Improvement on the explanation of cluster analysis for transcriptome data

March 2021

Momo Shirotori

Improvement on the explanation of cluster analysis for transcriptome data

Graduate School of Systems and Information Engineering

University of Tsukuba

March 2021

Momo Shirotori

Abstract

In this dissertation, we present the result of the study on improving the explanation of cluster analysis for transcriptome data. To improve the explanation of cluster analysis, it is necessary to perform accurate cluster analysis and select features that contribute to cluster separation. In cluster analysis that integrates multiple instances of transcriptome data, the batch effect deteriorates the analysis performance. To perform highly accurate cluster analysis, we propose a batch effect correction method based on the dimensionality reduction method. In addition, the transcriptome data have numerous features; that is, it is difficult to explain and interpret the cluster analysis results. Accordingly, we also propose a feature selection method that is useful for cluster analysis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, with the development of measuring devices such as next-generation sequencers and mass spectrometry, large amounts of high-dimensional data are being obtained. Therefore, there is a strong demand for effective data analysis technology for such data. Transcriptome data, which is a type of high-dimensional data, comprise data that comprehensively grasp the transcriptomes and measure the expression level of genes; consequently, they contain significant gene information as a feature. Decoding the structure of genetic information leads to the elucidation of various life phenomena, and an information science approach is used to acquire the life phenomena embedded in the data. In cluster analysis for transcriptome data, the results of cluster analysis are implicated using features, and it the differences between cell populations and their functional characteristics can be discovered by explaining and interpreting the characteristics of each cluster after analysis.

This chapter is organized as follows. The differences between the two types of transcriptome data are presented in Section 1.1. An overview of transcriptome analysis is presented in Section 1.2. We introduce the batch effect correction and feature selection methods in Section 1.3 and Section 1.4, respectively. Section 1.6 summarizes the contributions of this study. Section 1.7 illustrates the organization of the dissertation.

1.1 Two Types of Transcriptome Data

The concept that the genetic information of an organism is transmitted in the order of 「DNA → (transcription) → mRNA → (translation) → protein」 is called central dogma in molecular biology, and gene expression refers to the production of proteins from genetic information (see Figure 1.1). Transcriptome refers to the total amount of all mRNAs present in a cell under a specific situation, and the expression level of each gene can be quantified as the amount of transcription in an mRNA. The gene expression status under various conditions can be determined by measuring the expression level of mRNA in cells, which leads to the discovery of cell function.

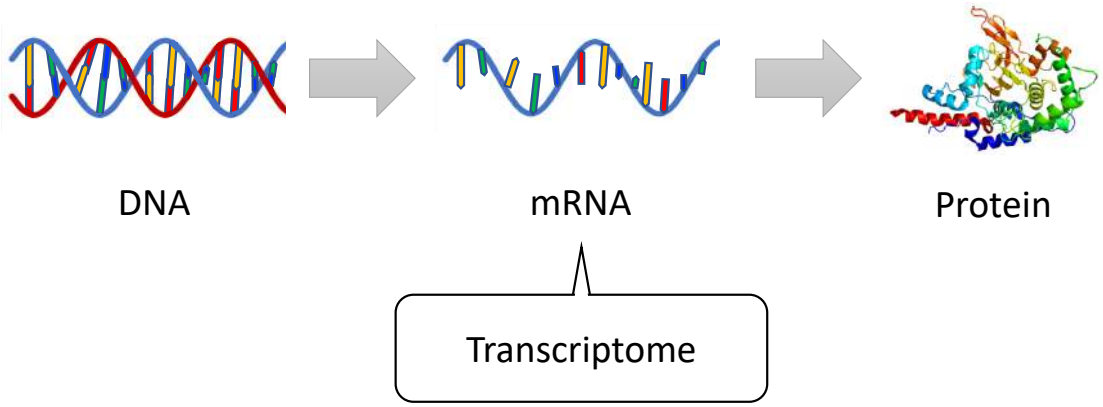


Figure 1.1: Overview of central dogma

Transcriptome measurement methods are roughly divided into microarrays and next-generation sequencers (NGS), and transcriptome analysis by NGS is called RNA-sequencing (RNA-seq) analysis. In recent years, a technique for detecting transcriptome on a cell-by-cell basis has been established, and single-cell RNA-sequencing (scRNA-seq) analysis, which analyzes the gene expression level of each cell, is garnering significant attention. Figure 1.2 presents an overview of the differences between RNA-seq data and scRNA-seq data.

RNA-seq data include data that measure the average gene expression level in all cells

and can identify the differences in samples under various conditions. In constant, scRNA-seq data comprise data that measure the gene expression level of individual cells; they involve several measurement protocols such as SMART-seq2 [1], CEL-seq [2], and Drop-seq [3], among others [4, 5, 6, 7, 8]. scRNA-seq analysis enables cell-specific analysis such as cell type identification [9] and intercellular gene control network inference [10, 11]. The measurement cost continues to decrease per year, while the number of cells that can be analyzed is increasing; however, computational analysis and the interpretability of the analysis results encounter various challenges [12]. scRNA-seq analysis is difficult owing to several parameters, such as high dimensionality, measurement noise, and differences in the sample size between rare and abundant cell populations [13]. One of the important characteristics of scRNA-seq data is a phenomenon called “dropout”. In this phenomenon, when a gene is observed in a cell at a low or moderate expression level, it may not be detected in another cell of the same cell type [14]. This dropout phenomenon occurs due to the low amount of mRNA in individual cells and the inefficient capture of mRNA, which can result in sparse data. However, some of these issues can be alleviated through proper normalization and corrections. In scRNA-seq analysis, the RNA-seq analysis methods can be used; however, in most case, the development of new methods is required.

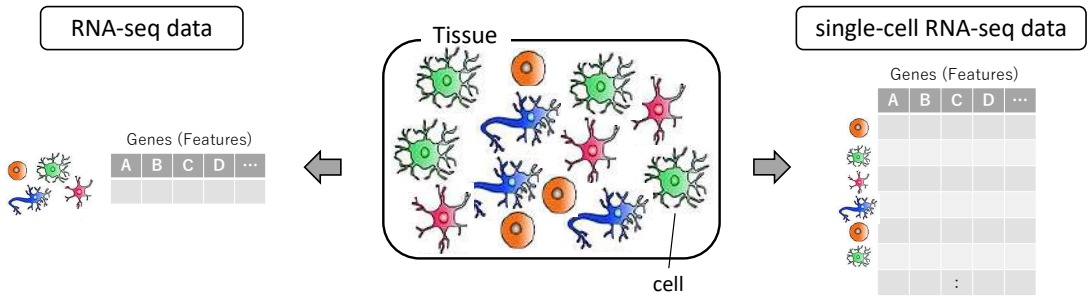


Figure 1.2: Differences between RNA-seq data and scRNA-seq data

1.2 Cluster Analysis for Transcriptome Data

In the general flow of transcriptome analysis, first, the transcriptome data are generated from a sample (for example, blood, tumor) obtained from a tissue using an analytical instrument. Subsequently, differential expression analysis [15, 16, 17, 18, 19, 20], pathway analysis [21], gene network analysis [22, 23], and cluster analysis [24, 25] are performed on the obtained transcriptome data (Figure 1.3). In this study, we focus on cluster analysis for transcriptome data.

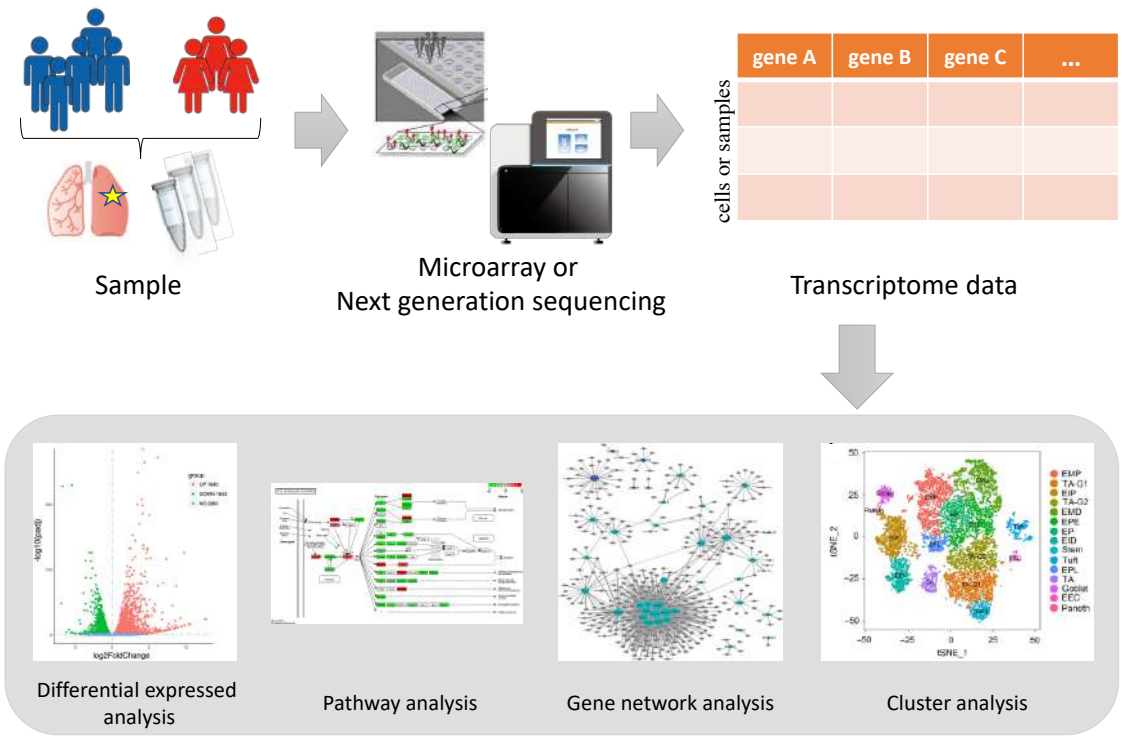


Figure 1.3: Workflow of transcriptome analysis

In cluster analysis of transcriptome data, the cell or gene groups are identified based on the transcriptome similarity without prior knowledge such as cluster information. In most cases, the number of clusters is unknown, and the transcriptome data, which are high-dimensional data, contain technical and biological noise, thereby, making cluster analysis even more difficult.

For the cluster analysis of high-dimensional data, the application of dimensionality reduction techniques may be beneficial. In many cases, the noise contained in the features can be significantly reduced and the data can be visualized in a two-dimensional or three-dimensional subspace by projecting them into a low-dimensional subspace. To accomplish this, dimensionality reduction methods such as PCA [26], t-SNE [27], and UMAP [28] are often used.

Cluster analysis is principally of two types, namely, sample and gene clustering; they are performed using methods such as k-means and DBSCAN [29]. In gene cluster analysis, the function of unknown genes can be estimated by grouping genes with similar expression patterns according to various conditions of tissue samples based on the sample space. Spellman, P. T. et al., [30] identified approximately 800 genes related to cell cycles and the genes expressed in each phase of the cell cycle. However, in the sample cluster analysis, the tissue state and disease can be classified by grouping the tissue samples according to the gene expression pattern based on the feature space [31, 32, 33, 34, 35, 36, 37, 38, 39]. In particular, sample cluster analysis for scRNA-seq data enables the classification of cell types, leading to the identification of unknown cells and estimation of the cell population function [40, 41, 42, 43, 44, 45, 46].

Furthermore, because the explanation of the cluster analysis result is strongly required, the genes effective for the identification of each cluster obtained by performing cluster analysis on the sample are identified. Genes that work functionally on the cell type of each cluster can be identified by identifying the genes whose expression decreases or increases in each cluster and those that work specifically on each cluster of disease and non-disease groups.

There are two important concepts associated with the explanation of the cluster analysis results, i.e., the batch effect and feature selection. They are described in the following sections.

1.3 Batch Effect Correction

The transcriptome data generated at different instances or at laboratories (batch) have differences between the data. This is called the batch effect; each data instance has its own batch effect. Figure 1.4 shows an example of the batch effect. If we directly combine batch 1 data (green) and batch 2 data (blue), the samples that are considered to be of the same cell type are separated between batches in a two-dimensional space, which is visualized as the sample clustering results.

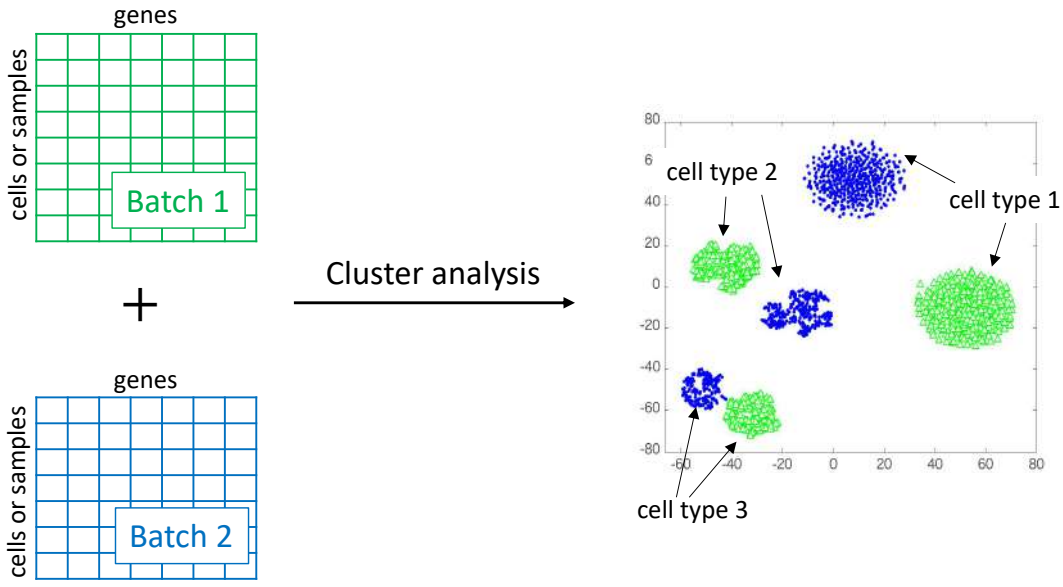


Figure 1.4: Example of batch effect

Thus, the batch effect can introduce incorrect structures into the data and hide the underlying biological knowledge [47]. Correspondingly, the performance of cluster analysis may deteriorate when it is performed by integrating a plurality of transcriptome data. In order to improve the explanation of the cluster analysis results, it is necessary to remove the batch effect and perform accurate cluster analysis before proceeding with the analysis.

1.3.1 Related works

Several methods have been proposed for data merging and comparison in the presence of batch effect using linear models [48, 49, 50, 47, 51]. The combat method [52] is a type of batch effect correction method for RNA-seq data. It estimates the blocking coefficient by sharing information across genes to stabilize the estimates in the presence of limited replicates using a location and scale (L/S) model, which performs scaling adjustments for each gene [53].

The L/S model assumes that the batch effect can be modeled by standardizing the means (location) and variance (scales) across batches. Let $X^{(1)} \in \mathbb{R}^{n_1 \times m}$ and $X^{(2)} \in \mathbb{R}^{n_2 \times m}$ denote the datasets in different batches, where n_i is the number of samples for batch i $(i = 1, 2)$ and m is the number of genes. Owing to the batch effect, $X^{(1)}$ and $X^{(2)}$ cannot be directly merged for further analysis. Thus, the objective is to merge $X^{(1)}$ and $X^{(2)}$ into a dataset, $X^* \in \mathbb{R}^{(n_1+n_2)\times m}$, by correcting the batch effect. Let $X = \begin{bmatrix} X^{(1)} \\ X^{(2)} \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times m}$ be the directly merged data and X_g be the gth column of X. Using the L/S model, X_g is represented as

$$X_g^* = \hat{\alpha}_g + \Phi\hat{\beta}_g + \frac{1}{\hat{\delta}_{i,g}}\left(X_g - \left(\hat{\alpha}_g + \Phi\hat{\beta}_g + \hat{\gamma}_{i,g}\right)\right)$$

where $\hat{\alpha}_g$,$\hat{\beta}_g$, and $\hat{\delta}_g$ are the estimated parameter vectors for gene g. The additional term $\hat{\alpha}_g + \Phi\hat{\beta}_g - \left(\hat{\alpha}_g + \Phi\hat{\beta}_g + \hat{\gamma}_{i,g}\right)/\hat{\delta}_{i,g}$ and multiplying term $\hat{\delta}_{i,g}$ indicate the location and scale adjustments, respectively.

1.4 Feature Selection

High-dimensional data such as transcriptome data encounter challenges such as an increase in the amount of calculation and deterioration of analysis accuracy due to numerous explanatory variables. Therefore, in the analysis of high-dimensional data, the features used in the analysis are reduced by only selecting some features. However, the original

data information may be lost, thereby reducing the accuracy of analysis; thus, it is difficult to make significant reductions while maintaining the accuracy.

We assume that transcriptome data have two types of features, first, the features that can be uniformly expressed throughout the sample (background noise) and second, those that are often expressed in some samples. Figure 1.5 illustrates the differences between the two types of features. The horizontal and vertical axes of the figure denote the index of the sample and the values of the expression level of a specific feature, respectively. The color bar at the bottom denotes the cluster information of the samples. The left figure is an example of the features that are highly expressed in some samples; it can be observed that the features are highly expressed in the samples that belong to the blue cluster. The right figure is an example of background noise. The background noise represents a feature such that the number of expressed samples in any interval does not change.

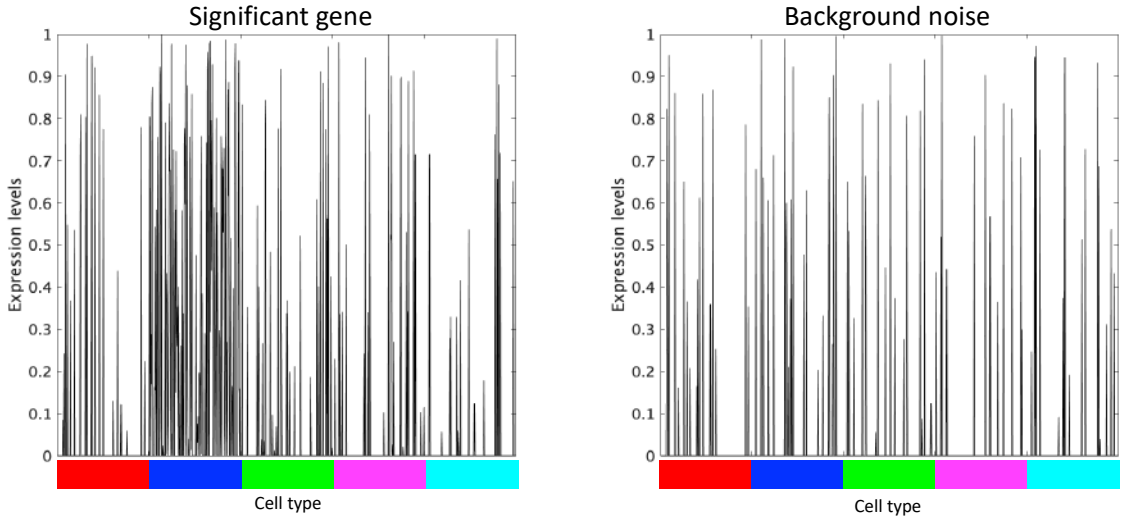


Figure 1.5: Differences between (left) significant gene for clustering and (right) background noise.

