*(2), 184–190.

Ding, J., Adiconis, X., Simmons, S. K., Kowalczyk, M. S., Hession, C. C., Marjanovic, N. D., ... Levin, J. Z. (2020a, June). Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nature Biotechnology*, *38*(6), 737–746. Retrieved 2022-11-14, from http://www.nature.com/articles/s41587-020-0465-8 (Number: 6 Publisher: Nature Publishing Group) doi: 10.1038/s41587-020-0465-8

Ding, J., Adiconis, X., Simmons, S. K., Kowalczyk, M. S., Hession, C. C., Marjanovic, N. D., ... Levin, J. Z. (2020b, Jun 01). Systematic comparison of single-cell and single-nucleus rna-sequencing methods. *Nature Biotechnology*, *38*(6), 737-746. Retrieved from https://doi.org/10.1038/s41587-020-0465-8 doi: 10.1038/s41587-020-0465-8

Eren, K., Deveci, M., Küçüktunç, O., & Çatalyürek, Ü. V. (2013). A comparative analysis of biclustering algorithms for gene expression data. *Briefings in bioinformatics*, *14*(3), 279–292.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd international conference on knowledge discovery and* (p. 226-231).

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (Vol. 96, pp. 226–231).

Franzén, O., Gan, L.-M., & Björkegren, J. L. M. (2019, 04). PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. *Database*, *2019*. Retrieved from https://doi.org/10.1093/database/baz046 doi: 10.1093/database/baz046

Freytag, S., Tian, L., Lönnstedt, I., Ng, M., & Bahlo, M. (2018, December). *Comparison of clustering tools in R for medium-sized 10x Genomics single-cell RNA-sequencing data* (Tech. Rep. No. 7:1297). F1000Research. Retrieved 2022-11-14, from https://f1000research.com/articles/7-1297 (Type: article) doi: 10.12688/f1000research.15809.2

Fu, L., & Medico, E. (2007). Flame, a novel fuzzy clustering method for the analysis of dna microarray data. *BMC bioinformatics*, *8*(1), 1–15.

Golub, G. H., & Reinsch, C. (1971). Singular value decomposition and least squares solutions. In F. L. Bauer, A. S. Householder, F. W. J. Olver, H. Rutishauser, K. Samelson, & E. Stiefel (Eds.), *Handbook for automatic computation: Volume ii: Linear algebra* (pp. 134–151). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-86940-2_10 doi: 10.1007/978-3-642-86940-2_10

Gralinska, E., Kohl, C., Fadakar, B. S., & Vingron, M. (2022). Visualizing cluster-specific genes from single-cell transcriptomics data using association plots. *Journal of molecular biology*, *434*(11), 167525.

Gralinska, E., & Vingron, M. (2023). Association plots: visualizing cluster-specific associations in high-dimensional correspondence analysis biplots. *Journal of the Royal Statistical Society Series C: Applied Statistics*, qlad039.

Grattan-Guinness, I. (2000). *The rainbow of mathematics: A history of the mathematical sciences*. WW Norton & Company.

Greenacre, M. (2007). Correspondence analysis in practice, chapman & hall. *CRC, Baton Rouge, Florida*.

Greenacre, M. (2017). *Correspondence analysis in practice, third edition* (3rd edition ed.). Chapman & Hall.

Greenacre, M. J. (1984). Theory and applications of correspondence analysis.

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., Zheng, S., Butler, A., ... Satija, R. (2021, June). Integrated analysis of multimodal single-cell data. *Cell*, *184*(13), 3573-3587.e29. doi: 10.1016/j.cell.2021.04.048

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., Zheng, S., Butler, A., ... others (2021). Integrated analysis of multimodal single-cell data. *Cell*, *184*(13), 3573–3587.

Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the american statistical association*, *67*(337), 123–129.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, *28*(1), 100–108.

