Subpopulation identification for single-cell RNA-sequencing data using functional data analysis

Kyungmin Ahn *[1] and Hironobu Fujiwara *[1]

[1]RIKEN Center for Biosystems Dynamics Research (BDR), Kobe 650-0047, Japan

Abstract

Motivation: In single-cell RNA-sequencing (scRNA-seq) analysis, a number of statistical tools in multivariate data analysis (MDA) have been developed to help analyze the gene expression data. This MDA approach is typically focused on examining discrete genomic units of genes that ignores the dependency between the data components. In this paper, we propose a functional data analysis (FDA) approach on scRNA-seq data whereby we consider each cell as a single function that does not allow permutation of the data components. To avoid a large number of dropouts (zero or zero-closed values) and reduce the high dimensionality of the data, we first perform a principal component analysis (PCA) and assign PCs to be the amplitude of the function. For the phase components, we propose two criteria: we use the PCs directly from PCA, and we sort the PCs by the genetic spatial information. For the latter, we embed the spatial information of genes by aligning the genomic gene locations to be the phase of the function. These two approaches allow us to apply FDA clustering methods to scRNA-seq analysis.

Results: To demonstrate the robustness of our method, we apply several existing FDA clustering algorithms to the gene expression data to improve the accuracy of the classification of the cell types against the conventional clustering methods in MDA. As a result, the FDA clustering algorithms achieve superior accuracy on simulated data as well as real data such as human and mouse scRNA-seq data.

keywords : functional data analysis, single-cell RNA-sequencing analysis, classification, clustering

1 Introduction

Single-cell RNA sequencing (scRNA-seq) analysis has been widely used to explore and measure the genome-wide expression profile of individual cells. Since the number of bioinformatics tools for scRNA-seq analysis is growing dramatically, there are many studies comparing several statistical methods for scRNA-seq analysis. Menon [34] reviewed three statistical clustering algorithms for scRNA-seq data to explicitly demonstrate their different behaviors in low- and high-read-depth data. Recently, Andrews and Hemberg [2] compared 12 clustering techniques on scRNA-seq data sets, therein illustrating that the different methods generally produced clustering with minimal overlap. Duò et al. [11] extended these initial studies to 14 clustering algorithms on a total of 12 different simulated and real data sets, therein showing the large differences in performance across data sets and clustering methods.

These statistical methods and algorithms for scRNA-seq analysis belong to the general framework of *multivariate data analysis* (MDA), which helps analyze the gene expression data to understand stochastic

[*] corresponding author

biological processes. However, several shortcomings have arisen when the data are treated as vectors of discrete samples instead of continuous samples. One of the great advantages of applying this framework, i.e., *functional data analysis* (FDA) [46], is that we consider the *dependency* or *connectivity* between the samples. This FDA approach can also incorporate other important variables such as *time* and *space*. Several works are performed on gene expression data by applying FDA to incorporate information that is inherent in the time and space orders and the smoothness of the process over time and space, respectively [29, 10, 38, 28, 31, 3, 54, 53].

Based on these ideas, we propose the FDA technique for scRNA-seq analysis to improve the accuracy of the classification of the cell types. An important aspect of this study is that we view the multivariate gene expression data as functional gene expression data. This different point of view from standard multivariate data analysis underlies the structure of raw observations being functional. This approach allows us to detect the functional nature of the scRNA-seq data and uncover the functional characteristics of cell populations. This eventually classifies the subpopulations of the cell types that cannot be detected by standard multivariate statistical methods. Given that a function does not allow the permutations of phase components of a function, we consider two approaches to the alignment of the phase components. One way is to directly use principal components in general, which are sorted by eigenvalues in descending order. The other way is to embed the spatial information of genes by sorting the gene locations (according to chromosome and position number) in ascending order. We applied these approaches using functional clustering methods on simulated data and real data to demonstrate the robustness of the efficiency and accuracy of the classification against the MDA clustering algorithms.