In cluster analysis, it is sufficient to use the features that are highly expressed in some samples. The analysis results can be evaluated from a small number of features by only selecting the features required for cluster analysis, which improves the explanation of the

analysis results.

1.4.1 Related works

Several feature selection methods are used for the transcriptome data [54, 55, 56, 57, 58, 59]. These approaches involve the selection of variable features based on the appearance of all samples (i.e., features with large variance) using regression models. The Seurat method [60, 61] is a package in R language that can perform integrated analysis and cluster analysis simultaneously; it has several tutorials and has been widely used in recent years.

The Seurat method applies a variance-stabilizing transformation to correct the mean-variance relationship [62]. It computes the log-transformed mean and variance values for each feature to learn those relationships. Then, it fits a curve to predict the variance of each feature by locally weighted scatter plot smooth. Given the expected variances, it performs transformation as follows:

$$z_{i,j} = \frac{x_{i,j} - \bar{x}_i}{\sigma_i},$$

where $z_{i,j}$ is the standardized value of feature i in cell j, $x_{i,j}$ is the raw value of feature i in cell j, $\bar{x}_i$ is the mean raw value of feature i, and σ_i is the expected standard deviation of feature i. Subsequently, for each feature, it computes the variance of standardized values $\hat{\sigma}_i$ across all cells.

$$\hat{\sigma_i} = \frac{1}{n}\sum_{j=1}^{n}(z_{i,j} - \mu_i)^2,$$

where $\mu_i = \frac{1}{n}\sum_{j=1}^{n} z_{i,j}$. This variance represents a measure of single cell dispersion after controlling for mean expression and it selects the features directly to rank them.

1.5 Dissertation Objectives and Motivations

The objective of this study is to improve the explanation of cluster analysis for transcriptome data. The transcriptome data are high-dimensional data that contain technical and

biological noise, making cluster analysis more difficult. There are two motivations of this study.

Motivation 1: Batch effect correction

The conventional L/S model was performed for RNA-seq data; it multiplies and adds the constant values to normalize the mean and variance of each feature. It is based on the assumption that the composition of a cell population is the same across batches. However, the general scRNA-seq data have different cell population compositions between batches; therefore, the method developed for RNA-seq data cannot be applied. In this study, we consider removing the batch effect in a low-dimensional space after dimensionality reduction by adjusting the original data to develop a batch effect correction method that can be applied to both RNA-seq and scRNA-seq data. We assume that the batch effect can be expressed by a constant multiple of a feature; moreover, it should be noted that most features only contain technical noise.

Motivation 2: Feature selection method

The feature selection method developed for RNA-seq data uses a linear regression model for the relationship between the mean and squared values of the coefficient variation for each feature and selects the features whose original variance value is larger than the estimated variance value. However, for scRNA-seq data, the variance cannot be used as a direct indicator of feature selection because these data contain positive correlations between the mean and variance of the expression level; additionally, various analysis tools have been developed to solve this problem of heteroscedasticity. However, it is difficult to significantly reduce the features while maintaining the accuracy of cluster analysis because significant amount of background noise gets selected. In this study, we attempt to distinguish between the features required for cluster analysis and the background noise without using the mean or variance values. We focus on the characteristic of the background noise that the number of expressed samples in any interval does not change.

1.6 Our Contribution

In this study, we consider two important aspects to improve the explanation of cluster analysis for transcriptome data: (1) Performing accurate cluster analysis and (2) selecting features that contribute to cluster separation. Accordingly, we propose two algorithms.

1.6.1 Scaling method for batch effect correction based on spectral clustering

In Chapter 2, we propose an effective scaling method to remove the batch effect. The proposed method performs scaling adjustment for each feature and it can remove the batch effect on manifold space after dimensionality reduction such as that in spectral clustering (SC). The contributions of the proposed method are summarized as follows. (1) The proposed method scales each feature by multiplying a constant value to ensure that the data samples from different batches resemble each other. (2) We formulate an optimization problem based on SC to obtain the scaling adjustment values. (3) We propose an approximation solution to solve the optimization problem and demonstrate how to determine the parameters. The proposed method is evaluated based on both artificial and gene expression datasets. For the gene expression datasets, we used the microarray and single-cell RNA-seq datasets. The results of numerical experiments verified that the proposed method outperforms the existing well-established batch effect correction methods on both microarray and single-cell datasets.

1.6.2 Feature selection based on principal component analysis of sample space

In Chapter 3, we propose an effective feature selection method to analyze sample clustering. The contributions of the proposed method are summarized as follows. (1) It can distinguish between the features required for clustering and the background noise by per-

forming principal component analysis (PCA), a dimensionality reduction method for the sample space. (2) The proposed method selects only significant features for clustering analysis; i.e., it removes the background noise from the larger subset features. The proposed method is evaluated based on both simple simulation and scRNA-seq datasets. The results of numerical experiments verified that it can remove the background noise from larger subset features while maintaining clustering accuracy.

1.7 Dissertation Organization

This dissertation is organized as follows. This chapter presents the background and an overview of our research. In Chapter 2, we propose the scaling method for batch effect correction based on SC. In Chapter 3, we propose feature selection method based on PCA of the sample space. Finally, Chapter 4 concludes this dissertation and presents the future work directions.

Chapter 2

Scaling method for batch effect correction based on spectral clustering

In this chapter, we consider performing accurate cluster analysis to improve the explanation of cluster analysis and present a novel scaling method for batch effect correction, referred to as the SMSC method. We focus on the scaling adjustment of the batch effects. The proposed method performs scaling adjustment for each feature and can remove the batch effects on the manifold space after dimensionality reduction such as that in spectral clustering (SC).

This chapter is organized as follows. In Section 2.1, we introduce the SC method. In Section 2.2, we give an overview of the proposed SMSC method. In Section 2.3, we present an approximation solution to solve the optimization problem. In Section 2.4 and Section 2.5, we demonstrate how to determine the two kinds of parameters. Performance analysis and comparison are presented in Section 2.6. Section 2.7 concludes this chapter.

2.1 Spectral Clustering

The SC method first performs nonlinear dimensionality reduction, followed by clustering in the low-dimensional space. SC is induced by undirected graph partitions, where the

graph has the edge weight $w_{i,j}$ between nodes i and j. Let $W = \{w_{i,j}\} \in \mathbb{R}^{n\times n}$, $D = \text{diag}\,(d_1, d_2, \ldots, d_n) \in \mathbb{R}^{n\times n}$ with $d_i = \sum_{j=1}^{n} w_{i,j}$, and $\mathbf{1}_n$ be the n-dimensional vector with one in all entries. Then, the graph partitioning problem is solved by the constrained minimization problem of the normalized cut (Ncut) function [63], as follows.

$$\min_{\boldsymbol{v}} \frac{\boldsymbol{v}^{\mathsf{T}}\,(D-W)\,\boldsymbol{v}}{\boldsymbol{v}^{\mathsf{T}} D\boldsymbol{v}}, \quad \text{subject to} \quad \mathbf{1}_n^{\mathsf{T}} D\boldsymbol{v} = 0, \tag{2.1}$$

where $\boldsymbol{v}$ is a label vector with entries denoting the sample labels. In [63], $\boldsymbol{v}$ is set such as $v_i \in \{1,\ -b\}$ with $b = \sum_{i:t_i>0} d_i / \sum_{i:t_i<0} d_i$, where $t_i \in \{\pm 1\}$ is an indicator. Based on v_i, the data samples are divided into two clusters. This discrete problem can be relaxed to find the Fielder vector $\boldsymbol{v} \in \mathbb{R}^n$[64] associated with the second smallest eigenvalue of the constrained generalized eigenvalue problem

$$L\boldsymbol{v} = \lambda D\boldsymbol{v}, \quad \text{subject to} \quad \mathbf{1}_n^{\mathsf{T}} D\boldsymbol{v} = 0,$$

where $L = D - W \in \mathbb{R}^{n\times n}$ is the Laplacian matrix and $\lambda \in \mathbb{R}$. The entries of the Fiedler vector are referred to as the coordinates of the data samples in the reduced space. The first $S\ell$ eigenvectors represent the reduced low-dimensional space. SC performs k-means clustering in the low-dimensional space $[\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_\ell]$.

2.2 The Framework of SMSC Method

The objective of SMSC method is to merge the two datasets $X^{(1)} \in \mathbb{R}^{n_1\times m}$ and $X^{(2)} \in \mathbb{R}^{n_2\times m}$ into a comparable dataset using the diagonal matrix $S = \text{diag}\,(s_1, s_2, \ldots, s_m) \in \mathbb{R}^{m\times m}$, such as

$$X^* = \phi\left(\begin{bmatrix} X^{(1)} \\ X^{(2)}S \end{bmatrix}\right),$$

where $X^* \in \mathbb{R}^{(n_1+n_2)\times \ell}$ is a merged dataset with corrected batch effects in the low-dimensional space, ϕ is the nonlinear mapping function, and $\ell \leq (n_1 + n_2)$ is the number of reduced

dimensions. The proposed method removes the batch effects on the manifold space after dimensionality reduction by SC.

SC is a problem of finding the Fiedler vector $\boldsymbol{v}$ using the matrices L and D, where the element $v_i \in \{1,\ -b\}$ has the cluster label information. In our method, we give the batch information to the Fiedler vector $\boldsymbol{v}$ as the cluster label information. Then, we modify the matrices L and D based on the given Fiedler vector $\boldsymbol{v}$. Since the matrices L and D after batch effects correction are unknown, we use the approximate form $\min_{L,D,\lambda,\boldsymbol{v}} \|L\boldsymbol{v} - \lambda D\boldsymbol{v}\|_2^2$. We modify the matrices L and D using the diagonal matrix S, and proposed an approximation solution to find the matrix S in Section 2.3.

We consider using the batch information as the cluster label information, i.e., batch 1 and batch 2 correspond to cluster 1 and cluster 2, respectively. The labels are set to 1 and $-b$, respectively. Therefore, we can estimate the Fiedler vector $\boldsymbol{v}$ using a discrete variable in (2.1) such that $\boldsymbol{v} = [1, 1, \ldots, 1, -b, -b, \ldots, -b] \in \mathbb{R}^{(n_1+n_2)}$, where b is the parameter, which will be discussed in Section 2.4. If the data in each batch are directly merged in a low-dimensional space using the given Fiedler vector $\boldsymbol{v}$, there exists a matrix S such that $\min_{s,\lambda_s} \|L_s\boldsymbol{v} - \lambda_s D_s\boldsymbol{v}\|_2^2$, where L_s, D_s and λ_s are calculated after batch effect correction based on the scaling adjustment.

The proposed method aims to find the matrix S to remove the batch effects to ensure that the data in two batches are similar to each other. Therefore, we solve the following equation:

$$\max_{s,\lambda_s} \|L_s\boldsymbol{v} - \lambda_s D_s\boldsymbol{v}\|_2^2. \tag{2.2}$$

This means that there exists a matrix S such that the data samples from different batches can be adjusted to resemble each other in a low-dimensional space.

In general, we assume that we have N datasets $X^{(1)}, X^{(2)}, \ldots, X^{(N)}$ in different batches. Without loss of generality, we define $X^{(1)}$ as the reference data C. Then, we correct the batch effects between the reference data C and the next dataset $X^{(2)}$, and merge data C and $X^{(2)}$ into a dataset by correcting the batch effects. Next, we update

the reference data as the corrected merged data, followed by correcting the batch effects between the updated reference data and dataset $X^{(3)}$. These steps are repeated until the update reference data are merged with the final dataset $X^{(N)}$.

2.3 Solution

We propose an effective method to approximate the solution of (2.2) for the scaling adjustment values. We use the Gaussian similarity function to calculate the graph weight $w_{i,j}$ as $w_{i,j} = \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2/2\sigma^2\right)$, where $\boldsymbol{x}_i$ is the ith row of X and σ is a parameter. We will discuss how to set the parameter σ in Section 2.5. We can formulate (2.2) as the following optimization problem for the scaling adjustment values $s_i\,(i = 1, 2, \ldots, m)$:

$$\min_{s,\mu}\{-\| \left(A_1 + A_2\boldsymbol{s} + A_3\boldsymbol{s}^2\right) - \mu \left(B_1 + B_2\boldsymbol{s} + B_3\boldsymbol{s}^2\right) \|_2^2, \tag{2.3}$$

where $\boldsymbol{s} = [s_1, s_2, \ldots, s_m] \in \mathbb{R}^m$, $\boldsymbol{s}^2 = [s_1^2, s_2^2, \ldots, s_m^2] \in \mathbb{R}^m$, A_i and B_i $(i = 1, 2, 3)$ are calculated from the two datasets $X^{(1)} \in \mathbb{R}^{n_1 \times m}$ and $X^{(2)} \in \mathbb{R}^{n_2 \times m}$. We describe the technical details of the proposed SMSC method in Appendix A. We solve (eq:min) using the simplex method [65], which is a direct search method that does not use numerical or analytical gradients. This method does not always converge to a local minimum solution; therefore, its initial values are established ten times randomly in the numerical experiments.

We summarize the procedures of the proposed SMSC method in Algorithm 1.

2.4 Determination of Parameter b

Suppose that $X^{(1)} \in \mathbb{R}^{n_1 \times m}$ and $X^{(2)} \in \mathbb{R}^{n_2 \times m}$ are the data in different batches and $w_{i,j}^{(k)}$ denotes the similarities between the data samples of batch $X^{(k)}$ $(k = 1, 2)$. The parameter b are set to $b = \sum_{i:t_i>0} d_i / \sum_{i:t_i<0} d_i$ in [63], where $t_i = 1$ indicates that data samples i belong to a cluster and $t_i = -1$ indicates that sample i belong to another cluster. The

Algorithm 1 Scaling method for batch correction based on spectral clustering (SMSC)

Input: N batches of data $X^{(i)} \in \mathbb{R}^{n_i \times m}$ $(i = 1, 2, \ldots, N)$, number of reduced dimensions ℓ, number of nearest neighbors k, hyperparameters σ and λ.

Output: Batch corrected and merged data $C \in \mathbb{R}^{(n_1+n_2+\cdots n_N)\times \ell}$

1: Define reference data $C \leftarrow X^{(1)}$.

2: **for** $i = 2 : N$ **do**

3: Compute $A_1, A_2, A_3, B_1, B_2, B_3$ in (2.3) between batch C and $X^{(i)}$.

4: Solve the minimization problem in (2.3) for adjusting value $\boldsymbol{s} = [s_1, s_2, \ldots, s_m]^{\mathsf{T}}$.

5: Generate the scaling matrix $S = \mathrm{diag}(s_1, s_2, \ldots, s_m)$.

6: Compute the corrected data $\widetilde{X}^{(i)} = X^{(i)}S$.

7: Calculate the new reference data $C \leftarrow \begin{bmatrix} C \\ \widetilde{X}^{(i)} \end{bmatrix}$

8: **end for**

choice of $t_i = 1$ or $t_i = -1$ is flexible. In this study, we determine the parameter b by

$$b = \max\left(\frac{\sum_{i=1}^{n_1} d_1^{(1)}}{\sum_{i=1}^{n_2} d_1^{(2)}}, \frac{\sum_{i=1}^{n_2} d_1^{(2)}}{\sum_{i=1}^{n_1} d_1^{(1)}}\right),$$

where $d_i^{(k)} = \sum_{j=1}^{n_k} w_{i,j}^{(k)}$ $(k = 1, 2)$. We can be more distinguished from the cluster with label 1 by using the larger value b.

2.5 Constraint of Parameter σ

We use the first approximation of the exponential function in reformulation (2.3). This approximation is valid under the constraint

$$0 < \frac{\|\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_j^{(k)}\|_2^2}{2\sigma^2} < 1.$$

Therefore, we have the following constraint:

$$\sigma^2 > \frac{\|\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_j^{(k)}\|_2^2}{2},$$

where $\boldsymbol{x}_i^{(k)}$ is the ith row of the data $X^{(k)}$ $(k = 1, 2, \ldots, N)$ and N is the number of batches.

2.6 Numerical Experiments

We evaluate the performance of the proposed SMSC method by comparing it with those of existing batch effect correction methods on both artificial and gene expression datasets. The compared methods are combat [52], limma [48], svaseq [49, 50], and MNN [47]. We also compare the SMSC method with the baseline method that does not correct the batch effects. For all methods, we apply SC on the merged data and test the performance in terms of clustering accuracy. The evaluation metrics for the clustering performance include the overall accuracy (OA) and normalized mutual information (NMI) [$\%$] [66]. The OA is defined as

$$\text{OA} = \sum_{i=1}^{K} \hat{n}_i / \sum_{i=1}^{K} n_i,$$

where n_i is the number of samples in class i, $\hat{n}_i$ is the number of samples clustered into class i, and K is the number of classes. The larger the values of OA and NMI, the better the clustering accuracy. In k-means clustering, we set the values of k as the number of classes. The k-means clustering are repeated 20 times with random initializations, and we show the mean performance with standard deviation. In SMSC, we set $\lambda = 10^{-3}$ and chose the value of $\alpha < \sigma < (\alpha + 100)$, where $\alpha = \|\boldsymbol{x}_i^{(k)} - \boldsymbol{x}_j^{(k)}\|_2/\sqrt{2} + 10^{-3}$ and $\boldsymbol{x}_i^{(k)}$ is the ith row of the data $X^{(k)},\ k = 1, 2, \ldots, N$, and N is the number of batches. We use the publicly available R code of the compared batch effect correction methods, i.e., combat, limma, svaseq, and MNN, followed by the execution of clustering in MATLAB 2019a. The proposed SMSC method are coded and executed in MATLAB 2019a.

We use a simple simulated dataset with two batches and ten dimensions. Table 2.1 gives the number of samples for each class of each batch. Figure 2.1 shows the samples for two dimensions and the remaining eight dimensions are initialized with uniformly

distributed random numbers in the interval [0, 1]. Different colors and different shapes denote different classes and different batches, respectively. In this case, the data samples in class 1 from different batches are separated. If we do not have the batch effects, the same classes (colors) from different batches (shapes) should be pictured adjacently.

For real-world dataset, we use four datasets from the Gene Expression Omnibus (GEO) [67], which contains gene expression data with more features than samples. Table 2.2 gives the number of samples for each class, number of features, and the ID of GEO for each dataset. Datasets (a)-(c) are gene expression data from a microarray and dataset (d) comprises read-count data for single-cell RNA-seq. We normalize the datasets (a)-(c) to the range [0, 1] and use trimmed mean of M values (TMM) method [68] for normalization to dataset (d). We select the features using the analysis of variance (ANOVA) (p value $\leq$0.05 after false discovery rate (FDR [69] correction) for correcting the batch effects. For all methods, the number of reduced dimensions is the same as that of classes. In SMSC, we define Batch 1 as the reference data.