Hornik, K., Feinerer, I., Kober, M., & Buchta, C. (2012). Spherical k-means clustering. *Journal of statistical software*, *50*, 1–22.

Horta, D., & Campello, R. J. (2014). Similarity measures for comparing biclusterings. *IEEE/ACM transactions on computational biology and bioinformatics*, *11*(5), 942–954.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, *24*(6), 417.

Hsu, L. L., & Culhane, A. C. (2020). Impact of data preprocessing on integrative matrix factorization of single cell data. *Frontiers in Oncology*, 973.

Hsu, L. L., & Culhane, A. C. (2023a). Correspondence analysis for dimension reduction, batch integration, and visualization of single-cell rna-seq data. *Scientific Reports*, *13*(1), 1–17.

Hsu, L. L., & Culhane, A. C. (2023b). Correspondence analysis for dimension reduction, batch integration, and visualization of single-cell rna-seq data. *Scientific Reports*, *13*(1), 1–17.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of classification*, *2*, 193–218.

Hwang, H., Montréal, H., Dillon, W. R., & Takane, Y. (2006). An extension of multiple correspondence analysis for identifying heterogeneous subgroups of respondents. *Psychometrika*, *71*(1), 161–171.

Ingolia, N. T., Ghaemmaghami, S., Newman, J. R., & Weissman, J. S. (2009). Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. *science*, *324*(5924), 218–223.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. 1. *New phytologist*, *11*(2), 37–50.

Johnson, W. E., Li, C., & Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, *8*(1), 118–127.

Jojic, V., Shay, T., Sylvia, K., Zuk, O., Sun, X., Kang, J., ... Koller, D. (2013). Identification of transcriptional regulators in the mouse immune system. *Nature immunology*, *14*(6), 633–643.

Jordan, C. (1874). Mémoire sur les formes bilinéaires. *Journal de mathématiques pures et appliquées*, *19*, 35–54.

Kluger, Y., Basri, R., Chang, J. T., & Gerstein, M. (2003). Spectral biclustering of microarray data: coclustering genes and conditions. *Genome research*, *13*(4), 703–716.

Kodzius, R., Kojima, M., Nishiyori, H., Nakamura, M., Fukuda, S., Tagami, M., ... others (2006). Cage: cap analysis of gene expression. *Nature methods*, *3*(3), 211–222.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, *29*(1), 1–27.

Langovaya, A., Kuhnt, S., & Chouikha, H. (2012). Correspondence analysis in the case of outliers. In *Classification and data mining* (pp. 63–70). Springer.

Lause, J., Berens, P., & Kobak, D. (2021). Analytic pearson residuals for normalization of single-cell rna-seq umi data. *Genome biology*, *22*(1), 1–20.

Lazzeroni, L., & Owen, A. (2002). Plaid models for gene expression data. *Statistica sinica*, 61–86.

Li, G., Ma, Q., Tang, H., Paterson, A. H., & Xu, Y. (2009). Qubic: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic acids research*, *37*(15), e101–e101.

Liu, X., Li, D., Liu, J., Su, Z., & Li, G. (2020). Recbic: a fast and accurate algorithm recognizing trend-preserving biclusters. *Bioinformatics*, *36*(20), 5054–5060.

Luecken, M. D., & Theis, F. J. (2019). Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular Systems Biology*, *15*(6), e8746. Retrieved from https://www.embopress.org/doi/abs/10.15252/msb.20188746 doi: https://doi.org/10.15252/msb.20188746

MacQueen, J. (1967). Classification and analysis of multivariate observations. In *5th berkeley symp. math. statist. probability* (pp. 281–297).

Marbach, D., Costello, J. C., Küffner, R., Vega, N. M., Prill, R. J., Camacho, D. M., ... others (2012). Wisdom of crowds for robust gene network inference. *Nature methods*, *9*(8), 796–804.

Marini, F., & Binder, H. (2019). pcaexplorer: an r/bioconductor package for interacting with rna-seq principal components. *BMC bioinformatics*, *20*, 1–8.