2 Methods

2.1 Pre-processing Steps

One of the crucial steps in biological experiments, such as scRNA-seq analysis, is to remove biological or technical errors in the gene expression data [26]. scrRNA-seq analysis is used to explore complex mixtures of cell types in an unsupervised manner. A standard scRNA-seq analysis involves several tasks that can be performed by various bioinformatics or biostatistics techniques. Zappia et al. [55] categorized these tasks into four broad phases of analysis: data acquisition, data cleaning, cell assignment, and gene identification. The first two phases are generally referred to as the pre-processing steps, and the last two phases are referred to as the statistical analysis steps. Data acquisition can be re-categorized as alignment, de-duplication, and quantification. Data cleaning involves quality control, normalization, and imputation. This work can be done by several existing R packages such as *SC3* [27], *Monocle* [52, 39, 40], and *Seurat* [51]. We implemented these pre-processing steps for scRNA-seq analysis using the *Seurat* 2.3.4 R packages for the downstream analysis. In particular, we normalize, find variable genes, and scale the data using *Seurat* for the analysis. Then, we use *scaled data*, which are the z-scored residuals of linear models, to predict gene expression for PCA and clustering.

2.2 Framework of building functional data

Functional data analysis was pioneered by Ramsay [41] and then expanded by Ramsay, Silverman, Dalzell, Ferraty and Vieu [46, 43, 45, 15]. A function in functional data analysis is defined in the Hilbert space $\mathcal{H}$, in particular, the $\mathbb{L}^2$ space for real square-integrable functions defined on $[a, b]$ with the inner product $\langle a, b\rangle = \int_a^b fg$. In general, we define a function from the observed multivariate data or functional data with *time* points for the downstream analysis. Then, we apply a smoothing method using a known basis for parametric methods or a kernel function for nonparametric methods. In this paper, using FDA on scRNA-seq analysis, we no longer consider *time* as a phase of the functions; however, we use the index of the principal components from PCA on the gene expression data.

One of the challenging problems in applying FDA to scRNA-seq analysis is the presence of high *dropouts*, i.e., zero-inflated counts of gene expression data. This also induces the fitting problem of constructing the function from raw multivariate gene expression data, which gives a tremendous number of spikes when constructing the functional objectives. To solve this issue, we implemented PCA to drastically lower the number of features (dimensions or genes). In this way, we can reduce not only the dimensionality but also the number of dropout values from the scRNA-seq data, which eventually smooth the original data by itself. This is also a common and general step in conventional scRNA-seq analysis for reducing the dimensionality of the gene expression data. Then, each single cell can now be considered as a single function with PC scores and the index of principal components. The important feature of this analysis is that we treat the index of the principal components as "time points", i.e., each PC acts like discrete *time* points in the functional gene expression data.

Sorting PCs Another problem in functional scRNA-seq analysis is the order of the principal components, i.e., the phase of the functions, when we build a function from scRNA-seq data. In MDA, data are considered discrete vectors; thus, the permutation of the data components is allowed in any statistical analysis. In FDA, however, the permutation of phase components will affect the statistical results in that it should be sorted in order of some characteristics of the data. Traditionally, the most general way of sorting the phase components in FDA is in time order, e.g., in seconds, months, or years. For functional scRNA-seq data analysis, we view the principal component scores of gene expression profiles as independent realizations of a smooth stochastic process. Therefore, we sort the order of the principal components with two criteria: 1) sort by eigenvalues, which is also the same framework as using the order of the PCs from PCA directly, and 2) sort by genetic spatial information.

Sort by eigenvalues After PCA, the PCs are sorted in descending order according to their variance. Without loss of generality, we discard the number of PCs according to the eigenvalues from the smallest to the largest to reduce the dimensionality of the data. One advantage of using from the largest to smallest eigenvalues here is being able to capture as much of the variance as possible without losing much information from the original data. We consider this framework, i.e., the PCs in descending order, as one criterion based on the order of the "magnitude" of the corresponding eigenvectors; thus, we treat this strength of variance as the time order to be the functional data. Hence, based on this framework, we build functional data from scRNA-seq data, which is exactly the same as using principal components directly from the results of PCA.