Table 2.1: Number of samples for each class

	Class 1	Class 2	Class 3
Batch 1	300	100	300
Batch 2	30	200	0

2.6.1 Simulation results

Figure 2.2 shows the performance by varying the number of reduced dimensions. As the number of reduced dimensions increases, the accuracy of all methods decreased. Among them, the proposed method achieves higher accuracy than the compared method on the simulated dataset. The proposed method obtains the best result in terms of OA and NMI when the dimension is reduced to $\ell = 2$ and outperforms the compared methods. These results show that the proposed SMSC method is more robust than other methods.

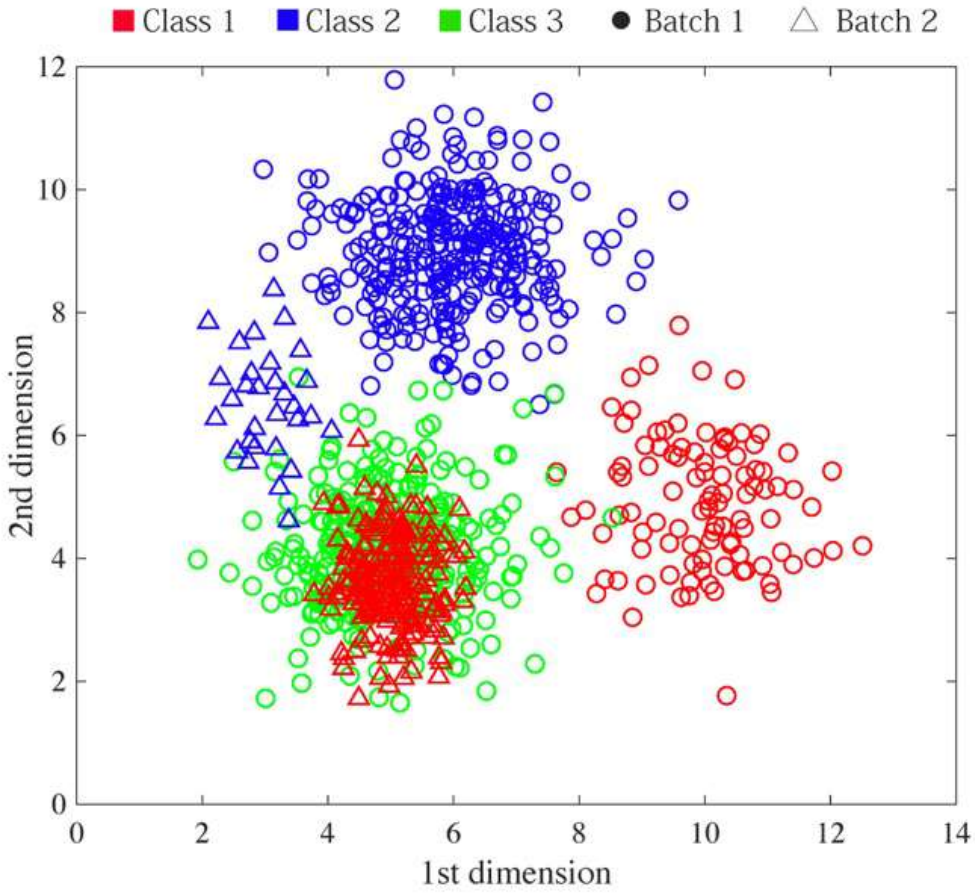


Figure 2.1: Two-dimensional visualization of the simulated data

Figure 2.3 shows the data samples reduced in a two-dimensional space and the true labels are denoted with different colors. The result for the svaseq method is similar to the uncorrected case which cannot remove the batch effects. For the combat, limma, and MNN methods, the data samples in class 2 from different batches are projected adjacent to each other. However, the data samples in class 1 from different batches are not projected adjacently. This means that the batch effects of Class 1 and Class 2 are not removed. For the proposed SMSC method, the data samples in each class from different batches are projected to resemble each other, i.e., the batch effects of all classes are removed. These results showed that SMSM performs better than the well-established methods to remove the batch effects such that the same classes (colors) from different batches (shapes) are

projected to resemble each other.

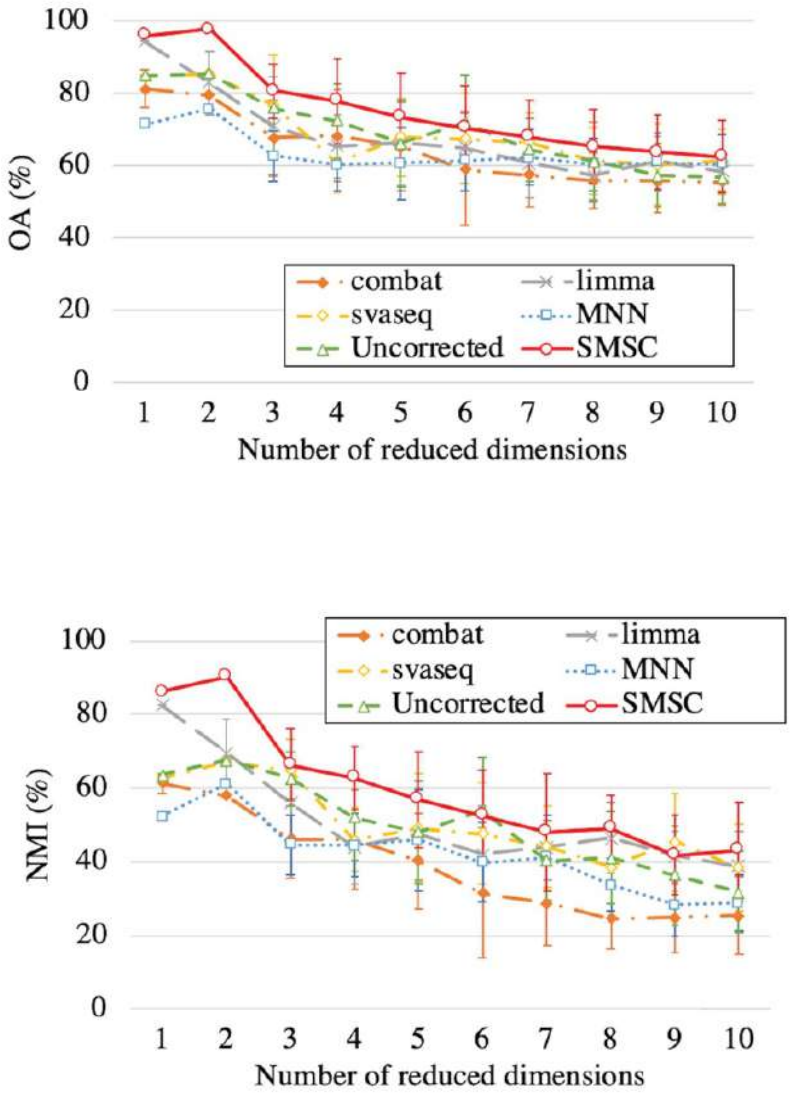


Figure 2.2: Mean and standard deviation of clustering performance vs. reduced dimensions.

2.6.2 Gene expression results

Table 2.3 gives the mean values with standard deviation in terms of OA and NMI for each method. The bold font denotes the best result for each dataset. The proposed SMSC

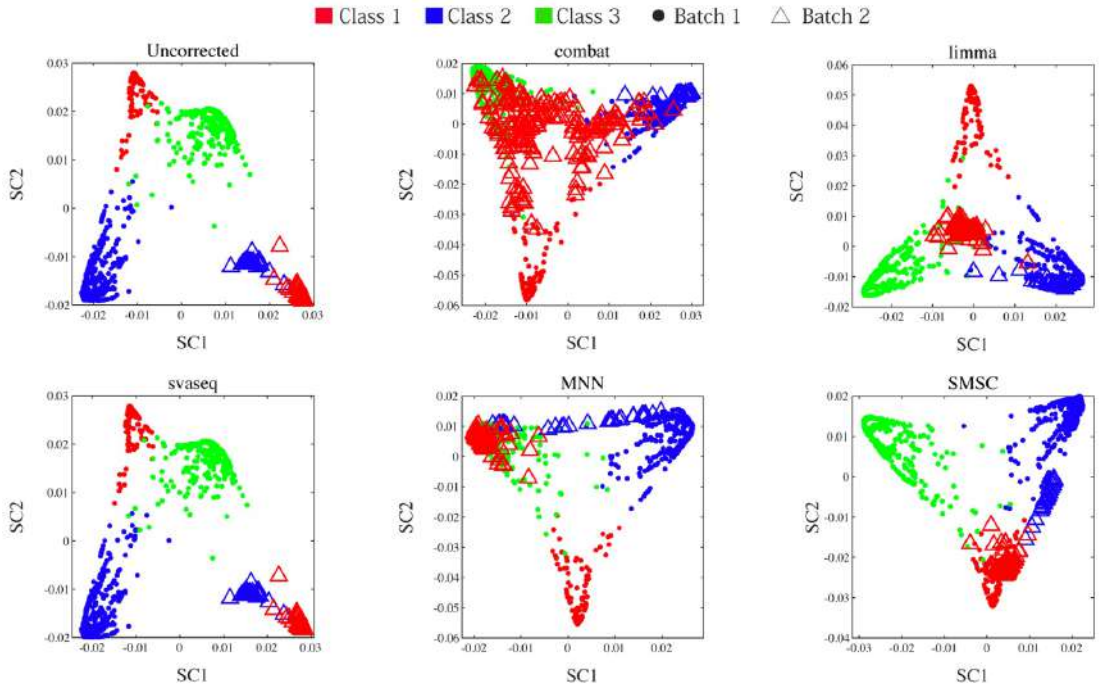


Figure 2.3: Data samples in reduced two-dimensional space.

method obtains the best result for all datasets. For Colorectal, Leukemia, and Breast cancer, the limma or svaseq is the second best method in terms of OA and NMI, the MNN is poor because it is developed for single-cell RNA-seq data. For the single cell RNA-seq data i.e., Kidney, the combat and limma methods do not perform well, while the MNN and svaseq method obtain better performance than the Uncorrected method in terms of OA and NMI. These results show that the proposed SMSC method performs well on both microarray and single-cell RNA-seq datasets.

2.7 Summary

In this chapter, we consider performing accurate cluster analysis to improve the explanation of cluster analysis. To accurately measure the biological variability and obtain precise statistical inference, we propose an effective batch effect correction method. The proposed method merges multiple data from different batches by scaling adjustment the

features in a low-dimensional space, which is different from the existing L/S model using the empirical Bayes method to find the constant values for normalization of each feature. We propose an approximation solution to solve the optimization problem for the scaling adjustment values. Furthermore, we propose an automatic tuning technique to reduce the number of hyperparameters that appeared in the proposed method. Numerical experiments show that the proposed method is effective when combined with spectral clustering. For accuracy, thereby making the proposed model more robust for interfering features. For the simulated dataset, the proposed method project data samples in the same classes from different bathes to resemble each other. For the gene expression datasets with more features than samples, the proposed method is more robust and outperforms the well-established methods on both microarray and single-cell RNA-seq datasets.

Table 2.2: The number of samples for each class of entire data

(a) Colorectal

	Class			
	1	2	Features	ID
Batch 1	11	11	54,675	GSE4107
Batch 2	0	36	54,675	GSE4526

(b) Leukemia

	Class				
	1	2	3	Features	ID
Batch 1	30	9	0	54,675	GSE2677
Batch 2	0	40	20	54,675	GSE6338

(c) Breast cancer

	Class			
	1	2	Features	ID
Batch 1	171	47	22,284	GSE4611
Batch 2	10	0	22,284	GSE3893
Batch 3	0	96	22,284	GSE2294

(d) Kidney

	Class										
	1	2	3	4	5	6	7	8	9	10	11
Batch 1	166	1	1459	216	578	123	84	29	9	49	65
Batch 2	78	1	590	143	200	15	112	4	2	17	5

	12	13	14	15	16	Features	ID
Batch 1	3	46	97	17	1	16,271	GSM2871078
Batch 2	2	12	150	52	0	16,271	GSM2871078

Table 2.3: Performance rate (mean% ± std) for real-world datasets

(a) Colorectal

	ACC	NMI
Uncorrected	66.67±15.38	44.78±28.05
Combat	78.10±16.69	38.64±34.35
Limma	82.76± 0.00	46.90± 0.00
svaseq	66.90± 8.61	19.60±11.86
MNN	62.07± 9.17	13.95± 5.75
SMSC	**85.35±17.30**	**61.52±40.07**

(b) Leukemia

	ACC	NMI
Uncorrected	66.41±6.48	50.29± 8.80
Combat	65.66±0.00	47.90± 6.45
Limma	64.80±0.36	48.24±4 .29
svaseq	66.31±8.69	53.22±13.58
MNN	54.19±4.30	24.57± 4.13
SMSC	**71.16±8.01**	**60.69±11.40**

(c) Brest cancer

	ACC	NMI
Uncorrected	62.53± 3.25	21.01± 6.63
Combat	60.56± 1.93	11.15± 6.14
Limma	52.55± 1.68	17.39± 2.34
svaseq	64.51±14.36	17.19±13.73
MNN	56.17± 0.00	2.11± 8.48
SMSC	**85.49±0.00**	**50.82±0.00**

(d) Kidney

	ACC	NMI
Uncorrected	52.46±5.51	27.75±9.78
Combat	48.44±6.58	27.74±6.36
Limma	50.24±3.41	21.45±3.65
svaseq	55.66±2.32	32.47±2.56
MNN	54.60±7.37	38.48±3.85
SMSC	**65.86±3.12**	**43.51±4.70**

Chapter 3

Feature selection based on principal component analysis of sample space

In this chapter, we consider selecting the significant features for cluster analysis to improve its explanation and present a novel feature selection method based on the principal components of the sample space. We focus on distinguishing between the features required for cluster analysis and the background noise. The proposed method performs principal component analysis (PCA) on the sample space of the data and selects the features that contribute to cluster separation.

This chapter is organized as follows. In Section 3.1, we introduce the PCA method. In Section 3.2, we present an overview of the proposed method. In Section 3.3, we discuss three methods for detecting distortions in the principal components. Performance analysis and comparison are presented in Section 3.4. Section 3.6 concludes this chapter.

3.1 Principal Component Analysis

PCA is a linear dimensionality reduction technique that aims at preserving the global structure. It attempt to obtain the best approximation of the data samples and find the low-dimensional space in which data samples variance becomes maximum after projection.

PCA transforms the sample $\boldsymbol{x}_i \in \mathbb{R}^m$ to an embedded sample $\boldsymbol{z}_i \in \mathbb{R}^\ell$ $(1 \leq \ell < m)$ of low-dimensional space with matrix $T \in \mathbb{R}^{m\times\ell}$. The transformation matrix T is defined as

$$T = \arg\max_{T} \left[\mathrm{tr} \left(T^{\mathsf{T}} \widetilde{C} T \left(T^{\mathsf{T}} T \right)^{-1} \right) \right], \tag{3.1}$$

where $\widetilde{C} = \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^{\mathsf{T}} \in \mathbb{R}^{m\times m}$, $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \in \mathbb{R}^m$. Let $\{\boldsymbol{\varphi}_i\}_{i=1}^{n}$ be the eigenvectors associated with the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ of the eigenvalue problem

$$S\boldsymbol{\varphi}_i = \lambda_i \boldsymbol{\varphi}_i.$$

Then, the solution of (3.1) is given by

$$T = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_\ell],$$

and the embedded data $\boldsymbol{z}_i \in \mathbb{R}^\ell$ are given by

$$\boldsymbol{z}_i = T^{\mathsf{T}} \boldsymbol{x}_i.$$

We can visualize the sample distributions in the low-dimensional space by performing PCA for the feature space. In the low-dimensional space of PCA, the expression patterns of the sample are similar if the distance between the samples is small; however, the expression patterns are different if they are far apart.

3.2 The Framework of Proposed Method

The objective of the proposed method is to remove the background noise from large subset features. The proposed method only selects significant features for cluster analysis using the principal component of the sample space, while maintaining the accuracy of clustering analysis for the samples.

Let $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{n\times m}$, $\boldsymbol{x}_i \in \mathbb{R}^n$ be a dataset with n samples and m features. First, we solve the following equation:

$$C\boldsymbol{t}_i = \lambda_i \boldsymbol{t}_i, \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n, \tag{3.2}$$

where $C = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^{\mathsf{T}} \in \mathbb{R}^{n\times n}$, $\boldsymbol{\mu} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i \in \mathbb{R}^m$. Then, we compute m principal components, $\boldsymbol{p}_i \in \mathbb{R}^m$ $(i = 1, 2, \ldots, m)$, by

$$\boldsymbol{p}_i = X^{\mathsf{T}}\boldsymbol{t}_i. \tag{3.3}$$

Next, we test the normality of the principal components using statistical tests, which are discussed in Section 3.3. We assume that the ℓ-principal component space with normality is denoted by $P = \{p_{i,j}\} \in \mathbb{R}^{m\times\ell}$. Then, we compute the distance, d_i $(i = 1, 2, \ldots, m)$, from the origin of each gene on space P such that

$$d_i = p_{i,1}^2 + p_{i,2}^2 + \cdots + p_{i,\ell}^2. \tag{3.4}$$

Finally, we select s genes in descending order of distance d_i, where s is the parameter.

The procedures of the proposed method are summarized in Algorithm 2.

Algorithm 2 Procedures of the proposed method

Input: Dataset $X \in \mathbb{R}^{n\times m}$, parameters ℓ and s.

Output: Selected s genes

1: Perform z-score normalization across samples for each feature.
2: Solve (3.2) and compute (3.3) to obtain the principal components $\boldsymbol{p}_i \in \mathbb{R}^m$ $(i = 1, 2, \ldots, m)$.
3: Let $P = \{p_{i,j}\} \in \mathbb{R}^{m\times\ell}$ be the ℓ-dimensional space with distortion-free components.
4: Compute the distance (3.4) for each feature.
5: Select s features in descending order of the values of distance d.

3.3 Normality Test for Principal Components

To select features in a low-dimensional space composed of distortion-free principal components, we consider detecting the distortions in the principal components by the following three methods. Here, distortion refers to a distribution that deviates from the normal distribution.

Chi-Square goodness of fit test

The Chi-Square goodness of fit test test examines the null hypothesis that a data sample x_i $(i = 1, 2, \ldots, n)$ follows a normal distribution with means and variances estimated from the data. The alternative hypothesis assumes that the data sample does not follow such a distribution. In this test, the data are grouped into bins; subsequently, the observed and expected counts of these bins are calculated, followed by the calculation of the Chi-Square test statistics using the following equation.

$$\chi^2 = \sum_{i=1}^{nb} \frac{(O_i - E_i)^2}{E_i},$$

where O_i is the number of observed counts and E_i is the number of expected counts based on the distribution of the hypothesis. If the counts are large enough, the test statistic will have an approximate Chi-Square distribution. It compares the value of the test statistic with a Chi-Square distribution, with equal degrees of freedom equal to $nb-1-np$, where nb is the number of bins and np is the number of estimated parameters used to determine the expected count (in this case, $np = 2$).