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

McInnes, L., Healy, J., & Melville, J. (2020). UMAP: Uniform manifold approximation and projection for dimension reduction. *arxiv*. Retrieved 2021-04-29, from http://arxiv.org/abs/1802.03426

Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, *98*(5), 873-895. Retrieved from https://www.sciencedirect.com/science/article/pii/S0047259X06002016 doi: https://doi.org/10.1016/j.jmva.2006.11.013

Mirzabekov, A. D., Lysov, Y. P., Shick, V. V., & Dubiley, S. A. (n.d.). Microchip method for the enrichment of specific dna sequences. Retrieved from https://www.osti.gov/biblio/321191

Murali, T., & Kasif, S. (2002). Extracting conserved gene expression motifs from gene

expression data. In *Biocomputing 2003* (pp. 77–88). World Scientific.

Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(1), 86–97.

Murtagh, F., & Contreras, P. (2017). Algorithms for hierarchical clustering: an overview, ii. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(6), e1219.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, *103*(23), 8577–8582.

Nguyen, L. H., & Holmes, S. (2019). Ten quick tips for effective dimensionality reduction. *PLoS computational biology*, *15*(6), e1006907.

Northcutt, A., Kick, D., Otopalik, A., Goetz, B., Harris, R., Santin, J., … Schulz, D. (2019, 12). Molecular profiling of single neurons of known identity in two ganglia from the crab cancer borealis. *Proceedings of the National Academy of Sciences*, *116*, 201911413. doi: 10.1073/pnas.1911413116

Patrikainen, A., & Meila, M. (2006). Comparing subspace clusterings. *IEEE Transactions on Knowledge and Data Engineering*, *18*(7), 902–916.

Paul, F., Arkin, Y., Giladi, A., Jaitin, D. A., Kenigsberg, E., Keren-Shaul, H., … others (2015). Transcriptional heterogeneity and lineage commitment in myeloid progenitors. *Cell*, *163*(7), 1663–1677.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, *2*(11), 559–572.

Pease, A. C., Solas, D., Sullivan, E. J., Cronin, M. T., Holmes, C. P., & Fodor, S. (1994). Light-generated oligonucleotide arrays for rapid dna sequence analysis. *Proceedings of the National Academy of Sciences*, *91*(11), 5022–5026.

Pollack, J. R., Perou, C. M., Alizadeh, A. A., Eisen, M. B., Pergamenschikov, A., Williams, C. F., … Brown, P. O. (1999). Genome-wide analysis of dna copy-number changes using cdna microarrays. *Nature genetics*, *23*(1), 41–46.

Pons, P., & Latapy, M. (2005). Computing communities in large networks using random walks. In *Computer and information sciences-iscis 2005: 20th international*

symposium, istanbul, turkey, october 26-28, 2005. proceedings 20 (pp. 284–293).

Pontes, B., Giráldez, R., & Aguilar-Ruiz, J. S. (2015). Biclustering on expression data: A review. *Journal of biomedical informatics*, *57*, 163–180.

Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., ... Zitzler, E. (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, *22*(9), 1122–1129.

Ramesh, A., Schubert, R. D., Greenfield, A. L., Dandekar, R., Loudermilk, R., Sabatino, J. J., ... Wilson, M. R. (2020). A pathogenic and clonally expanded b cell transcriptome in active multiple sclerosis. *Proceedings of the National Academy of Sciences*, *117*(37), 22932-22943. Retrieved from https://www.pnas.org/doi/abs/10.1073/pnas.2008523117 doi: 10.1073/pnas.2008523117

Rao, A., Barkley, D., França, G. S., & Yanai, I. (2021). Exploring tissue architecture using spatial transcriptomics. *Nature*, *596*(7871), 211–220.

Ray, S., & Turi, R. H. (1999). Determination of number of clusters in k-means clustering and application in colour image segmentation. In *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques* (Vol. 137, p. 143).