Sort by genetic spatial information Another way of sorting the PCs in the scRNA-seq analysis is to sort the PCs by genetic spatial information. Genes are spatially located in a genome, and this genetic spatial information can be applied to each principal component to sort the PCs on the functional gene expression data. scRNA-seq analysis is typically focused on examining discrete genomic units of genes, and this approach ignores spatial information. While this simplifies the complex data, it also loses information that may elucidate the hidden nature of the gene expression characteristics that are associated with the shape of the gene expression data [53]. This application of FDA on scRNA-seq data interprets the gene expression data as functional data that will capture the undiscovered characteristics of cell populations and will eventually help to understand stochastic biological processes such as identifying cell states and cell types. The previous sorting criterion is that we directly use the PCs from the PCA of scRNA-seq data. Here, we embed genes into the principal components. In particular, we find the first-most-variable (weighted) gene for each principal component and assign it to be the representative of the components. To sort the genes according to the genetic spatial information, we found the gene information using *ensembl* and *Genome Browser* to extract the chromosome number and molecular position number for each gene. Since there are two pieces of spatial information, the chromosome number and the molecular gene position, we first sort by chromosome number and then sort by molecular position from the smallest to the largest.

Smoothing After building functional gene expression data from discretized multivariate gene expression data, the functional data can be smoothed, leading to the Karhunen-Loéve representation of the observed sample paths as a sum of a smooth mean trend. To recover the nature of the functional statistics setting, smoothing must be performed on the discretized data, especially where observations are very noisy. A truncated version of the random part of this representation serves as a statistical approximation of the random process [48]. We first assume that the function $g(s)$ is observed through the model.

$$g_i(s_l) = f_i(s_l) + \epsilon_i(s_l), \quad i = 1, 2, \cdots, n, \quad l = 1, 2, \cdots, m$$

where $\epsilon_i(s_l)$ is the residual error, and s_l is the l-th principal component, $l = 1, \ldots, m$. Then, we can reconstruct the original function $f(s)$ from the observed function $g(s)$ using a linear smoother,

$$\hat{f} = \sum_{l=1}^{m} \xi_{lr} g_l$$

where ξ_{lr} is the weight that the point s_r gives to the point s_l and $g_l = g(s_l)$. Then, the function can be smoothed in two ways: using parametric or non-parametric methods.

Parametric smoothing method: B-spline basis A parametric method is also known as a basis representation since we use a known basis to smooth the data. There is no universal basis to use; however, we generally use a B-spline basis and a Fourier basis. A basis is a set of known functions $\{b_j\}_1^\infty$ with which any function can be arbitrarily approximated using a linear combination of a sufficiently large number J of these functions.

$$f(s) = \sum_{j=1}^{\infty} c_j b_j(s) \approx \sum_{j=1}^{J} c_j b_j(s)$$

where c_j is the coefficient of the basis function b_j.

Non-parametric smoothing method: Nadaraya-Watson estimator Non-parametric smoothing methods, also known as a kernel smoothing method, can be used to represent functional data. In this analysis, we use the Nadaraya-Watson estimator [37] with Gaussian kernel:

$$\xi_{lr} = \frac{K(\frac{s_r - s_l}{h})}{\sum_{j=1}^{J} K(\frac{s_r - s_l}{h})}$$

where $K(\cdot)$ is the kernel function and h is the bandwidth.

Generalized Cross-Validation For parametric and non-parametric smoothing methods, smoothing penalization is crucial for estimating the coefficient of the basis and kernel parameter, respectively. The choice of smoothing parameter is important; however, there is no universal criterion that would ensure an optimal choice. In general, we select the parameter η using generalized cross-validation (GCV).

$$GCV(\eta) = \frac{1}{n}\sum_{i=1}^{n}(g_i - \hat{g}_i^{\eta})^2\omega_i\Xi(\eta)$$

where $\Xi(\eta)$ denotes the type of penalizing function [23] and ω_i is the weight at point s_l.

2.3 Clustering Methods

Clustering algorithms are statistical tools used to identify the sub-population of subjects, such as cell types, in scRNA-seq analysis. We evaluate three MDA clustering algorithms and three FDA clustering algorithms on gene expression data in this paper. All methods are implemented and publicly available as R packages or scripts. See the references for each method and further details in Table 1.