Skewness test

Skewness is an indicator that demonstrates the deviation of the distribution of a data sample, x_i $(i = 1, 2, \ldots, n)$, from the normal distribution and exhibits left-right symmetry; it is calculated by the following equation.

$$skewness = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^{\frac{3}{2}}},$$

where $\bar{x}$ is the mean of x_i. If the skewness is negative, the distribution leans toward the right, and if it is positive, the distribution leans toward the left. The skewness is 0 when the distribution is left-right symmetric. Figure 3.1 illustrates an example of skewness. In general, if the skewness is between -0.5 and 0.5, the distribution is considered to be approximately symmetric.

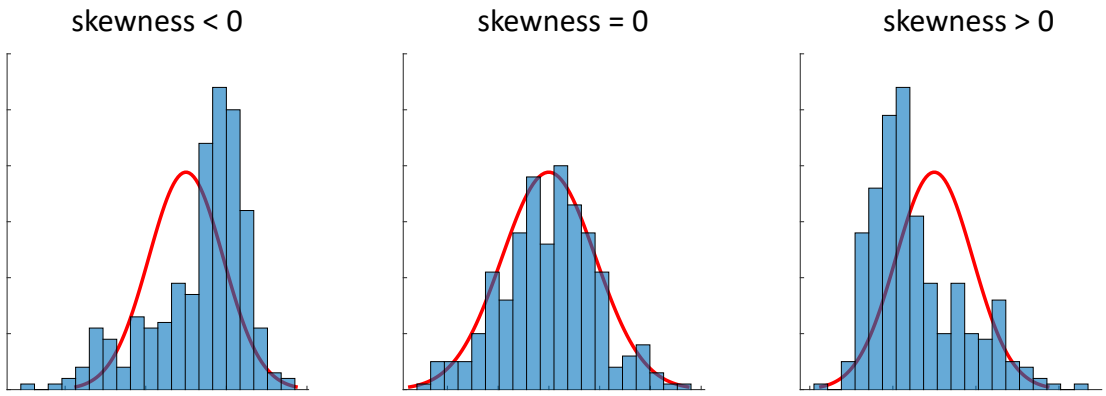


Figure 3.1: Example of three types of skewness

Kurtosis test

Kurtosis is a measure of the prominent tendency of the distribution of a data sample x_i $(i = 1, 2, \ldots, n)$; it is calculated using the following equation.

$$kurtosis = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2},$$

where $\bar{x}$ is the mean of x_i. The kurtosis of the normal distribution is 3. The kurtosis value of data samples with several outliers is greater than 3; moreover, the kurtosis and tail of the distribution are steep and long, respectively. The kurtosis value of data samples with few outliers is less than 3; moreover, the kurtosis and tail of the distribution are gentle and short, respectively. Figure 3.2 illustrates an example for both cases.

3.4 Numerical Experiments

We verify that the proposed method maintains the clustering accuracy by comparing it with that of the existing feature selection methods. The compared methods include Seurat and Brennecke. We project the samples to a low-dimensional space by PCA and perform k-means clustering in that space. In k-means clustering, the values of k denote the number of classes; it is repeated 20 times with random initializations, and we show the mean

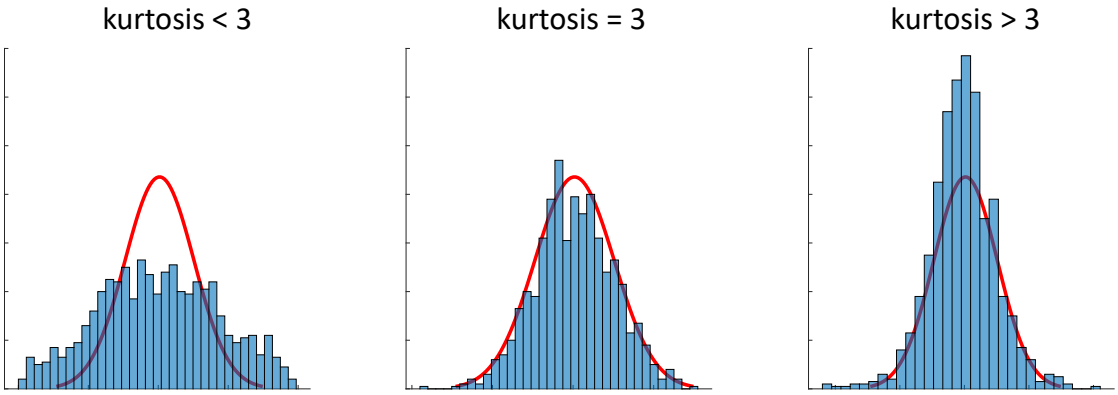


Figure 3.2: Example of three types of kurtosis

performance. The evaluation metrics for the clustering performance include the normalized mutual information (NMI) [%] [66] and the rand index (RI) [70]. The larger the values of NMI and RI, the better the clustering accuracy. For visualization, we perform uniform manifold approximation and projection (UMAP) [28] for the low-dimensional PCA space and demonstrate the two-dimensional UMAP space. The proposed method was coded and executed in MATLAB 2019a.

3.4.1 Datasets and their processing

We generate a simulation dataset from the uniformly distributed random numbers in the interval [0, 1]. It contains 1000 samples and 4500 features. Figure 3.3 shows the heat map of the simulation dataset, where each row and column represent a sample and feature, respectively. The 1000 samples are assigned to five classes; the different colors of the right bar denote different classes. The top color bar denotes different types of features. The first 500 features are effective for clustering five classes, and the remaining 4000 features are generated as the background noise. Thus, the first 500 features are required for clustering.

We downloaded the Gierahn dataset [71] from Gene Expression Omnibus (GEO)[67] under accession number GSE92495, where the tuberculosis-exposed human peripheral

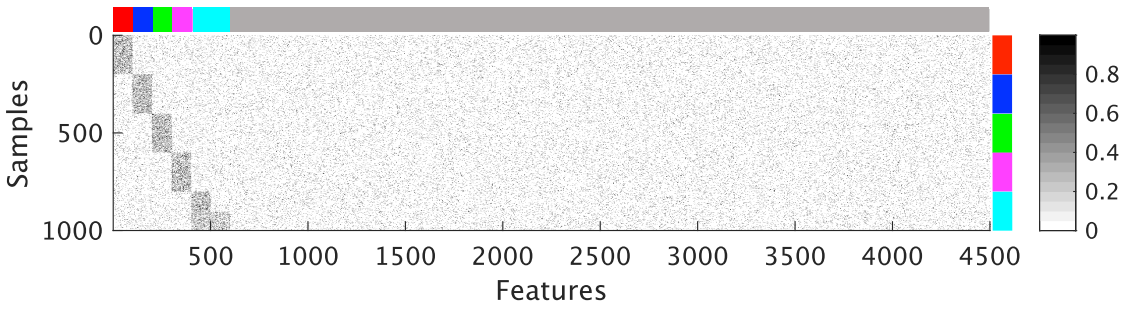


Figure 3.3: Heat map of the simulation dataset.

blood mononuclear (PBMC) cell sample was used. It contains 4,296 cells, 6,713 genes, and 6 classes.

We also downloaded the Pollen dataset [72] from GSE71315, where the human neocortex cell sample is used. It includes 50 single cell libraries from gestational weeks (GW)16, GW21 cells previously analyzed [73] and primary cells derived from GW21 brain that were cultured in differentiation media for 3 days. It contains 276 cells, 13,007 genes, and expression values were in size factor normalized counts, according to DESeq. They classified the cells into 7 clusters by hierarchical clustering.

The details of the datasets are listed in Table 3.1.

Table 3.1: Details of datasets

	# of samples	# of features	# of classes	Accession number
1. Simulation dataset	1000	4500	5	–
2. Gierahn dataset	4,296	6,713	6	GEO: GSM2486333
3. Pollen dataset	276	13,007	7	GEO: GSE71315

3.4.2 Simulation dataset

Figure 3.4 shows the distributions of obtained the first to twelfth principal components. We used the seventh to eleventh principal components, which were observed to be free from distortions based on the Chi-Square goodness of fit test; then, we computed the distance of the distortion-free space from the origin for each feature.

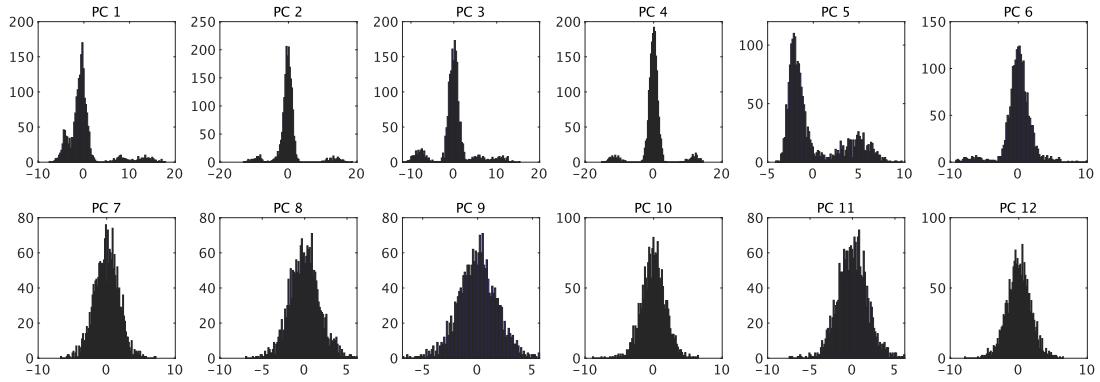


Figure 3.4: Distributions of the obtained 1st to 12th principal components.

Figure 3.5 shows a sample projection into a two-dimensional space. Each point and shape denote each sample and the correct class of the samples, respectively. For Brennecke, the different shapes are almost mixed, i.e., it does not perform clustering well because the selected 200 features include significant amount of background noises. For Seurat, the selected 200 features include those features that contribute to cluster separation, but the background noise is also selected; therefore the different shapes are slightly mixed. For the proposed method, the samples are completely clustered because it selects the necessary features for clustering and removes the background noise. These results show that the proposed method can remove the unnecessary features for clustering using PCA.

Figure 3.6 shows the clustering accuracy by varying the number of selected features, s. The number of reduced dimensions of PCA was set to be 4. For Seurat and Brennecke, the accuracy performance was poor when the number of selected features was small. This

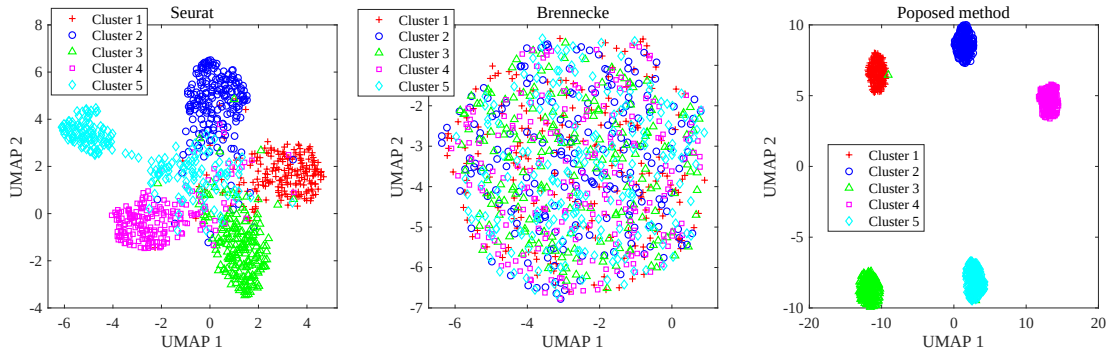


Figure 3.5: The two-dimensional space after reducing to seven-dimensions using the selected 200 features by each method.

is because a significant amount of background noise was present in the top rankings. The proposed method performed well even when the number of selected features was small; additionally, it could select the significant features for clustering from the top rankings. These results show that the proposed method can remove the unnecessary genes for clustering without loss of accuracy.

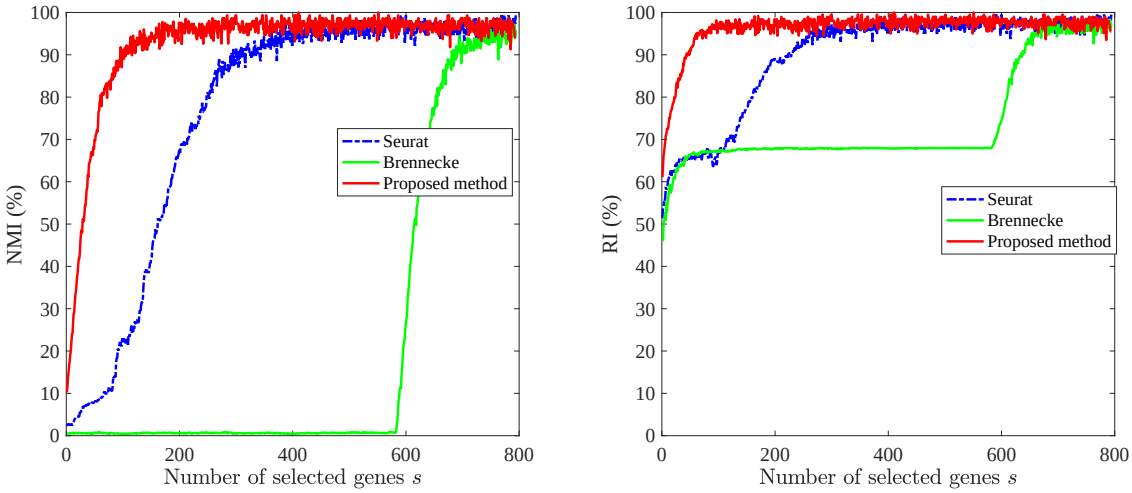


Figure 3.6: The clustering performance vs. selected genes.

3.4.3 Gierahn dataset

First, we computed the principal components after selecting 2000 genes using Seurat. Then, we excluded the first principal component because the distributions after the second principal component are close to the normal distribution. Subsequently, parameters ℓ and s of the proposed method were set to be six and 200, respectively. The number of selected features for Seurat and Brennecke was 200 each.

Figure 3.7 shows the two-dimensional UMAP space. Each point and the different shapes denote each sample and the correct classes of samples, respectively. For Seurat and Brenncke, it does not perform clustering well because the selected 200 features do not include the significant gene for clustering. For the proposed method, it can obtain a good clustering structure even if using only 200 features are used. These results show that the proposed method can select the significant features for clustering and remove the background noises.

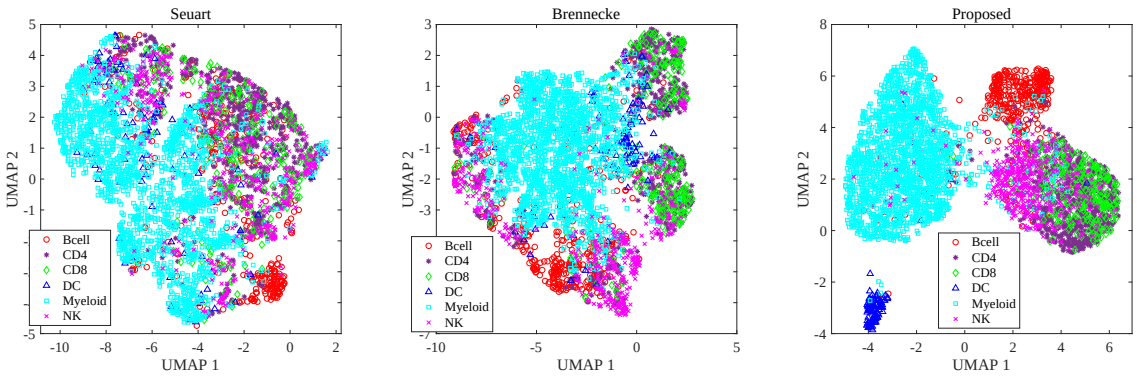


Figure 3.7: The two-dimensional UMAP space after reducing to 10 dimensions by PCA.

Figure 3.8 shows the clustering accuracy by varying the number of selected genes s. The number of reduced dimensions of PCA was set to be 10. The Seurat and Brennecke methods cannot perform clustering well when the number of selected genes is small. For the proposed method, the clustering accuracy for a small number of genes was observed to be better than that when all 2000 genes were used. These results show that the proposed

method can remove the background noise while maintaining the clustering accuracy.

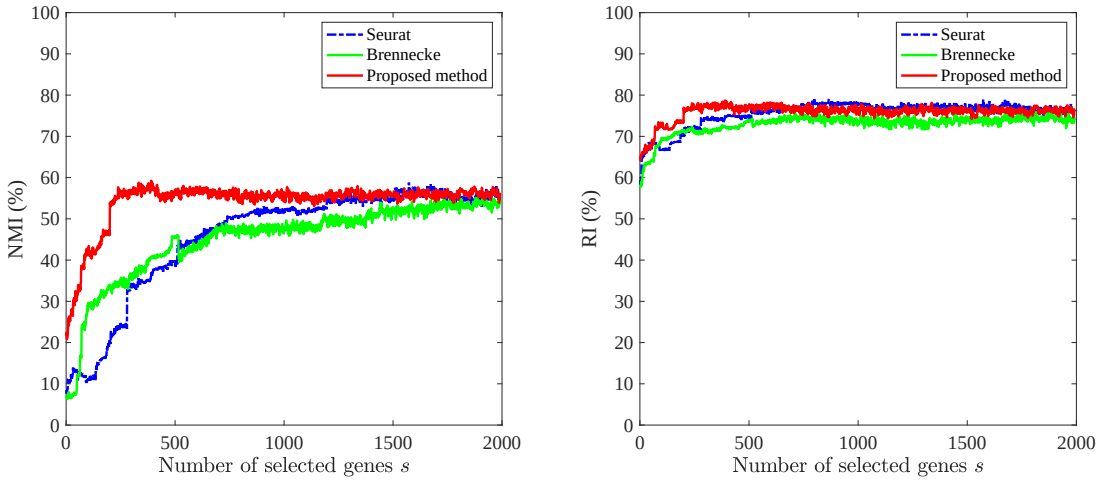


Figure 3.8: Clustering performance vs. selected genes.

3.4.4 Pollen dataset

We computed the principal components after selecting 1000 genes by Seurat. Then, we computes the distance from the origin in the distortion-free space. Subsequently, parameters ℓ and s of the proposed method were set three and 100, respectively. The number of selected features for Seurat and Brennecke was 100 each.

Figure 3.9 shows the two-dimensional UMAP space. Each point and the different shapes denote each sample and the correct classes of samples, respectively. For Seurat, it can obtain the better clustering structure than Brennecke results when the selected genes is 100. For the proposed method, the Interneuron class (blue), the Dividing R. G class (red), and the Radial Glia class (black) are completely clustered. These results show that the proposed method can improve the visualization of the clustering structure.