Řeháková, B. (1986). *Multivariate descriptive statistical analysis (mnohorozměrná deskriptivní statistická analỳza)*. JSTOR.

Robertson, G., Hirst, M., Bainbridge, M., Bilenky, M., Zhao, Y., Zeng, T., ... others (2007). Genome-wide profiles of stat1 dna association using chromatin immunoprecipitation and massively parallel sequencing. *Nature methods*, *4*(8), 651–657.

Rosebrock, D., Arora, S., Mutukula, N., Volkman, R., Gralinska, E., Balaskas, A., ... Elkabetz, Y. (2022). Enhanced cortical neural stem cell identity through short SMAD and WNT inhibition in human cerebral organoids facilitates emergence of outer radial glial cells. *Nature Cell Biology*, *24*(6), 981–995. Retrieved 2022-07-01, from https://www.nature.com/articles/s41556-022-00929-5 doi: 10.1038/s41556-022-00929-5

Rotival, M., Zeller, T., Wild, P. S., Maouche, S., Szymczak, S., Schillert, A., ... others (2011). Integrating genome-wide genetic variations and monocyte expression data reveals trans-regulated gene modules in humans. *PLoS genetics*, *7*(12),

e1002367.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Roy, S., Lagree, S., Hou, Z., Thomson, J. A., Stewart, R., & Gasch, A. P. (2013). Integrated module and gene-specific regulatory inference implicates upstream signaling networks. *PLoS computational biology*, *9*(10), e1003252.

Saelens, W., Cannoodt, R., & Saeys, Y. (2018). A comprehensive evaluation of module detection methods for gene expression data. *Nature communications*, *9*(1), 1090.

Saelens, W., Cannoodt, R., Todorov, H., & Saeys, Y. (2019). A comparison of single-cell trajectory inference methods. *Nature biotechnology*, *37*(5), 547–554.

Shalon, D., Smith, S. J., & Brown, P. O. (1996). A dna microarray system for analyzing complex dna samples using two-color fluorescent probe hybridization. *Genome research*, *6*(7), 639–645.

Sill, M., Kaiser, S., Benner, A., & Kopp-Schneider, A. (2011). Robust biclustering by sparse singular value decomposition incorporating stability selection. *Bioinformatics*, *27*(15), 2089–2097.

Slovin, S., Carissimo, A., Panariello, F., Grimaldi, A., Bouché, V., Gambardella, G., & Cacchiarelli, D. (2021). Single-cell rna sequencing analysis: a step-by-step overview. *RNA Bioinformatics*, 343–365.

Soon, W. W., Hariharan, M., & Snyder, M. P. (2013). High-throughput sequencing for biology and medicine. *Molecular Systems Biology*, *9*(1), 640. Retrieved from https://www.embopress.org/doi/abs/10.1038/msb.2012.61 doi: https://doi.org/10.1038/msb.2012.61

Steinley, D. (2004). Properties of the hubert-arable adjusted rand index. *Psychological methods*, *9*(3), 386.

Stewart, G. W. (1993). On the early history of the singular value decomposition. *SIAM review*, *35*(4), 551–566.

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W. M., … Satija, R. (2019). Comprehensive integration of single-cell data. *Cell*, *177*(7), 1888–1902.

Sugar, C. A., & James, G. M. (2003). Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 750–763.

TAUB, E., FLOYD, DeLEO, J. M., & Thompson, E. B. (1983). Sequential comparative hybridizations analyzed by computerized image processing can identify and quantitate regulated rnas. *Dna*, *2*(4), 309–327.

Ter Braak, C. J., & Verdonschot, P. F. (1995). Canonical correspondence analysis and related multivariate methods in aquatic ecology. *Aquatic sciences*, *57*, 255–289.