Type	Methods	Description	Reference
MDA	k-means	The data given by x are clustered by the k-means method, which aims to partition the points into k groups such that the sum of squares from points to the assigned cluster centres is minimized	[25, 16, 30, 32]
	hierarchical	A hierarchical cluster analysis using a set of dissimilarities for the n objects being clustered.	[4, 36, 24, 22, 1]
	mclust	Model-based clustering based on parameterized finite Gaussian mixture models.	[49, 17, 19, 18]
FDA	k-means	The method searches the locations around which are grouped data (for a predetermined number of groups) on functional data.	[14, 25]
	funHDDC	The funHDDC (High-Dimensional Data Clustering) algorithm allows one to cluster functional univariate or multivariate data by modeling each group within a specific functional subspace.	[6]
	funFEM	The funFEM algorithm allows to cluster functional data by modeling the curves within a common and discriminative functional subspace.	[5]

Table 1: Clustering Methods on MDA and FDA

For MDA analysis, we perform k-means and hierarchical algorithms, which are the most popular clustering algorithms that have been used recently for single cell RNA sequencing analysis. Many Bioconductors, such as *Seurat, Monocle, and SC3*, perform these clustering techniques to classify the subpopulations to identify and characterize cell populations. In addition to these algorithms, we also apply a recent clustering algorithm, *mclust*, which is a model-based clustering method based on parameterized finite Gaussian mixture models. These models are estimated by the expectation-maximization (EM) algorithm initialized by the hierarchical model-based agglomerative clustering method. Then, we select the optimal model using the Bayesian information criterion (BIC). For clustering methods in FDA, the functional k-means method is the same as the one in MDA; however, we define the observations in the Hilbert space, $\mathcal{H}$. *funHDDC* is a model-based algorithm that is based on a functional latent mixture model that fits the functional data in group-specific functional subspaces. *funFEM* is also a model-based method but is based on a functional mixture model that allows the clustering of the data in a discriminative functional subspace.

3 Results

To evaluate the robustness of our approach, we perform the clustering methods that we have described in section 2.3 on both simulated data and real data. For comparison, we calculate the success rate (%) to evaluate the accuracy of the classification for each dataset and for each method. For the calculation of the success rate, we implement the classification algorithm and then compare the predicted label and true label for each cell. Then, we count the number of correctly matched labels for each cell and calculate the average out of the total number of cells.

3.1 Simulated Data

We generated the simulated data using a Gaussian process to observe the improvement in performance of the classification for FDA clustering algorithms. In this simulated experiment, we focused on comparing MDA classification and FDA classification. Therefore, we assume that all the pre-processing steps on the scRNA-seq data, such as normalization, scaling, and PCA, are performed. Then, we can only focus on how the FDA approach improves the success rate for classification against MDA classification using PC scores and PCs based on PCA.

We first consider two samples of i.i.d. curves, $X_i(s)$ and $Y_i(s)$, generated by independent stochastic processes with different means such that $X_i(s), Y_i(s) \in \mathbb{L}^2(I)$, where I is a compact interval of $\mathbb{R}$. We use Karhunen-Loéve decomposition to generate the sample curves [21, 20, 33]:

$$\begin{aligned} X_i(s) &= m_0(s) + \sum_{j=1}^{\infty} Z_{ji,1}\sqrt{\lambda_k}\theta_j(s) \qquad i = 1, \dots, n_1, \\ Y_i(s) &= m_1(s) + \sum_{j=1}^{\infty} Z_{ji,2}\sqrt{\lambda_k}\theta_j(s) \qquad i = 1, \dots, n_2, \end{aligned} \tag{1}$$

where s is the index of principal components (s_l, l is the l-th principal component); m_0 and m_1 are the mean of each sample for X_i and Y_i, respectively; $(Z_{ki,1})_{k=1}^{\infty}$, $(Z_{ki,2})_{k=1}^{\infty}$ are two sequences of independent standard normal variables; θ_j is the $J/2$ harmonic Fourier basis; and λ_k is the coefficient variance; n_1 and n_2 are the number of cells in groups 1 and 2, respectively. Because of the infinity of the basis functions, we truncate it into the finite case in terms of J known basis functions θ_j.