Figure 3.10 shows the clustering accuracy by varying the number of selected genes s. We set the number of reduced dimensions of PCA is six. For the Seurat, the clustering accuracy is not well when the number of selected genes is small. The clustering accuracy of the proposed method is best when selecting 175 genes and better than the Seurat in all

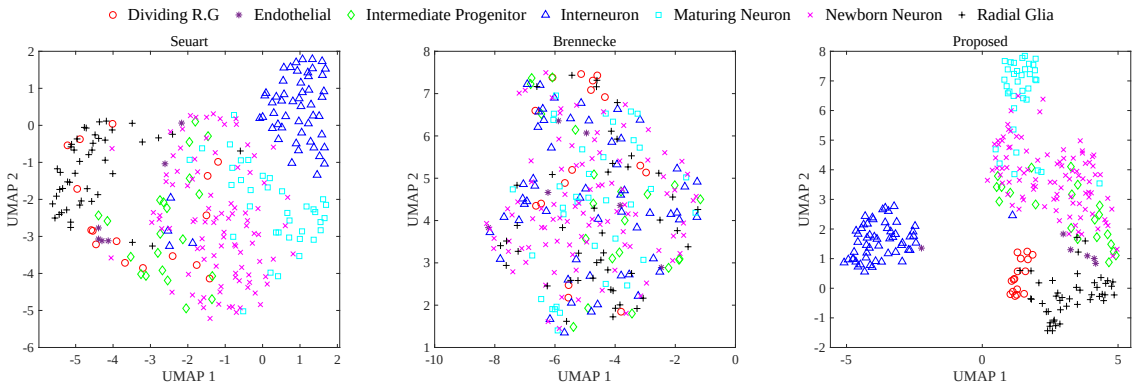


Figure 3.9: The two-dimensional UMAP space after reducing to six dimensions by PCA.

situations. These results show that the proposed method ranks genes appropriately which means can remove the background noise and select necessary genes for clustering.

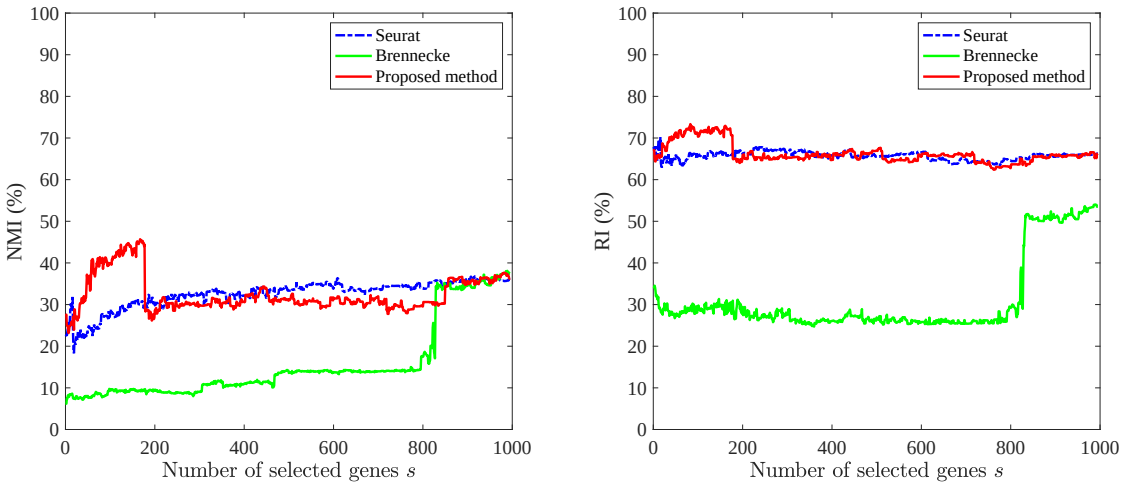


Figure 3.10: Clustering performance vs. selected genes.

We show the clustering performance by varying the number of reduced dimensions of PCA for Gierahn and Pollen datasets in Appendix B.

3.5 Related experiments

We compute the following statistics for each feature to identify the features that work functionally in each cluster and visualized by the Feature Plot. The FeaturePlot is a diagram in which the sample is colored according to the expression level of each feature, and it is possible to visually capture the features that are functioning specifically in the cluster.

Statistical values	
p_val	The calculated values from statistics by Wilcoxon rank-sum test
avg_logFC	log fold-chage of the average expression between the two groups
pct_1	The percentage of cells where the gene is detected in the first group
pct_2	The percentage of cells where the gene is detected in the second group
p_val_adj	Adjusted p-value, based on bonferroni correction using all genes in the dataset

3.5.1 Gierahn dataset

We calculated the statistics for each of the 200 selected features for each cluster. Table 3.2 gives the top 4 features of each cluster based on the ascending order of *avg_logFC*.

Figure 3.11 shows the FeaturePlot of the features 'IGKC', 'IL7R', 'IL7R', 'TXN', 'IL1B', and 'IFITM1', which have the largest *avg_logFC* values for each cluster. The largest features of *avg_logFC* for each cluster are specifically working in a specific cluster. The features in Table 3.2 are mostly listed in the supplementary table 4 of genes enriched within each cluster by Gierahn, et al [71].

3.5.2 Pollen dataset

We calculated the statistics for each of the 100 selected features for each cluster. Table 3.3 gives the top 3 features of each cluster based on the ascending order of *avg_logFC*.

Figure 3.12 shows the FeaturePlot of the features 'MKI67', 'MT-RNR1', 'CCND2', 'DLX6-AS1', 'STAB2', 'SEMA3C', and 'CRYAB', which have the largest *avg_logFC* values for each cluster. The largest features of *avg_logFC* for each cluster are specifically working in a specific cluster. The features in Table 3.3 are mostly listed in the Additional file 14: Table S7 of genes enriched within each cluster by Pollen, et al [72].

3.6 Summary

In this chapter, we considered selecting the significant features for cluster analysis to improve its explanation. To remove the unnecessary background noise, we proposed an effective feature selection method. The proposed method performed PCA for the sample space of the data; then, it distinguished between the features required for cluster analysis and the background noise. The proposed method selected the distance of each feature from the origin in descending order in a low-dimensional space composed of distortion-free principal components. We presented three methods for determining the distortions in the obtained principal component. Numerical experiments demonstrated that the proposed method is effective for removing the background noise and improving the clustering accuracy. For the simple simulation dataset, we showed that the Seurat and Brennecke methods select a significant amount of background noise owing to its presence in the top rankings. For all used datasets, the proposed method improved the clustering accuracy and efficiently visualized the clustering structure even when the number of selected features was samll.

Table 3.2: Statistic values for each cluster

gene	*p_val*	*avg_logFC*	*pct_1*	*pct_2*	*p_val_adj*	Cluster
'IGKC'	8.9E-172	2.235	0.410	0.030	4.5E-170	Bcell
'MS4A1'	0.0E+00	2.157	0.723	0.044	0.0E+00	Bcell
'BANK1'	3.1E-168	1.280	0.380	0.024	1.0E-166	Bcell
'TCF4'	6.1E-87	1.091	0.364	0.061	1.5E-85	Bcell
'IL7R'	1.4E-130	1.204	0.765	0.311	7.1E-129	CD4
'CD3D'	3.2E-110	1.016	0.577	0.182	8.2E-109	CD4
'TRAC'	1.2E-90	1.003	0.481	0.146	1.2E-89	CD4
'CAMK4'	6.2E-103	0.974	0.347	0.060	1.1E-101	CD4
'IL7R'	1.4E-96	1.540	0.851	0.353	4.7E-95	CD8
'CD2'	2.6E-97	1.492	0.755	0.225	1.8E-95	CD8
'CD3D'	9.9E-56	1.128	0.602	0.222	2.2E-54	CD8
'TRAC'	2.6E-50	0.971	0.535	0.178	4.3E-49	CD8
'TXN'	2.1E-72	3.102	1.000	0.492	2.3E-71	DC
'IDO1'	1.1E-135	2.658	0.971	0.150	2.9E-134	DC
'TBC1D4'	1.0E-260	2.443	0.817	0.036	8.0E-259	DC
'DUSP5'	1.1E-140	1.724	0.663	0.050	4.2E-139	DC
'IL1B'	7.4E-35	1.583	0.299	0.137	1.6E-34	Myeloid
'CYP1B1'	1.0E-107	1.278	0.380	0.081	2.3E-106	Myeloid
'APOBEC3A'	5.4E-89	1.172	0.328	0.070	4.0E-88	Myeloid
'KYNU'	1.7E-181	1.094	0.616	0.157	1.7E-179	Myeloid
'CXCL1'	5.2E-20	1.031	0.176	0.078	8.8E-20	Myeloid
'IFITM1'	1.5E-117	1.354	0.652	0.210	1.1E-115	NK
'IFITM2'	1.9E-58	0.972	0.624	0.333	7.4E-57	NK
'IL32'	2.9E-29	0.822	0.442	0.228	7.6E-28	NK
'HSH2D'	1.0E-11	0.450	0.138	0.057	2.0E-10	NK

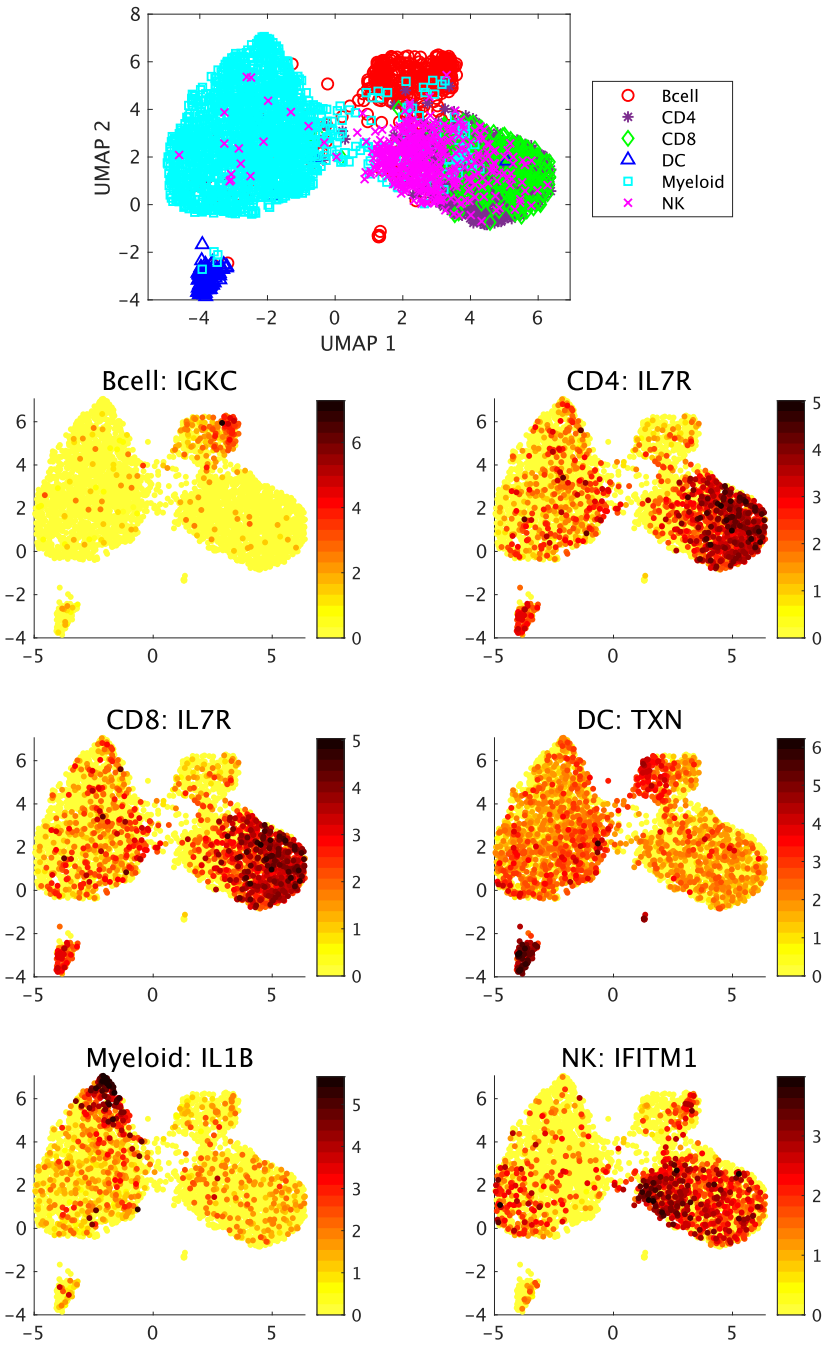


Figure 3.11: FeaturePlot with the largest *avg_logFC* features in each cluster.

Table 3.3: Statistic values for each cluster

gene	*p_val*	*avg_logFC*	*pct_1*	*pct_2*	*p_val_adj*	Cluster
'MKI67'	8.9E-25	3.825	1.000	0.108	2.2E-23	Dividing R.G
'TPX2'	4.5E-24	3.641	1.000	0.108	7.5E-23	Dividing R.G
'TOP2A'	6.1E-16	3.636	1.000	0.250	4.3E-15	Dividing R.G
'MT-RNR1'	2.1E-04	2.348	1.000	0.956	2.3E-03	Endothelia
'ATP1A2'	3.3E-01	1.965	0.333	0.207	4.3E-01	Endothelia
'RPS6'	1.1E-03	1.583	1.000	0.970	9.4E-03	Endothelia
'CCND2'	2.5E-04	0.942	1.000	0.937	8.4E-03	Intermediate Progenitor
'PER2'	8.7E-02	0.710	0.667	0.624	2.2E-01	Intermediate Progenitor
'HNRNPA1'	4.2E-04	0.616	1.000	0.992	8.4E-03	Intermediate Progenitor
'DLX6-AS1'	1.6E-36	6.619	1.000	0.299	1.8E-35	Interneuron
'GAD1'	3.8E-31	5.617	0.673	0.041	2.2E-30	Interneuron
'ERBB4'	2.5E-43	4.338	0.982	0.118	5.7E-42	Interneuron
'SATB2'	8.6E-20	2.029	0.972	0.329	1.7E-18	Maturing Neuron
'MEF2C'	4.2E-11	1.898	0.806	0.333	2.1E-10	Maturing Neuron
'LIMCH1'	3.9E-09	1.875	0.694	0.250	1.6E-08	Maturing Neuron
'SEMA3C'	1.3E-15	1.819	0.773	0.330	1.7E-14	Newborn Neuron
'MLLT3'	4.1E-17	1.195	0.979	0.782	1.1E-15	Newborn Neuron
'PPP2R2B'	1.3E-14	0.976	0.856	0.397	1.1E-13	Newborn Neuron
'CRYAB'	9.3E-07	6.069	0.378	0.121	2.7E-06	Radial Glia
'CLU'	5.7E-30	4.123	0.956	0.242	4.0E-28	Radial Glia
'FAM107A'	4.2E-28	3.831	0.733	0.074	1.1E-26	Radial Glia

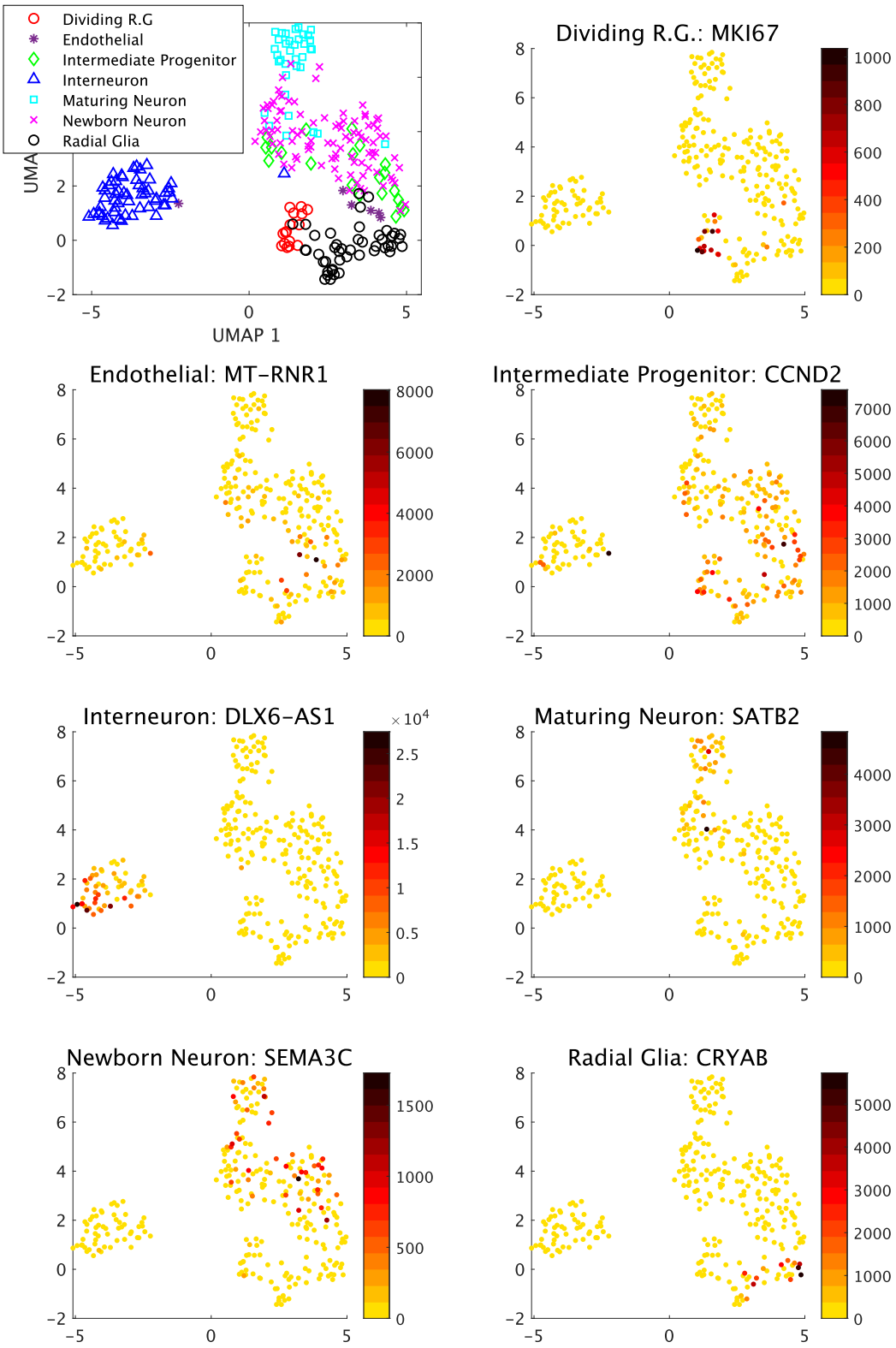


Figure 3.12: FeaturePlot with the largest *avg_logFC* features in each cluster.

Chapter 4

Summary

In cluster analysis for transcriptome data, the discover the differences between cell populations and their functional characteristics can be discovered by explaining and interpreting the characteristics of each cluster after analysis; accordingly, the explanations of the cluster analysis results are strongly required. The high-dimensional transcriptome data contain technical and biological noise, which further complicate the cluster analysis process. This study attempted to improve the explanation of cluster analysis by considering two issues; i.e., (1) performing accurate cluster analysis and (2) selecting features that contribute to cluster separation. Accordingly, we proposed two novel algorithms, including the scaling method for batch effect correction based on SC and the feature selection method base on PCA of the sample space.