THE TABULA SAPIENS CONSORTIUM. (2022, May). The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, *376*(6594), eabl4896. Retrieved 2022-11-14, from https://www.science.org/doi/10.1126/science.abl4896 doi: 10.1126/science.abl4896

Tirosh, I., Izar, B., Prakadan, S. M., Wadsworth, M. H., Treacy, D., Trombetta, J. J., ... Garraway, L. A. (2016, April). Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science (New York, N.Y.)*, *352*(6282), 189–196. doi: 10.1126/science.aad0501

Townes, F. W., Hicks, S. C., Aryee, M. J., & Irizarry, R. A. (2019). Feature selection and dimension reduction for single-cell rna-seq based on a multinomial model. *Genome biology*, *20*, 1–16.

Traag, V. A., Waltman, L., & Van Eck, N. J. (2019). From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, *9*(1), 5233.

Tsuyuzaki, K., Sato, H., Sato, K., & Nikaido, I. (2020). Benchmarking principal component analysis for large-scale single-cell rna-sequencing. *Genome biology*, *21*(1), 1–17.

Van Buuren, S., & Heiser, W. J. (1989). Clustering n objects into k groups under optimal scaling of variables. *Psychometrika*, *54*, 699–706.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(86), 2579–2605. Retrieved from http://jmlr.org/papers/v9/vandermaaten08a.html

Van de Velden, M., D'Enza, A. I., & Palumbo, F. (2017). Cluster correspondence analysis. *Psychometrika*, *82*, 158–185.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, *17*, 395–416.

Waern, K., Nagalakshmi, U., & Snyder, M. (2011). Rna sequencing. *Yeast Systems Biology: Methods and Protocols*, 125–132.

Wang, M., Hu, Q., Lv, T., Wang, Y., Lan, Q., Xiang, R., ... Liu, L. (2022). High-resolution 3d spatiotemporal transcriptomic maps of developing drosophila embryos and larvae. *Developmental Cell*, *57*(10), 1271-1283.e4. Retrieved from https://www.sciencedirect.com/science/article/pii/S1534580722002465 doi: https://doi.org/10.1016/j.devcel.2022.04.006

Wang, Z., Li, G., Robinson, R. W., & Huang, X. (2016). Unibic: Sequential row-based biclustering algorithm for analysis of gene expression data. *Scientific reports*, *6*(1), 1–10.

Wolf, F. A., Angerer, P., & Theis, F. J. (2018a). SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, *19*(1), 15. Retrieved 2021-04-16, from https://doi.org/10.1186/s13059-017-1382-0 doi: 10.1186/s13059-017-1382-0

Wolf, F. A., Angerer, P., & Theis, F. J. (2018b). Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, *19*, 1–5.

Wolf, F. A., Hamey, F. K., Plass, M., Solana, J., Dahlin, J. S., Göttgens, B., ... Theis, F. J. (2019). Paga: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome biology*, *20*, 1–9.

Xie, J., Ma, A., Zhang, Y., Liu, B., Cao, S., Wang, C., ... Ma, Q. (2020). Qubic2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale rna-seq data. *Bioinformatics*, *36*(4), 1143–1149.

Yang, X., & Vingron, M. (2018). Classifying human promoters by occupancy patterns identifies recurring sequence elements, combinatorial binding, and spatial interactions. *BMC biology*, *16*, 1–19.

Yosef, N., Shalek, A. K., Gaublomme, J. T., Jin, H., Lee, Y., Awasthi, A., ... others (2013). Dynamic regulatory network controlling th17 cell differentiation. *Nature*, *496*(7446), 461–468.

Zeisel, A., Muñoz-Manchado, A. B., Codeluppi, S., Lönnerberg, P., La M-anno, G., Juréus, A., ... Linnarsson, S. (2015, March). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, *347*(6226), 1138–1142. Retrieved 2022-07-26, from https://www.science.org/doi/full/10.1126/science.aaa1934 doi: 10.1126/science.aaa1934

Zhao, J., Jaffe, A., Li, H., Lindenbaum, O., Sefik, E., Jackson, R., ... Kluger, Y. (2021). Detection of differentially abundant cell subpopulations in scrna-seq data. *Proceedings of the National Academy of Sciences*, *118*(22), e2100293118.