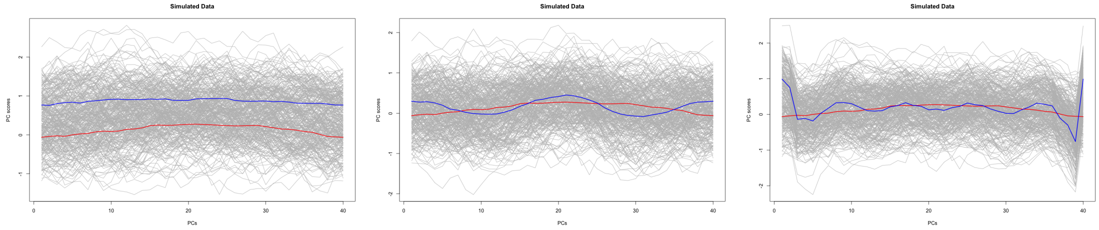
For the initial settings, we generated 150 cell functions for each sample, X and Y. Hence, we have a total of 300 cells in this simulated data set. Then, we fixed the number of principal components to 40 ($l = 1, 2, \cdots, 40$) assuming that 40 principal components are retained based on some statistical criteria. We set $J = 40$ to have sufficient peaks of the functions for the Fourier basis functions and assign $m_0(s) = s(1 - s)$ for the mean of the sample $X_i(s)$. For the coefficient variances λ_k, we set

$$\lambda_k = \begin{cases} \frac{1}{k+1} & \text{if } k \in \{1, 2, 3\} \\ \frac{1}{(k+1)^2} & \text{if } k \geq 0 \end{cases}$$

For the mean sample for $Y_i(s)$, we use three different cases to generate three different data sets to compare the classification efficiency depending on the shape and scale of the functions.

$$\begin{aligned} m_1(s) &= m_0(s) + \sqrt{\lambda_1}\theta_1 && \text{(a)} \\ &= m_0(s) + \sqrt{\lambda_5}\theta_5 && \text{(b)} \\ &= m_0(s) + \sum_{k=10}^{\infty} \sqrt{\lambda_k}\theta_k && \text{(c)} \end{aligned}$$

We assigned samples of $X_i(s)$ and $Y_i(s)$ as group 1 and group 2, respectively, to group into two different "true" groups. Based on the arguments above, we simulated three different samples of $Y_i(s)$ for the different means (a), (b), and (c). Figure 1 shows simulated data using Equation 1. The solid red and blue curves are $m_0(s)$ and $m_1(s)$, which are the means of each sample, respectively. Each panel shows different means of $Y_i(s)$ for (a), (b), and (c). The difference between the samples in the first panel (case a) is simply the amplitude of the functions. It is easy to see that the shape is very similar and that only the height (y-axis) is different. For the second case (case b), we generated a sample of Y whereby the mean of its sample lies on the same horizontal line as the mean of sample X; however, the shape of the function is different. From the second panel of Figure 1, the shape of the mean of sample X is smoother than the shape of the mean of sample Y and is a flat curve rather than several distinct peaks in Y. The last panel (case c) shows a similar generated function from the second case; however, this time, the sample Y has high peaks (spikes) at both the beginning and end of the curve, which give the sample Y a very different shape compared to sample X. Based on these three different data sets, our goal is to evaluate whether the classification algorithms can cluster into the correct group using the MDA and FDA clustering methods.



(a) Case (a)

(b) Case (b)

(c) Case (c)