In Chapter 2, we considered the first issue of cluster analysis for transcriptome data, i.e., performing accurate cluster analysis. We focused on removing the batch effect on the manifold space after dimensionality reduction using the SC method by performing scaling adjustments on each feature. We proposed an effective batch effect correction method. The proposed method merged multiple data instances from different batches by performing scaling adjustments on the features in a low-dimensional space, which is different from the existing L/S model that implements the empirical Bayes method to find the constant values for normalizing of each feature. Furthermore, we proposed an approximation

solution to solve the optimization problem for the scaling adjustment values and an automatic tuning technique to reduce the number of hyperparameters that appeared in the proposed method. In Comparison to the well-established methods, the proposed method was found to be effective when combined with SC; additionally, it was more robust and exhibited excellent performance on both microarray and single-cell RNA-seq datasets.

In Chapter 3, we considered the second issue of cluster analysis for transcriptome data, i.e., selecting the features that contribute to cluster separation. We focused on removing the background noise that is unnecessary for clustering by performing PCA in the sample space to distinguish between the features required cluster separation and the background noises. Accordingly, we proposed an effective feature selection method. The proposed method selected the distance of each feature from the origin in descending order in a low-dimensional space composed of distortion-free principal components. We adopted three methods to determine the distortions in the obtained principal component, i.e., the evaluation of the Chi-Square goodness of fit test, skewness, and kurtosis. Numerical experiments demonstrated that the conventional method selects a significant amount of background noise owing to its presence in the top rankings. Furthermore, we showed that the proposed method can remove the background noise while maintaining the accuracy of clustering analysis for samples; additionally, it can improve the clustering accuracy as well as the visualization of the clustering structure even when the number of selected features is small.

In future work, we will consider more accurate clustering and visualization methods. In addition, we will consider the development of an analysis tool for transcriptome data including the two proposed methods.

Acknowledgements

I would like to thank all people who have helped and inspired me during my research.

I especially would like to express my deep and sincere gratitude to my supervisor, Professor Tetsuya Sakurai. His wide knowledge and logical way of thinking have led to the completion of this dissertation. It is my great honor to thank Professor Keisuke Kameyama, Professor Jun Sakuma, Associate Professor Mamiko Sakata-Yanagimoto, and Assistant Professor Xiucai Ye for being my dissertation committee members and providing valuable advice and comments on evaluating this dissertation in its final form. This dissertation would not have been possible without their support. Let me thank them for all advice and help they have given me.

I would like to express my pleasure to Professor Takeshi Kitagawa. He has provided a comfortable environment for my research. It is my pleasure to thank Associate Professor Akira Imakura, Assistant Professor Claus Aranha, Assistant Professor Yasunori Futamura, Assistant Professor Keiichi Morikuni, Assistant Professor Xiucai Ye, Assistant Professor Ranjith Kumar Bakku, and Assistant Professor Keita Tokuda for their great efforts. Also, I would like to thank other members of the Mathematical Modeling & Algorithms Laboratory for their friendship.

I owe my deepest gratitude to Associate Professor Haruka Ozaki, Assistant Professor Akihiro Kuno, Ms. Sayaka Suzuki for their valuable help. They have carefully provided much useful advice for my research or even in my life.

I am deeply grateful to all my family, especially my parents, Kenji Matsuda and Midori Matsuda, and my husband, Mahiro Shirotori. Thanks to their support, I could con-

centrate on my research. I am also grateful to all my friends.

Appendix A: Technical details

We show an effective method to approximate the solution of the following equation:

$$\max_{s,\lambda_s} \|L_s \boldsymbol{v} - \lambda_s D_s \boldsymbol{v}\|_2^2 \tag{4.1}$$

We assume that we have two datasets $X^{(1)} = \{x_{i,j}^{(1)}\} \in \mathbb{R}^{n_1 \times m}$ and $X^{(2)} = \{x_{i,j}^{(2)}\} \in \mathbb{R}^{n_2 \times m}$, and the Fiedler vector $\boldsymbol{v} = [\boldsymbol{v}^{(1)\mathsf{T}}, \boldsymbol{v}^{(2)\mathsf{T}}]^{\mathsf{T}} \in \mathbb{R}^{(n_1+n_2)}$ of (4.1) is known by the batch information, where the values of $\boldsymbol{v}^{(1)} \in \mathbb{R}^{n_1}$ are one and those of $\boldsymbol{v}^{(2)} \in \mathbb{R}^{n_2}$ are $-b$. The similarity matrix based on the scaling adjustment $W_s \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ and $D_s \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ can be blocked such as

$$W_s = \begin{bmatrix} W_s^{(1,1)} & W_s^{(1,2)} \\ W_s^{(2,1)} & W_s^{(2,2)} \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}, \quad W_s^{(p,q)} = \{w^{(p,q)}\} \ (p,q=1,2),$$

$$D_s = \begin{bmatrix} D_s^{(1,1)} & O_{n_1,n_2} \\ O_{n_1,n_2}{}^{\mathsf{T}} & D_s^{(2,2)} \end{bmatrix} = \mathrm{diag}\left(d_1^{(1,1)}, d_2^{(1,1)}, \ldots, d_{n_1}^{(1,1)}, d_1^{(2,2)}, d_2^{(2,2)}, \ldots, d_{n_2}^{(2,2)}\right),$$

$$d_i^{(1,1)} = \sum_{j=1}^{n_1} w_{i,j}^{(1,1)} + \sum_{j=1}^{n_2} w_{i,j}^{(1,2)}, \quad d_i^{(2,2)} = \sum_{j=1}^{n_1} w_{i,j}^{(2,1)} + \sum_{j=1}^{n_2} w_{i,j}^{(2,2)},$$

where $W_s^{(p,q)}$ and $D_s^{(p,q)}$ $(p,q=1,2)$ are calculated from the entries of $X^{(1)}$ and $X^{(2)}$, and O_{n_1,n_2} is a $n_1 \times n_2$ matrix with zero in all entries. Then, (4.1) can be written as

$$\begin{aligned} & \max_{s,\lambda_s} \|L_s \boldsymbol{v} - \lambda_s D_s \boldsymbol{v}\|_2^2 \\ \Leftrightarrow \quad & \max_{s,\lambda_s} \Big(\left\| W_s^{(1,1)} \boldsymbol{v}^{(1)} + W_s^{(1,2)} \boldsymbol{v}^{(2)} - \left((1-\lambda_s) D_s^{(1,1)} \boldsymbol{v}^{(1)}\right) \right\|_2^2 \\ & \qquad + \left\| W_s^{(2,1)} \boldsymbol{v}^{(1)} + W_s^{(2,2)} \boldsymbol{v}^{(2)} - \left((1-\lambda_s) D_s^{(2,2)} \boldsymbol{v}^{(2)}\right) \right\|_2^2 \Big). \end{aligned} \tag{4.2}$$

Let $\mathbf{1}_n$ be the n-dimensional vector with one in all entries, $\mathbf{0}_m$ be the m-dimensional vector with zero in all entries, $S = \mathrm{diag}\,(s_1, s_2, \ldots, s_m) \in \mathbb{R}^{m\times m}$, $\boldsymbol{s} = [s_1, s_2, \ldots, s_m]^\mathsf{T} \in \mathbb{R}^m$, $\boldsymbol{s}^2 = [{s_1}^2, {s_2}^2, \ldots, {s_m}^2]^\mathsf{T} \in \mathbb{R}^m$, $\boldsymbol{\alpha}_i = [{x_{i,1}^{(1)}}^2, {x_{i,2}^{(1)}}^2, \ldots, {x_{i,m}^{(1)}}^2]^\mathsf{T} \in \mathbb{R}^m$, $\boldsymbol{\beta}^{(p,q)} = [x_{i,1}^{(p)}x_{j,1}^{(q)}, x_{i,2}^{(p)}x_{j,2}^{(q)}, \ldots, x_{i,m}^{(p)}x_{j,m}^{(q)}]^\mathsf{T} \in \mathbb{R}^m$ $(p, q = 1, 2)$, $\boldsymbol{\gamma}_i = [{x_{i,1}^{(2)}}^2, {x_{i,2}^{(2)}}^2, \ldots, {x_{i,m}^{(2)}}^2]^\mathsf{T} \in \mathbb{R}^m$, and $\boldsymbol{t} = \frac{1}{2\sigma^2}\begin{bmatrix}1 & \boldsymbol{s}^\mathsf{T} & {\boldsymbol{s}^2}^\mathsf{T}\end{bmatrix}^\mathsf{T} \in \mathbb{R}^{2m+1}$.

Formulation of the first term of (2)

The (i, j) entry of $W_s^{(1,1)}$ is

$$w_{i,j}^{(1,1)} = \begin{cases} \exp\left(-\dfrac{\left\|\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_j^{(1)}\right\|_2^2}{2\sigma^2}\right) \approx 1 - \frac{\left\|\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_j^{(1)}\right\|_2^2}{2\sigma^2} = 1 - \boldsymbol{t}^\mathsf{T}\boldsymbol{k}_{i,j}^{(1,1)}, & i \neq j, \\ 0, & i = j, \end{cases}$$

where $\boldsymbol{x}_i^{(1)}$ is the ith row of $X^{(1)}$, σ is a parameter, and $\boldsymbol{k}_{i,j}^{(1,1)} = \begin{bmatrix}\mathbf{1}_m^\mathsf{T}\left(\boldsymbol{\alpha}_i - 2\boldsymbol{\beta}_{i,j}^{(1,1)} + \boldsymbol{\alpha}_j\right) & {\mathbf{0}_{2m}}^\mathsf{T}\end{bmatrix}^\mathsf{T} \in \mathbb{R}^{2m+1}$. Here, we used the first-order approximation of the exponential function $\exp(-x) \approx 1 - x$ for $0 < x < 1$. The ith row of $W_s^{(1,1)}$ is ${\boldsymbol{w}_i^{(1,1)}}^\mathsf{T} = \tilde{\boldsymbol{e}}_i^\mathsf{T} - \boldsymbol{t}^\mathsf{T}\left[\boldsymbol{k}_{i,1}^{(1,1)}, \boldsymbol{k}_{i,2}^{(1,1)}, \ldots, \boldsymbol{k}_{i,n_1}^{(1,1)}\right]$, where $\tilde{\boldsymbol{e}}_i$ is the n_1-dimensional vector with zero in the ith entry and ones in all other entries. Thus, we have

$$W_s^{(1,1)}\boldsymbol{v}^{(1)} = (n_1 - 1)\,\mathbf{1}_{n_1} + \begin{bmatrix} -n_1\mathbf{1}_m^\mathsf{T}\boldsymbol{\alpha}_1 + \sum_{j=1}^{n_1}\mathbf{1}_m^\mathsf{T}\left(2\boldsymbol{\beta}_{1,j}^{(1,1)} - \boldsymbol{\alpha}_j\right) & \mathbf{0}^\mathsf{T} & \mathbf{0}^\mathsf{T} \\ -n_1\mathbf{1}_m^\mathsf{T}\boldsymbol{\alpha}_2 + \sum_{j=1}^{n_1}\mathbf{1}_m^\mathsf{T}\left(2\boldsymbol{\beta}_{2,j}^{(1,1)} - \boldsymbol{\alpha}_j\right) & \mathbf{0}^\mathsf{T} & \mathbf{0}^\mathsf{T} \\ \vdots & & \\ -n_1\mathbf{1}_m^\mathsf{T}\boldsymbol{\alpha}_{n_1} + \sum_{j=1}^{n_1}\mathbf{1}_m^\mathsf{T}\left(2\boldsymbol{\beta}_{n_1,j}^{(1,1)} - \boldsymbol{\alpha}_j\right) & \mathbf{0}^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{bmatrix}\boldsymbol{t}.$$

Since the (i, j) entry of $W_s^{(1,2)}$ is

$$w_{i,j}^{(1,2)} = \exp\left(-\frac{\left\|\boldsymbol{x}_i^{(1)} - S\boldsymbol{x}_j^{(2)}\right\|_2^2}{2\sigma^2}\right) \approx 1 - \boldsymbol{t}^\mathsf{T}\boldsymbol{k}_{i,j}^{(1,2)},$$

the ith row of $W_s^{(1,2)}$ is ${\boldsymbol{w}_i^{(1,2)}}^{\mathsf{T}} = \mathbf{1}_{n_2}^{\mathsf{T}} - \boldsymbol{t}^{\mathsf{T}} \left[\boldsymbol{k}_{i,1}^{(1,2)}, \boldsymbol{k}_{i,2}^{(1,2)}, \ldots, \boldsymbol{k}_{i,n_2}^{(1,2)}\right]$, where $\boldsymbol{x}_i^{(2)}$ is the ith row of $X^{(2)}$, and $\boldsymbol{k}_{i,j}^{(1,2)} = \left[\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_i \quad -2{\boldsymbol{\beta}_{i,j}^{(1,2)}}^{\mathsf{T}} \quad {\boldsymbol{\gamma}_j}^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{2m+1}$. Hence, we have

$$W_s^{(1,2)}\boldsymbol{v}^{(2)} \;=\; -n_2 b\mathbf{1}_{n_1} + b\begin{bmatrix} n_2\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_1 & -2\left(\sum_{j=1}^{n_2}\boldsymbol{\beta}_{1,j}^{(1,2)}\right)^{\mathsf{T}} & \left(\sum_{j=1}^{n_2}\boldsymbol{\gamma}_j\right)^{\mathsf{T}} \\ n_2\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_2 & -2\left(\sum_{j=1}^{n_2}\boldsymbol{\beta}_{2,j}^{(1,2)}\right)^{\mathsf{T}} & \left(\sum_{j=1}^{n_2}\boldsymbol{\gamma}_j\right)^{\mathsf{T}} \\ \vdots & & \\ n_2\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_{n_1} & -2\left(\sum_{j=1}^{n_2}\boldsymbol{\beta}_{n_1,j}^{(1,2)}\right)^{\mathsf{T}} & \left(\sum_{j=1}^{n_2}\boldsymbol{\gamma}_j\right)^{\mathsf{T}} \end{bmatrix}\boldsymbol{t}.$$

Then, the ith diagonal entry $d_i^{(1,1)}$ of $D_s^{(1,1)}$ is

$$d_i^{(1,1)} = (n_1+n_2-1) + \boldsymbol{t}^{\mathsf{T}}\begin{bmatrix} -(n_1+n_2)\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_i + \sum_{j=1}^{n_1}\mathbf{1}_m^{\mathsf{T}}\left(2\boldsymbol{\beta}_{i,j}^{(1,1)} - \boldsymbol{\alpha}_j\right) \\ 2\sum_{j=1}^{n_2}\boldsymbol{\beta}_{i,j}^{(1,2)} \\ -\sum_{j=1}^{n_2}\boldsymbol{\gamma}_j \end{bmatrix}.$$

Hence, we have

$$\begin{aligned} D_s^{(1,1)}\boldsymbol{v}^{(1)} &= (n_1+n_2-1)\mathbf{1}_{n_1} \\ &+ \begin{bmatrix} -(n_1+n_2)\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_1 + \sum_{j=1}^{n_1}\mathbf{1}_m^{\mathsf{T}}\left(2\boldsymbol{\beta}_{1,j}^{(1,1)} - \boldsymbol{\alpha}_j\right) & 2\left(\sum_{j=1}^{n_2}\boldsymbol{\beta}_{1,j}^{(1,2)}\right)^{\mathsf{T}} & -\left(\sum_{j=1}^{n_2}\boldsymbol{\gamma}_j\right)^{\mathsf{T}} \\ -(n_1+n_2)\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_2 + \sum_{j=1}^{n_1}\mathbf{1}_m^{\mathsf{T}}\left(2\boldsymbol{\beta}_{2,j}^{(1,1)} - \boldsymbol{\alpha}_j\right) & 2\left(\sum_{j=1}^{n_2}\boldsymbol{\beta}_{2,j}^{(1,2)}\right)^{\mathsf{T}} & -\left(\sum_{j=1}^{n_2}\boldsymbol{\gamma}_j\right)^{\mathsf{T}} \\ \vdots & & \\ -(n_1+n_2)\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_{n_1} + \sum_{j=1}^{n_1}\mathbf{1}_m^{\mathsf{T}}\left(2\boldsymbol{\beta}_{n_1,j}^{(1,1)} - \boldsymbol{\alpha}_j\right) & 2\left(\sum_{j=1}^{n_2}\boldsymbol{\beta}_{n_1,j}^{(1,2)}\right)^{\mathsf{T}} & -\left(\sum_{j=1}^{n_2}\boldsymbol{\gamma}_j\right)^{\mathsf{T}} \end{bmatrix}\boldsymbol{t}. \end{aligned}$$

Therefor, the first term of equation (4.2) can be written as

$$\begin{aligned} & \|W_s^{(1,1)}\boldsymbol{v}^{(1)} + W_s^{(1,2)}\boldsymbol{v}^{(2)} - (1-\lambda_1)\,D_s^{(1,1)}\boldsymbol{v}^{(1)}\|_2^{\,2} \\ \Leftrightarrow\;& \|E_1 + \begin{bmatrix} E_2 & bG & bH \end{bmatrix}\boldsymbol{t} - \mu\left(F_1 + \left(\begin{bmatrix} F_2 & -G & -H \end{bmatrix}\boldsymbol{t}\right)\right)\|_2^{\,2} \\ \Leftrightarrow\;& \|\left(\left(E_1 + \frac{E_2}{2\sigma^2}\right) + \frac{bG}{2\sigma^2}\boldsymbol{s} + \frac{bH}{2\sigma^2}\boldsymbol{s}^2\right) - \mu\left(\left(F_1 + \frac{F_2}{2\sigma^2}\right) - \frac{G}{2\sigma^2}\boldsymbol{s} - \frac{H}{2\sigma^2}\boldsymbol{s}^2\right)\|_2^{\,2}, \end{aligned} \tag{4.3}$$

where $\mu = 1-\lambda_1$, the ith entry of $E_1 \in \mathbb{R}^{n_1}$ is $n_1 - n_2 b - 1$, the ith entry of $E_2 \in \mathbb{R}^{n_1}$ is $(-n_1+n_2 b)\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_i + \sum_{j=1}^{n_2}\mathbf{1}_m^{\mathsf{T}}\left(2\boldsymbol{\beta}_{i,j}^{(1,1)} - \boldsymbol{\alpha}_j\right)$, the ith row of $G \in \mathbb{R}^{n_1\times m}$ is $-2\sum_{j=1}^{n_2}{\boldsymbol{\beta}_{i,j}^{(1,2)}}^{\mathsf{T}}$, the ith row of $H \in \mathbb{R}^{n_1\times m}$ is $\sum_{j=1}^{n_2}{\boldsymbol{\gamma}_j}^{\mathsf{T}}$, the ith entry of $F_1 \in \mathbb{R}^{n_1}$ is n_1+n_2-1, and the ith entry of $F_2 \in \mathbb{R}^{n_1}$ is $(-n_1-n_2)\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_i + \sum_{j=1}^{n_1}\mathbf{1}_m^{\mathsf{T}}\left(2\boldsymbol{\beta}_{i,j}^{(1,1)} - \boldsymbol{\alpha}_j\right)$.