A | Summary

Cell clustering is a crucial step in current single-cell RNA sequencing (scRNA-seq) methods, where marker genes are identified and used for cell type annotation. However, this process can be time-consuming and laborious. To address this, biclustering algorithms have been developed to simultaneously identify functional gene sets and cell clusters. However, most existing biclustering algorithms are designed for microarray and bulk RNA sequencing data, and only a few are suitable for scRNA-seq analysis. These algorithms often suffer from issues such as limited scalability and accuracy. In this study, we propose *Correspondence Analysis based biclustering on Networks (CAbiNet)*, a graph-based biclustering approach specifically designed for scRNA-seq data. CAbiNet integrates multiple analysis steps by efficiently co-clustering cells and their marker genes, and visualizing the biclustering results in a non-linear embedding. We introduce two visualization approaches that enable the joint display of genes and cells in a two-dimensional space. Additionally, a random forest regression model is trained to predict the quality of clustering results, facilitating the selection of optimal parameters. CAbiNet fills the gap for a high-performing biclustering algorithm in scRNA-seq and spatial transcriptomics data analysis. It streamlines existing workflows and offers an intuitive and interactive visual exploration of cells and their marker genes in a single plot for efficient cell type annotation. CAbiNet is available as an R package on GitHub at https://github.com/VingronLab/CAbiNet.

B | Zusammenfassung

Das Clustering von Zellen ist ein entscheidender Schritt bei den derzeitigen Methoden der Einzelzell-RNA-Sequenzierung (scRNA-seq), bei denen Markergene identifiziert und zur Annotation von Zelltypen verwendet werden. Dieser Prozess kann jedoch zeitaufwändig und mühsam sein. Aus diesem Grund wurden Biclustering-Algorithmen entwickelt, um gleichzeitig funktionale Gensätze und Zellcluster zu identifizieren. Die meisten vorhandenen Biclustering-Algorithmen sind jedoch für Mikroarray- und Massen-RNA-Sequenzierungsdaten konzipiert, und nur wenige sind für die scRNA-seq-Analyse geeignet. Diese Algorithmen leiden oft unter Problemen wie begrenzter Skalierbarkeit und Genauigkeit. In dieser Studie schlagen wir Correspondence Analysis based biclustering on Networks (CAbiNet) vor, einen graphbasierten Biclustering-Ansatz, der speziell für scRNA-seq-Daten entwickelt wurde. CAbiNet integriert mehrere Analyseschritte durch effizientes Co-Clustering von Zellen und ihren Markergenen und visualisiert die Biclustering-Ergebnisse in einer nichtlinearen Einbettung. Wir stellen zwei Visualisierungsansätze vor, die die gemeinsame Darstellung von Genen und Zellen in einem zweidimensionalen Raum ermöglichen. Zusätzlich wird ein Random-Forest-Regressionsmodell trainiert, um die Qualität der Clustering-Ergebnisse vorherzusagen, was die Auswahl der optimalen Parameter erleichtert. CAbiNet füllt die Lücke für einen leistungsstarken Biclustering-Algorithmus in der scRNA-seq- und räumlichen Transkriptomik-Datenanalyse. Es rationalisiert bestehende Arbeitsabläufe und bietet eine intuitive und interaktive visuelle Erkundung von Zellen und ihren Markergenen in einem einzigen Diagramm für eine effiziente Zelltyp-Annotation. CAbiNet ist als R-Paket auf GitHub unter https://github.com/VingronLab/CAbiNet verfügbar.

C | Selbstständigkeitserklärung

Name: Zhao

Vorname: Yan

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht.

Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einver- standen.

Berlin, 2024 Yan Zhao