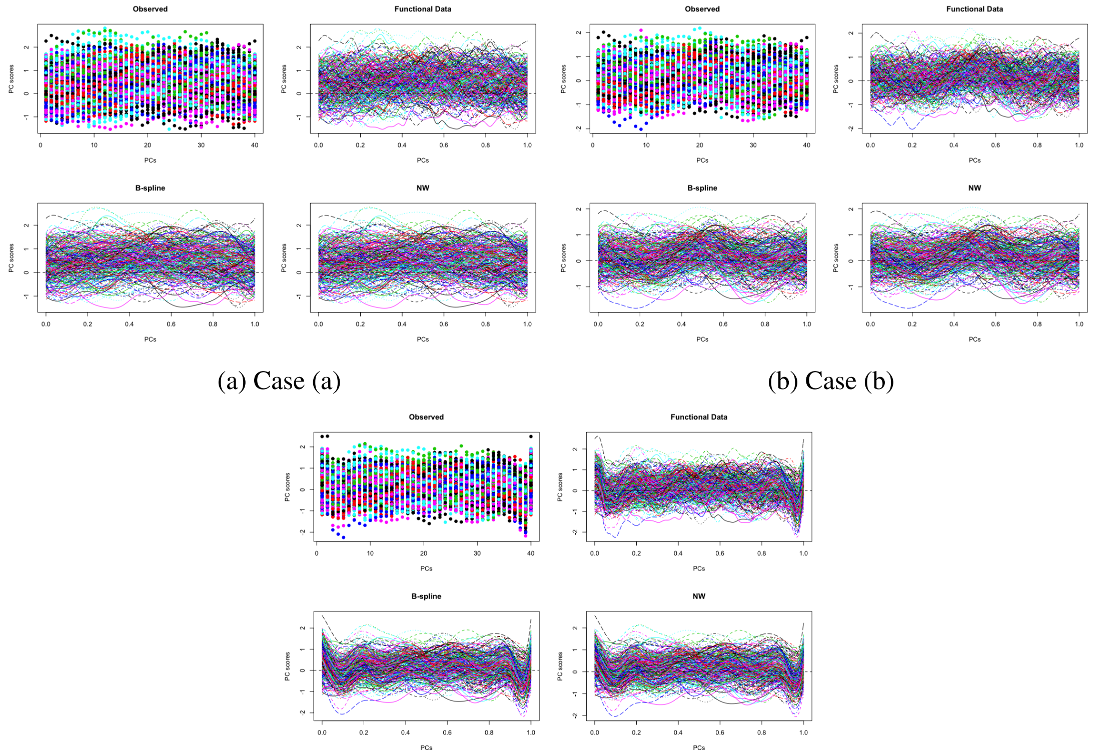
Figure 1: Three simulated data examples with means of $X_i(s)$ and $Y_i(s)$ for cases (a), (b), and (c). The gray solid curves are observations. The red curve shows the mean of $X_i(s)$, and the blue curve shows the mean of $Y_i(s)$ for each sample. Upon first glance at the observations (gray), the difference between the two samples is not clear due to the noise; however, it is clear that the means for the two samples are different.

Figure 2 shows the observed multivariate data based on generated simulated data, functional data, smoothed functional data using a B-spline basis and a non-parametric method with the Nadaraya-Watson estimator for cases (a), (b), and (c), respectively. Here, we use the `R` packages `fda` [44, 42, 47] and `fda.usc` [14] in the `R` statistical software to convert functional data from the observed multivariate data. Since the observed data have no prominent spikes or outliers, it is difficult to intuitively distinguish between functional data and smoothed functions in these simulated data. However, the number of peaks on both parametric and non-parametric smoothed data is less than functional data without smoothing.

Type	**Smoothing**	**Clustering**	(a)		(b)		(c)	
			Aligned	Unaligned	Aligned	Unaligned	Aligned	Unaligned
MDA		k-means	83.0		54.3		56.0	
		hierarchical	78.0		53.3		50.7	
		Mclust	83.0		54.3		50.0	
FDA	No Particular Smoothing	k-means	81.0	82.0	52.6	53.0	50.3	51.0
		funHDDC	84.7	63.0	57.3	51.3	62.0	50.7
		funFEM	83.3	82.3	55.0	50.3	62.0	51.0
	Parametric (b-spline)	k-means	77.0	81.3	54.3	51.3	59.0	55.3
		funHDDC	85.0	83.0	59.7	54.6	98.3	52.3
		funFEM	84.7	83.0	59.3	54.3	97.0	54.7
	Non-Parametric (NW)	k-means	82.0	82.3	52.0	51.0	52.0	50.3
		funHDDC	85.0	84.0	59.7	55.3	72.7	57.7
		funFEM	84.7	83.0	59.3	55.7	72.3	56.0

Table 2: Success rate (%) for classification. Dark gray shows the highest accuracy for the aligned dataset, and light gray shows the highest accuracy for the unaligned dataset.

We perform the clustering algorithms on these simulated data, and the classification results are shown in Table 2. For MDA, we use three clustering algorithms: the k-means, hierarchical, and *mclust* methods. The functional k-means, funFEM, and funHDDC clustering algorithms are applied to the functional



(a) Case (a)

(b) Case (b)

(c) Case (c)

Figure 2: Visualization of each cell with reduced dimensions of simulated gene expression data for cases (a), (b), and (c), respectively. Each color and curve (for functional data) shows the individual cell, where the x-axis shows the index of the principal components and the y-axis represents the principal component scores. Each panel of the subfigures shows the four observed data sets. The first panel shows the original data, which are considered discrete multivariate data. The second panel is the function data converted from the original discrete data. The third and fourth panels show the smoothed data with B-spline smoothing and kernel smoothing using the Nadaraya-Watson estimator, respectively.

data. To evaluate the consistency of the classification analysis by switching the phase components of the functions, we randomly sort the order of the components (*unaligned*). Table 2 shows the results of the classification in percentage for each method and each data. In Table 2, the success rates (%) of applying the functional clustering algorithms outperforms the results of other MDA clustering methods. In particular, the clustering results of both smoothed functions show the highest accuracy rate in each data set. This is because the smoothing methods remove some unimportant additive noise from each function.