Formulation of the second term of (2)

The (i,j) entry of $W_s^{(1,2)}$ is

$$w_{i,j}^{(2,1)} = \exp\left(-\frac{\|S\boldsymbol{x}_i^{(2)} - \boldsymbol{x}_j^{(1)}\|_2^{\,2}}{2\sigma^2}\right) \approx 1 - \frac{\|S\boldsymbol{x}_i^{(2)} - \boldsymbol{x}_j^{(1)}\|_2^{\,2}}{2\sigma^2} = 1 - \boldsymbol{t}^\mathsf{T}\boldsymbol{k}_{i,j}^{(2,1)},$$

where $\boldsymbol{k}_{i,j}^{(2,1)} = \begin{bmatrix}\mathbf{1}_m^\mathsf{T}\boldsymbol{\alpha}_j & -2\boldsymbol{\beta}_{i,j}^{(2,1)^\mathsf{T}} & \boldsymbol{\gamma}_i^\mathsf{T}\end{bmatrix}^\mathsf{T} \in \mathbb{R}^{2m+1}$. The ith row of $W_s^{(2,1)}$ is $\boldsymbol{w}_i^{(2,1)^\mathsf{T}} = \mathbf{1}_{n_1}^\mathsf{T} - \boldsymbol{t}^\mathsf{T}\left[\boldsymbol{k}_{i,1}^{(2,1)}, \boldsymbol{k}_{i,2}^{(2,1)}, \ldots, \boldsymbol{k}_{i,n_1}^{(2,1)}\right]$. Hence, we have

$$W_s^{(2,1)}\boldsymbol{v}^{(1)} = n_1\mathbf{1}_{n_2} + \begin{bmatrix} -\sum_{j=1}^{n_1}\mathbf{1}_m^\mathsf{T}\boldsymbol{\alpha}_j & 2\left(\sum_{j=1}^{n_1}\boldsymbol{\beta}_{1,j}^{(2,1)}\right)^\mathsf{T} & -n_1\boldsymbol{\gamma}_1^\mathsf{T} \\ -\sum_{j=1}^{n_1}\mathbf{1}_m^\mathsf{T}\boldsymbol{\alpha}_j & 2\left(\sum_{j=1}^{n_1}\boldsymbol{\beta}_{2,j}^{(2,1)}\right)^\mathsf{T} & -n_1\boldsymbol{\gamma}_2^\mathsf{T} \\ & \vdots & \\ -\sum_{j=1}^{n_1}\mathbf{1}_m^\mathsf{T}\boldsymbol{\alpha}_j & 2\left(\sum_{j=1}^{n_1}\boldsymbol{\beta}_{n_2,j}^{(2,1)}\right)^\mathsf{T} & -n_1\boldsymbol{\gamma}_{n_2}^\mathsf{T} \end{bmatrix}\boldsymbol{t}.$$

Then, the (i,j) entry of $W_s^{(2,2)}$ is

$$w_{i,j}^{(2,2)} = \begin{cases} \exp\left(-\dfrac{\|S\boldsymbol{x}_i^{(2)} - S\boldsymbol{x}_j^{(2)}\|_2^{\,2}}{2\sigma^2}\right) \approx 1 - \frac{\|S\boldsymbol{x}_i^{(2)} - S\boldsymbol{x}_j^{(2)}\|_2^{\,2}}{2\sigma^2} = 1 - \boldsymbol{t}^\mathsf{T}\boldsymbol{k}_{i,j}^{(2,2)}, & i \neq j, \\ 0, & i = j, \end{cases}$$

where $\boldsymbol{k}_{i,j}^{(2,2)} = \begin{bmatrix}\mathbf{0}_{m+1} & \boldsymbol{\gamma}_i^\mathsf{T} - 2\boldsymbol{\beta}_{i,j}^{(2,2)^\mathsf{T}} + \boldsymbol{\gamma}_j^\mathsf{T}\end{bmatrix}^\mathsf{T} \in \mathbb{R}^{2m+1}$. The ith row of $W_s^{(2,2)}$ is $\boldsymbol{w}_i^{(2,2)^\mathsf{T}} = \hat{\boldsymbol{e}}_i^\mathsf{T} - \boldsymbol{t}^\mathsf{T}\left[\boldsymbol{k}_{i,1}^{(2,2)}, \boldsymbol{k}_{i,2}^{(2,2)}, \ldots, \boldsymbol{k}_{i,n_2}^{(2,2)}\right]$, where $\hat{\boldsymbol{e}}_i$ is the n_2-dimensional vector with zero in the ith entry and ones in all other entries. Hence, we have

$$W_s^{(2,2)}\boldsymbol{v}^{(2)} = -b\left(n_2 - 1\right)\mathbf{1}_{n_2} + b\begin{bmatrix} 0 & \mathbf{0}^\mathsf{T} & \left(n_2\boldsymbol{\gamma}_1 - \sum_{j=1}^{n_2}\left(2\boldsymbol{\beta}_{1,j}^{(2,2)} - \boldsymbol{\gamma}_j\right)\right)^\mathsf{T} \\ 0 & \mathbf{0}^\mathsf{T} & \left(n_2\boldsymbol{\gamma}_2 - \sum_{j=1}^{n_2}\left(2\boldsymbol{\beta}_{2,j}^{(2,2)} - \boldsymbol{\gamma}_j\right)\right)^\mathsf{T} \\ \vdots & \vdots & \vdots \\ 0 & \mathbf{0}^\mathsf{T} & \left(n_2\boldsymbol{\gamma}_{n_2} - \sum_{j=1}^{n_2}\left(2\boldsymbol{\beta}_{n_2,j}^{(2,2)} - \boldsymbol{\gamma}_j\right)\right)^\mathsf{T} \end{bmatrix}\boldsymbol{t}.$$

Then, the i the diagonal entry $d_i^{(2,2)}$ of $D_s^{(2,2)}$ is

$$d_i^{(2,2)} = (n_1 + n_2 - 1) + \boldsymbol{t}^\mathsf{T}\begin{bmatrix} -\sum_{j=1}^{n_1}\mathbf{1}_m^\mathsf{T}\boldsymbol{\alpha}_j \\ 2\sum_{j=1}^{n_1}\boldsymbol{\beta}_{i,j}^{(2,1)} \\ -(n_1 + n_2)\boldsymbol{\gamma}_i + \sum_{j=1}^{n_2}\left(2\beta_{i,j}^{(2,2)} - \boldsymbol{\gamma}_j\right) \end{bmatrix}.$$

Hence, we have

$$
\begin{aligned}
D_s^{(2,2)}\boldsymbol{v}^{(2)} &= -b(n_1+n_2-1)\mathbf{1}_{n_2} \\
&+b\begin{bmatrix}
\sum_{j=1}^{n_1}\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_j & -2\left(\sum_{j=1}^{n_1}\boldsymbol{\beta}_{1,j}^{(2,1)}\right)^{\mathsf{T}} & \left((n_1+n_2)\boldsymbol{\gamma}_1-\sum_{j=1}^{n_2}\left(2\boldsymbol{\beta}_{1,j}^{(2,2)}-\boldsymbol{\gamma}_j\right)\right)^{\mathsf{T}} \\
\sum_{j=1}^{n_1}\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_j & -2\left(\sum_{j=1}^{n_1}\boldsymbol{\beta}_{2,j}^{(2,1)}\right)^{\mathsf{T}} & \left((n_1+n_2)\boldsymbol{\gamma}_2-\sum_{j=1}^{n_2}\left(2\boldsymbol{\beta}_{2,j}^{(2,2)}-\boldsymbol{\gamma}_j\right)\right)^{\mathsf{T}} \\
 & \vdots & \\
\sum_{j=1}^{n_1}\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_j & -2\left(\sum_{j=1}^{n_1}\boldsymbol{\beta}_{n_2,j}^{(2,1)}\right)^{\mathsf{T}} & \left((n_1+n_2)\boldsymbol{\gamma}_{n_2}-\sum_{j=1}^{n_2}\left(2\boldsymbol{\beta}_{n_2,j}^{(2,2)}-\boldsymbol{\gamma}_j\right)\right)^{\mathsf{T}}
\end{bmatrix}\boldsymbol{t}.
\end{aligned}
$$

Therefore, the second term of equation of (4.2) can be written as

$$
\begin{aligned}
& \|W_s^{(2,1)}\boldsymbol{v}^{(1)}+W_s^{(2,2)}\boldsymbol{v}^{(2)}-(1-\lambda)D_s^{(2,2)}\boldsymbol{v}^{(2)}\|_2^{\,2} \\
\Leftrightarrow\ & \|\tilde{E}_1+\begin{bmatrix}-\tilde{G} & \tilde{H} & \tilde{E}_2\end{bmatrix}\boldsymbol{t}-\mu\left(\tilde{F}_1+\begin{bmatrix}\tilde{G} & -\tilde{H} & \tilde{F}_2\end{bmatrix}\boldsymbol{t}\right)\|_2^{\,2} \\
\Leftrightarrow\ & \|\left(\left(\tilde{E}_1-\frac{\tilde{G}}{2b\sigma^2}\right)+\frac{\tilde{H}}{2b\sigma^2}\boldsymbol{s}+\frac{\tilde{E}_2}{2\sigma^2}\boldsymbol{s}^2\right)-\mu\left(\left(\tilde{F}_1+\frac{\tilde{G}}{2\sigma^2}\right)-\frac{\tilde{H}}{2\sigma^2}\boldsymbol{s}+\frac{\tilde{F}_2}{2\sigma^2}\boldsymbol{s}^2\right)\|_2^{\,2}
\end{aligned}
\tag{4.4}
$$

where the ith entry of $\tilde{E}_1 \in \mathbb{R}^{n_2}$ is $n_1 - n\,(n_2-1)$, the ith entry of $\tilde{G} \in \mathbb{R}^{n_2}$ is $b\sum_{j=1}^{n_1}\mathbf{1}_m^{\mathsf{T}}\boldsymbol{\alpha}_j$, the ith row of $\tilde{E}_2 \in \mathbb{R}^{n_2\times m}$ is $\left((n_2 b-n_1)\boldsymbol{\gamma}_i+\sum_{j=1}^{n_2}\left(2\boldsymbol{\beta}_{i,j}^{(2,2)}-\boldsymbol{\gamma}_j\right)\right)^{\mathsf{T}}$, the i th row of $\tilde{H} \in \mathbb{R}^{n_2\times m}$ is $2b\left(\sum_{j=1}^{n_1}\boldsymbol{\beta}_{i,j}^{(2,1)}\right)^{\mathsf{T}}$, the ith entry of $\tilde{F}_1 \in \mathbb{R}^{n_2}$ is $-b(n_1+n_2-1)$, and the ith row of $\tilde{F}_2 \in \mathbb{R}^{n_2\times m}$ is $b\left((n_1+n_2)\boldsymbol{\gamma}_i-\sum_{j=1}^{n_2}\left(2\boldsymbol{\beta}_{i,j}^{(2,2)}-\boldsymbol{\gamma}_j\right)\right)^{\mathsf{T}}$.

Approximate the solution of (2)

From (4.3) and (4.4), we can obtain the factors s_i $(i=1,2,\ldots,m)$ by solving the optimization problem

$$
\max_{\boldsymbol{s},\mu}\|\left(A_1+A_2\boldsymbol{s}+A_3\boldsymbol{s}^2\right)-\mu\left(B_1+B_2\boldsymbol{s}+B_3\boldsymbol{s}^2\right)\|_2^{\,2},
$$

where

$$A_1 = \begin{bmatrix} E_1 + (E_2/2\sigma^2) \\ \tilde{E}_1 - \left(\tilde{G}/2b\sigma^2\right) \end{bmatrix} \in \mathbb{R}^{n_1+n_2}, \quad A_2 = \begin{bmatrix} bG/2\sigma^2 \\ \tilde{H}/2b\sigma^2 \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times m},$$

$$A_3 = \begin{bmatrix} bH/2\sigma^2 \\ \tilde{E}_2/2\sigma^2 \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times m}, \quad B_1 = \begin{bmatrix} F_1 + (F_2/2\sigma^2) \\ \tilde{F}_1 + \left(\tilde{G}/2\sigma^2\right) \end{bmatrix} \in \mathbb{R}^{n_1+n_2},$$

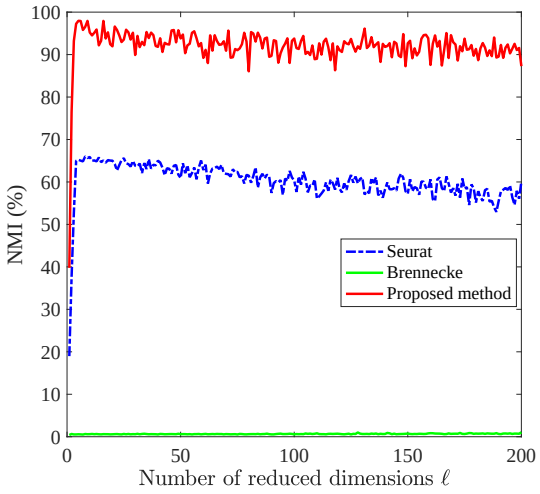
$$B_2 = \begin{bmatrix} -G/2\sigma^2 \\ -\tilde{H}/2\sigma^2 \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times m}, \quad B_3 = \begin{bmatrix} -H/2\sigma^2 \\ \tilde{F}_2/2\sigma^2 \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times m}.$$

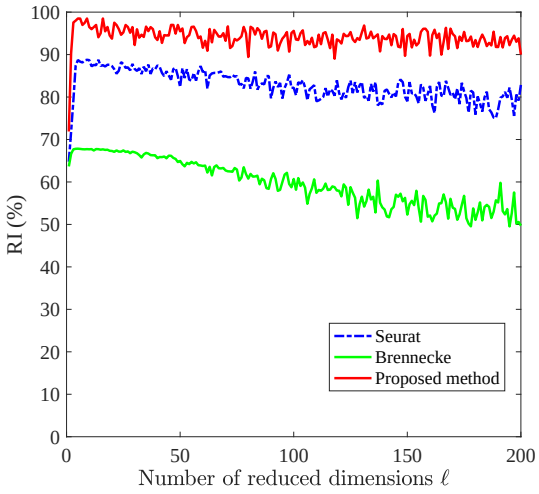
Appendix B: Clustering performance

We show the clustering performance by varying the number of reduced dimensions of PCA for three datasets.

Simulation datasets results

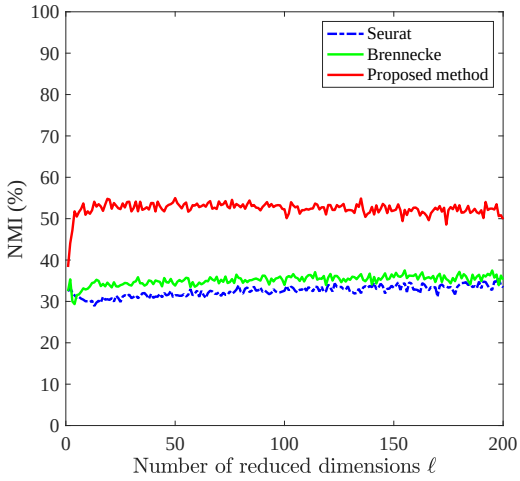
We select 200 genes for each method.

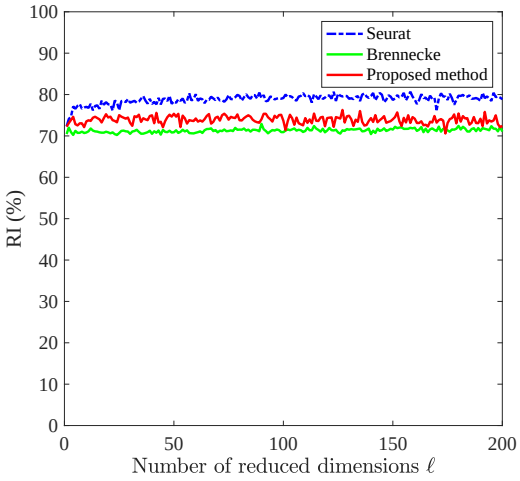




Gierahn dataset

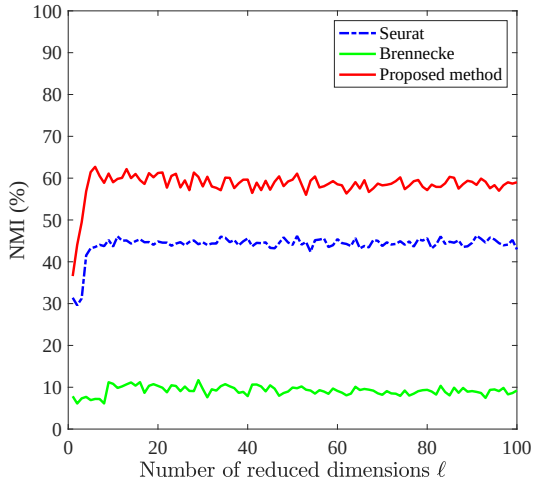
We select 200 genes for each method.

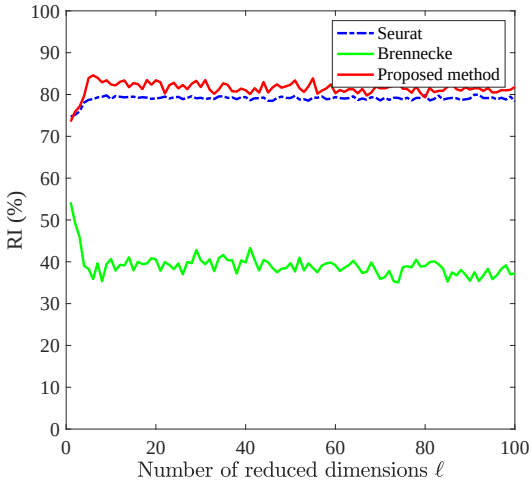




Pollen dataset

We select 100 genes for each method.





Bibliography

[1] Simone Picelli, Asa K Björklund Omid R Faridani and, Gösta Winberg, Sven Sagasser, and Rickard Sandberg. Full-length RNA-seq from single cells using Smart-seq2. *Nature Protocols*, 9(1):171–181, 2014.

[2] Tamar Hashimshony, Florian Wagner, Noa Sher, and Itai Yanai. CEL-Seq: Single-Cell RNA-Seq by Multiplexed Linear Amplification. *Cell Reports*, 2(3):666–673, 2012.

[3] Evan Z. Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa GoldmanItay Tirosh, Allison R. Bialas, Nolan Kamitaki, Emily M. Martersteck, John J. Trombetta, David A. Weitz, Joshua R. Sanes, Alex K. Shalek, Aviv Regev, and Steven A. McCarroll. Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell*, 161(5):1202–1214, 2015.

[4] Tamar Hashimshony, Naftalie Senderovich, Gal Avital, Agnes Klochendler, Yaron de Leeuw, Leon Anavy, Dave Gennert, Shuqiang Li, Kenneth J. Livak, Orit Rozenblatt-Rosen, Yuval Dor, Aviv Regev, and Itai Yanai. CEL-Seq2: sensitive highly-multiplexed single-cell RNA-Seq. *Genome Biology*, 17(1):77, 2016.

[5] Allon M. Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A. Weitz, and Marc W. Kirschner. Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. *Cell*, 161(5):1187–1201, 2015.