In the comparison of the three cases (a), (b), and (c), case (c) shows the highest classification rate (98.3%) since the shape and spikes on both sides affect the differences between the samples X and Y. Case (a) shows the second-highest classification rate (85.0%) among the three data sets. This implies that the height (or y-axis or vertical difference) also plays a major role in grouping the observations into the correct group. Case (b) shows the lowest accuracy rate (59.7%) due to the similarity between the samples X and Y. As expected, the average of the classification rates for the unaligned phase components of the functions is lower than the results of the aligned functional data. Particularly, case (c) shows the large difference in the classification rate between aligned and unaligned functional data. This implies that the order of the phase components of the function is also a major factor in applying functional data analysis methods to achieve the highest accuracy rate.

3.2 Real Data

The real scRNA-seq data sets were collected from *conquer* [51] and used for our classification evaluations: GSE 52529-GPL16791 (here denoted **Trapnell**) [52], EMTAB 2805 (**Buettner**) [7], GSE 77847 (**Meyer**) [35], and GSE 81903 (**Shekhar**) [50]. These data sets are not expected to be used with the aim of detecting subpopulations of cell populations. Hence, the cluster labels known as "true" cluster labels might not represent the strongest signal present in the data [11]. In other words, general classification algorithms cannot detect the transcriptional signal or the characteristics of each cluster, which statistically derives different cluster labels rather than true cluster labels. Duó et al. [11] noted that these labels can be biased in favor of current methodologies. Therefore, our goal for this real data analysis is to detect the *true* subpopulations. Hence, it is important to validate the performance of functional clustering algorithms considering cells as a functional shape using these datasets to uncover the functional nature of scRNA-seq gene expression data.

The descriptions of each data set, including the number of cells and subpopulations, are shown in Table 3. For example, **Trapnell**, **Buettner**, **Myer**, and **Shekhar** have 3, 3, 2, and 4 subpopulations, respectively. The selected cell phenotype was used to define the "true" partition of cells when evaluating the clustering methods. The details of the subpopulations and the methods for finding these true subpopulations for each data are explained and described in [51].

Dataset	Organism	Sequencing Protocol	Cells	Methods	Subpopulations	Description	Reference
Trapnell	Homo Sapiens	SMARTer C1	288	Monocle	3	Primary myoblasts over a time course of serum-induced differentiation.	[52]
Buettner	Mus musculus	SMARTer C1	288	single-cell latent variable model (scLVM)	3	mESC in different cell cycle stages.	[7]
Meyer	Mus musculus	SMARTer C1	96	Two-way ANOVA	2	Dnmt3a loss of function in Flt3-ITD and Dnmt3a-mutant AML .	[35]
Shekhar	Mus musculus	Smart-Seq2	383	Random Forrest Classifier	4	P17 retinal cells from the Kcng4-cre;stop-YFP X Thy1-stop-YFP Line # 1 mice.	[50]

Table 3: Description of scRNA-seq data

For the pre-processing steps, such as quality control, normalization, the detection of variable genes across the single cells, and scaling, we use the *Seurat* R 2.3.4 package [51] to perform the downstream analysis. *Seurat* can also perform *t-SNE* analysis and clustering methods, such as k-means; however, we did not implement these clustering methods using *Seurat* and rather used general-purpose R packages or scripts (Table 1) for clustering the cell subpopulations. More details about these pre-processing steps and procedures using the *Seurat* Bioconductor are described in [8].