[6] Diego Adhemar Jaitin, Ephraim Kenigsberg, Hadas Keren-Shaul, Naama Elefant, Franziska Paul, Irina Zaretsky, Alexander Mildner, Nadav Cohen, Steffen Jung, Amos Tanay, and Ido Amit. Massively parallel single cell RNA-Seq for marker-free decomposition of tissues into cell types. *Science*, 343(6172):776–779, 2015.

[7] Todd M Gierahn, Marc H Wadsworth II, Travis K Hughes, Bryan D Bryson, Andrew Butler, Rahul Satija, Sarah Fortune, J Christopher Love, and Alex K Shalek. Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput. *Nature Methods*, 14(4):395–398, 2017.

[8] Saiful Islam, Amit Zeisel, Simon Joost, Gioele La Manno, Pawel Zajac, Maria Kasper, Peter Lönnerberg, and Sten Linnarsson. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature Methods*, 11(2):163–166, 2014.

[9] Karthik Shekhar, Sylvain W Lapan, Irene E Whitney, Nicholas M Tran, Evan Z Macosko, Monika Kowalczyk, Xian Adiconis, Joshua Z Levin, James Nemesh, Melissa Goldman, Steven A McCarroll, Constance L Cepko, Aviv Regev, and Joshua R Sanes. Comprehensive Classification of Retinal Bipolar neurons by Single-Cell Transcriptomics. *Cell*, 166:1308–1323, 2016.

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

[11] Andrea Ocone, Laleh Haghverdi, Nikola S Mueller, and Fabian J Theis. Reconstructing gene regulatory dynamics from high-dimensional single-cell snapshot data. *Bioinformatics*, 31(12):i89–i96, 2015.

[12] Aisha A. AlJanahi, Mark Danielsen, and Cynthia E. Dunbar. An introduction to the analysis of single-cell RNA-sequencing data. *Molecular Therapy - Methods & Clinical Development*, 10:189–196, 2018.

[13] Peng Qiu. Embracing the dropouts in single-cell RNA-seq analysis. *Nature Communications*, 11(1169), 2020.

[14] Peter V Kharchenko, Lev Silberstein, and David T Scadden. Bayesian approach to single-cell differential expression analysis. *Nature Methods*, 11(7):740–742, 2014.

[15] Simon Anders, Davis J McCarthy, Yunshun Chen, Michal Okoniewski, Gordon K Smyth, Wolfgang Huber, and Mark D Robinson. Count-based differential expression analysis of RNA sequencing data using R and Bioconductor. *Nature Protocols*, 8(9):1765–1786, 2013.

[16] Sophie Lamarre, Pierre Frasse, Mohamed Zouine, Delphine Labourdette, Elise Sainderichin, Guojian Hu, Véronique Le Berre-Anton, Mondher Bouzayen, and Elie Maza. Optimization of an RNA-Seq Differential Gene Expression Analysis Depending on Biological Replicate Number and Library Size. *Frontiers in Plant Science*, 9:108, 2018.

[17] Adam McDermaid, Brandon Monier, Jing Zhao, Bingqiang Liu, and Bingqiang LiuQin Ma. Interpretation of differential gene expression resultsof RNA-seq data: review and integration. *Briefings in Bioinformatics*, 20(6):2044–2054, 2018.

[18] Charlotte Soneson and Mauro Delorenzi. A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics*, 14:91, 2013.

[19] Cole Trapnell, Adam Roberts, Loyal Goff, Geo Pertea, Daehwan Kim, David R Kelley, Harold Pimentel, Steven L Salzberg, John L Rinn, and Lior Pachter. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols*, 7(3):562–578, 2012.

[20] Charlotte Soneson, Michael I. Love, and Mark D. Robinson. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Research*, 7(3):562–578, 2015.

[21] Purvesh Khatri, Marina Sirota, and Atul J Butte. Ten Years of Pathway Analysis: Current Approaches and Outstanding Challenges. *PLoS Computational Biology*, 8(2):e1002375, 2012.

[22] Şerban Nacu, Rebecca Critchley-Thorne, Peter Lee, and Susan Holmes and. Gene expression network analysis and applications toimmunology. *Bioinformatics*, 23(7):850–858, 2007.

[23] Mukesh Bansal, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego di Bernardo. How to infer gene networks from expression profiles. *Molecular Systems Biology*, 3:78, 2007.

[24] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene monitoring. *Science*, 286(5439):531–537, 1999.

[25] Linda Vidman, David Källberg, and Patrik Rydén. Cluster analysis on high dimensional RNA-seq data with applications to cancer research - An evaluation study. *PLoS One*, 14(12):e0219102, 2019.

[26] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6(2):559–572, 1901.

[27] Laurens van der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[28] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software,*, 3(29):861, 2018.

[29] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clustersin Large Spatial Databases with Noise. In *Pro-*

ceedings of the Second International Conference on Knowledge Discovery and Data Mining, page 226–231, 1996.

[30] Paul T. Spellman, Gavin Sherlock, Michael Q. Zhang, Vishwanath R. Iyer, Kirk Anders, Michael B. Eisen, Patrick O. Brown, David Botstein, and Bruce Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9(12):3273–3297, 1998.

[31] Greg Bloom, Ivana V. Yang, David Boulware, Ka Yin Kwong, Domenico Coppola, Steven Eschrich, John Quackenbush, and Timothy J. Yeatman. Multi-platform, multi-site, microarray-based human tumor classification. *The American Journal of Pathology*, 164(1):9–16, 2004.

[32] Sridhar Ramaswamy, Pablo Tamayo, Ryan Rifkin, Sayan Mukherjee, Chen-Hsiang Yeang, Michael Angelo, Christine Ladd, Michael Reich, Eva Latulippe, Jill P. Mesirov, Tomaso Poggio, William Gerald, Massimo Loda, Eric S. Lander, and Todd R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America*, 98(26):15149–15154, 2001.

[33] Kerby A. Shedden, Jeremy M.G. Taylor, Thomas J. Giordano, Rork Kuick, David E. Misek, Gad Rennert, Donald R. Schwartz, Stephen B. Gruber, Craig Logsdon, Diane Simeone, Sharon L.R. Kardia, Joel K. Greenson, Kathleen R. Cho, David G. Beer, Eric R. Fearon, and Samir Hanash. Accurate Molecular Classification of Human Cancers Based on Gene Expression Using a Simple Classifier with a Pathological Tree-Based Framework. *The American Journal of Pathology*, 163(5):1985–1995, 2003.

[34] Richard W. Tothill, Adam Kowalczyk, Danny Rischin, Alex Bousioutas, Izhak Haviv, Ryan K. van Laar, Paul M. Waring, John Zalcberg, Robyn Ward, Andrew V.

Biankin, Robert L. Sutherland, Susan M. Henshall, Kwun Fong, Jonathan R. Pollack, David D.L. Bowtell, and Andrew J. Holloway. An Expression-Based Site of Origin Diagnostic Method Designed for Clinical Application to Cancer of Unknown Origin. *American Association for Cancer Research*, 65(10):4031–4040, 2005.

[35] Nitzan Rosenfeld, Ranit Aharonov, Eti Meiri, Shai Rosenwald, Yael Spector, Merav Zepeniuk, Hila Benjamin, Norberto Shabes, Sarit Tabak, Asaf Levy, Danit Lebanony, Yaron Goren, Erez Silberschein, Nurit Targan, Alex Ben-Ari, Shlomit Gilad, Netta Sion-Vardy, Ana Tobar, Meora Feinmesser, Oleg Kharenko, Ofer Nativ, Dvora Nass, Marina Perelman, Ady Yosepovich, Bruria Shalmon, Sylvie Polak-Charcon, Eddie Fridman, Amir Avniel, Isaac Bentwich, Zvi Bentwich, Dalia Cohen, Ayelet Chajut, and Iris Barshack. MicroRNAs accurately identify cancer tissue origin. *Nature Biotechnology*, 26(4):462–469, 2008.

[36] Kalle A Ojala, Sami K Kilpinen, and Olli P Kallioniemi. Classification of unknown primary tumors with a data-driven method based on a large microarray reference database. *Genome Medicine*, 3:63, 2011.

[37] Douglas T. Ross, Uwe Scherf, Michael B. Eisen, Charles M. Perou, Christian Rees, Paul Spellman, Vishwanath Iyer, Stefanie S. Jeffrey, Matt Van de Rijn, Mark Waltham, Alexander Pergamenschikov, Jeffrey C.F. Lee, Deval Lashkari, Dari Shalon, Timothy G. Myers, John N. Weinstein, David Botstein, and Patrick O. Brown. Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, 24:pages227–235, 2000.

[38] Hugo M. Horlings, Ryan K. van Laar, Jan-Martijn Kerst, Helgi H. Helgason, Jelle Wesseling, Jacobus J.M. van der HoevenMarc O. Warmoes, Arno Floore, Anke Witteveen, Jaana Lahti-Domenici, Annuska M. Glas, Laura J. Van't Veer, and Daphne de Jong. Gene Expression Profiling to Identify the Histogenetic Origin of

Metastatic Adenocarcinomas of Unknown Primary. *Journal of Clinical Oncology*, 26(27):4435–4441, 2008.

[39] Gerald Quon and Quaid Morris. ISOLATE: a computational strategy for identifying the primary origin of cancers using high-throughput sequencing. *Bioinformatics*, 25(21):2882–2889, 2009.

[40] Tallulah S.Andrews and MartinHemberg. Identifying cell populations with scRNASeq. *Molecular Aspects of Medicine*, 59:114–122, 2018.

[41] Guo-Cheng Yuan, Long Cai, Michael Elowitz, Tariq Enver, Guoping Fan, Guoji Guo, Rafael Irizarry, Peter Kharchenko, Junhyong Kim, Stuart Orkin, John Quackenbush, Assieh Saadatpour, Timm Schroeder, Ramesh Shivdasani, and Itay Tirosh. Challenges and emerging directions in single-cell analysis. *Genome Biology*, 18:84, 2017.

[42] Philipp Angerer, Lukas Simon, Sophie Tritschler, F. Alexander Wolf, David Fischer, and Fabian J. Theis. Single cells make big data: new challenges and opportunities in transcriptomics. *Current Opinion in Systems Biology*, 4:85–91, 2017.

[43] Vilas Menon. Clustering single cells: a review of approaches on high-and low-depth single-cell RNA-seq data. *Briefings in Functional Genomics*, 17(4):240–245, 2017.

[44] Chen Xu and Zhengchang Su. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, 31(12):1974–1980, 2015.

[45] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database. *PLoS Computational Biology*, 14(6):e1006245, 2018.

[46] Angelo Duò, Mark D. Robinson, and Charlotte Soneson. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research*, 7:1141, 2018.

[47] Laleh Haghverdi, Aaron T. L. Lun, Michael D. Morgan, and John C. Marioni. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421–429, 2018.

[48] Matthew E. Ritchie, Belinda Phipson, Di Wu, Yifang Hu, Charity W. Law, Wei Shi, and Gordon K. Smyth. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Research*, 43(7):e47, 2015.

[49] Jeffrey T. Leek and John D. Storey. Capturing Heterogeneity in Gene Expression Studies by Surrogate Variable Analysis. *PLoS Genetics*, 3(9):e161, 2007.

[50] Jeffrey T. Leek. svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Research*, 42(21):e161, 2014.

[51] Davide Risso, John Ngai, Terence. P. Speed, and Sandrine Dudoit. Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature Biotechnology*, 32(9):896–905, 2014.

[52] W. Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8(1):118–127, 2007.

[53] Cheng Li and Wing Hung Wong. *DNA-Chip Analyzer (dChip)*, pages 120–141. Springer, 2003.

[54] Catalina A. Vallejos, John C. Marioni, and Sylvia Richardson. BASiCS: Bayesian Analysis of Single-Cell Sequencing Data. *PLoS Computational Biology*, 11(6):e1004333, 2015.

[55] Aaron T. L. Lun, Karsten Bach, and John C. Marioni. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biology*, 17:75, 2016.

[56] Aaron T. L. Lun, Karsten Bach, and John C. Marioni. A step-by-step workflow for low-level analysis ofsingle-cell RNA-seq data with Bioconductor. *F1000Research*, 5:2122, 2016.

[57] Florian Buettner, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells. *Nature Biotechnology*, 33:155–160, 2015.

[58] Hung-I Harry Chen, Yufang Jin, Yufei Huang, and Yidong Chen. Detection of high variability in gene expression from single-cell RNA-seq profiling. *BMC Genomics*, 17:508, 2016.

[59] Philip Brennecke, Simon Anders, Jong Kyoung Kim, Aleksandra A Kołodziejczyk, Xiuwei Zhang, Valentina Proserpio, Bianka Baying, Vladimir Benes, Sarah A Teichmann, John C Marioni, and Marcus G Heisler. Accounting for technical noise in single-cell RNA-seq experiments. *Nature Methods*, 10:1093–1095, 2013.

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

[61] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of Single-Cell Data. *Cell*, 177:1888–10–2, 2019.

[62] Christian Mayer, Christoph Hafemeister, Rachel C. Bandler, Robert Machold, Renata Batista Brito, Xavier Jaglin, Kathryn Allaway, Andrew Butler, Gord Fishell, and Rahul Satija. Developmental diversification of cortical inhibitory interneurons. *Nature*, 555:457–462, 2018.

[63] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[64] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[65] Jeffrey C. Lagarias, James A. Reeds, Margaret H. Wright, and Paul E Wright. Convergence Properties of the Nelder–Mead Simplex Method in Low Dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.

[66] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[67] Ron Edgar, Michael Domrachev, and Alex E Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002. https://www.ncbi.nlm.nih.gov/.

[68] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 5:621–628, 2008.

[69] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[70] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, December 1971.

[71] Todd M Gierahn, Marc H Wadsworth II, Travis K Hughes, Bryan D Bryson, Andrew Butler, Rahul Satija, Sarah Fortune, J Christopher Love, and Alex K Shalek. Seq-

Well: portable, low-cost RNA sequencing of single cells at high throughput. *Nature Methods*, 14:395–398, 2017.

[72] Siyuan John Liu, Tomasz J. Nowakowski, Alex A. Pollen, Jan H. Lui, Max A. Horlbeck, Frank J. Attenello, Daniel He, Jonathan S. Weissman, Arnold R. Kriegstein, Aaron A. Diaz, and Daniel A. Lim. Single-cell analysis of long non-coding RNAs in the developing human neocortex. *Genome Biology*, 17:67, 2016.

[73] Alex A Pollen, Tomasz J Nowakowski, Joe Shuga, Xiaohui Wang, Anne A Leyrat, Jan H Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, Naveen Ramalingam, Gang Sun, Myo Thu, Michael Norris, Ronald Lebofsky, Dominique Toppani, Darnell W Kemp II, Michael Wong, Barry Clerkson, Brittnee N Jones, Shiquan Wu, Lawrence Knutsson, Beatriz Alvarado, Jing Wang, Lesley S Weaver, Andrew P May, Robert C Jones, Marc A Unger, Arnold R Kriegstein, and Jay A A West. Low-coverage single-cell mRNA sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature Biotechnology*, 32(10):pages1053–1058, 2014.

Research achievements

Peer-reviewed Journal Article

1. **Momo Matsuda**, Keiichi Morikuni, Akira Imakura, Xiucai Ye, and Tetsuya Sakurai, Multiclass spectral feature scaling method for dimensionality reduction, *Intelligent Data Analysis, in press* **24** (6), pp. 1273–1287 (2020).

2. **Momo Matsuda**, Xiucai Ye and Tetsuya Sakurai, Scaling method for batch effect correction of gene expression data based on spectral clustering, *Current Bioinformatics* **15**(0), pp.1–9 (2020). [Epub ahead of print]

Peer-reviewed International Conference

Oral Presentation

1. **Momo Matsuda**, Keiichi Morikuni, and Tetsuya Sakurai, Spectral feature scaling method for supervised dimensionality reduction, In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)* , Stockholm, pp. 2560–2566 (2018).

2. Akira Imakura, **Momo Matsuda**, Xiucai Ye, and Tetsuya Sakurai, Complex moment-based supervised eigenmap for dimensionality reduction, In: *the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, Honolulu, pp.3910-3918 (2019).

Non-Peer-reviewed International Conference

Oral Presentation

1. **Momo Matsuda**, Keiichi Morikuni, Tetsuya Sakurai, Feature scaling method for supervised spectral clustering, SIAM Conference on Parallel Processing for Science Computing, Tokyo, March, 2018.

Poster Presentation

1. **Momo Matsuda**, Keiichi Morikuni, Tetsuya Sakurai, Feature Scaling Method for Spectral Classification, International Workshop on Eigenvalue Problems: Algorithms; Software and Applications, in Petascale, Tsukuba, March, 2018.

2. **Momo Matsuda**, Keiichi Morikuni, Akira Imakura, Xiucai Ye, Tetsuya Sakurai, Supervised spectral feature scaling for classification, International Symposium on “Digital Science Now” in association with the G20 Ministerial Meeting on Trade and Digital Economy, Tsukuba, June, 2019.

Non-Peer-reviewed National Conference

Oral Presentation

1. **松田 萌望**, 保國 惠一, 櫻井 鉄也, スペクトラルクラスタリングにおける特徴量スケーリング, 日本応用数 理学会 2017 年度研究部会連合発表会, 東京, 2017 年 3 月.

2. **松田 萌望**, 保國 惠一, 櫻井 鉄也, 特徴量スケーリングを用いた教師ありスペクトラルクラスタリング, 第 46 回 数値解析シンポジウム (NAS2017), 滋賀, 2017 年 6 月.

3. **松田 萌望**, 保國 惠一, 櫻井 鉄也, 特徴量スケーリングを用いたスペクトラルクラス分類, 日本応用数理学会 2018 年度研究部会連合発表会, 大阪, 2018 年 3 月.

4. **松田 萌望**, 保國 惠一, 今倉 暁, 櫻井 鉄也, 多クラス分類問題に対するスペクトラル特徴量スケーリング, 第 47 回 数値解析シンポジウム (NAS2018), 福井, 2018 年 6 月.

5. **松田 萌望**, 保國 惠一, 今倉 暁, 櫻井 鉄也, 高次元データのスペクトラルクラス分類における特徴量スケーリング, 第 33 回 情報論的学習理論と機械学習研究会 (IBISML), 沖縄, 2018 年 6 月.

Poster Presentation

1. **松田 萌望**, 保國 惠一, 櫻井 鉄也, 特徴量スケーリングを用いた教師ありスペクトラルクラスタリング, 日本応用数理学会 2017 年度年会, 東京, 2017 年 9 月.

2. **松田 萌望**, 保國 惠一, 櫻井 鉄也, 高次元特徴量のスケーリング法と教師付きスペクトラルクラスタリング, 第 20 回 情報論的学習理論ワークショップ (IBIS2017), 東京, 2017 年 11 月.

3. **松田 萌望**, 保國 惠一, 今倉 暁, 櫻井 鉄也, スペクトラル特徴量スケーリングの多クラス分類問題への拡張, 日本応用数理学会 2018 年度年会, 愛知, 2018 年 9 月.

Award

1. 日本応用数理学会 2018 年度年会 優秀ポスター賞, 2018 年 9 月.

2. 筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻 専攻長表彰, 2019 年 3 月.