After applying the pre-processing steps on the scRNA-seq data, we used variable genes [8] to perform PCA to reduce the dimensionality of the gene expression data. In this analysis, we can visualize the distribution or pattern of the cell populations by plotting PC1 vs. PC2. Figure 3 shows the PC1 vs. PC2 plot for each scRNA-seq data set after performing PCA. In this figure, it is complicated to group a set of objects into the "true" groups as given in Table 3 without any statistical clustering analysis. One of the main reasons for these results is that the "true" cluster labels do not represent the strongest signal present in the multivariate data. For example, the **Trapnell**, **Buettner**, **Myer**, and **Shekhar** datasets have 3, 3, 2, and 4 subpopulations, respectively; however, none of the plots in Figure 3 show a clear distinctive number of clusters for each dataset. In particular, the PC plot for **Buettener** has only one large group, although there are four "true" subpopulations. In this sense, we want to see how FDA, which considers each cell as a functional shape, performs in classification against multivariate data for these phenomena whereby the signal of the "true" labels is not sufficiently strong. This functional approach will identify the hidden functional nature of scRNA-seq data.

We visualized the scree plot and performed jackstraw [9] to determine the optimal number of princi-

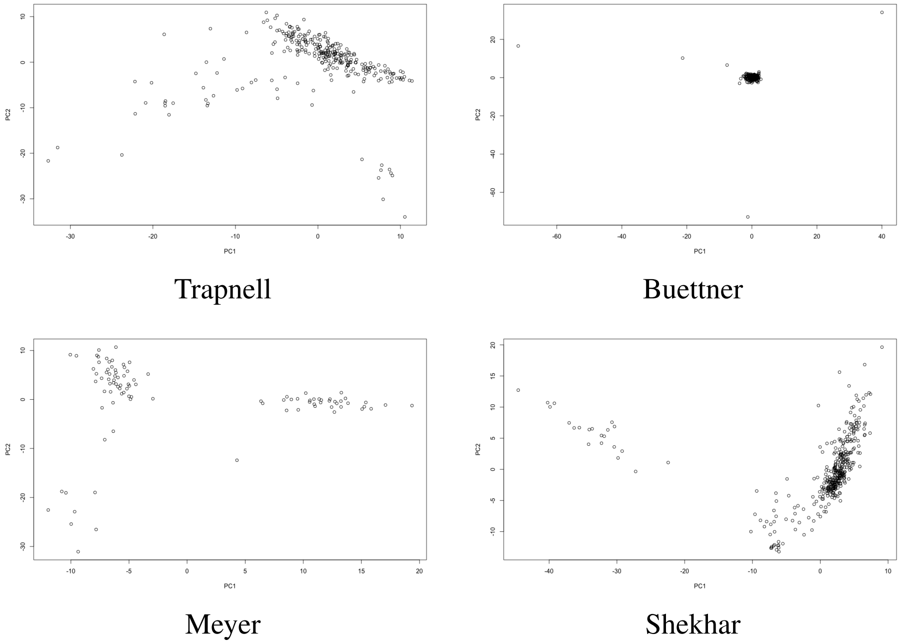


Figure 3: PC1 vs. PC2 plot for each scRNA-seq data set. All data sets could not be separated into "true" numbers of subpopulations: 3, 3, 2, and 4. In particular, **Buettener** shows only one large group, although there are three "true" subpopulations, which indicates that the conventional methods might not detect the "true" clusters.

pal components to reduce the dimensionality of the original data. A scree plot in PCA is a useful tool that visualizes saturation in the relationship between the number of principle components and the percentage of the variance explained. We generally decide the number of principal components that corresponds to the "elbow" part of the curve to have sufficient information of the original data. We chose PC1-20 for the first two data sets and PC1-15 for the last two data sets for the downstream analysis.

One of the important features of the function in the Hilbert space, $\mathcal{H}$, is that unlike the vectors in MDA, it does not allow the permutations of the components, i.e., the phase components. Based on this characteristic of the function, we performed two analyses depending on the sorting criterion: eigenvalues and genetic spatial information.

Sort by eigenvalues We first perform and build the function where phase components are sorted according to the eigenvalues. It is the simplest way that we directly use principal components from PCA results. For example, we use PC1 to PC 20 for **Trapnell** and **Buettner** and PC1 to PC15 for **Meyer** and **Shekhar** in descending order of eigenvalues. Then, the scRNA-seq gene expression data are reconstructed after PCA, where the x-axis represents the principal components, which have 20 grid points for the first two data sets and 15 grid points for the last two data sets, while the y-axis represents the PC scores for each cell.

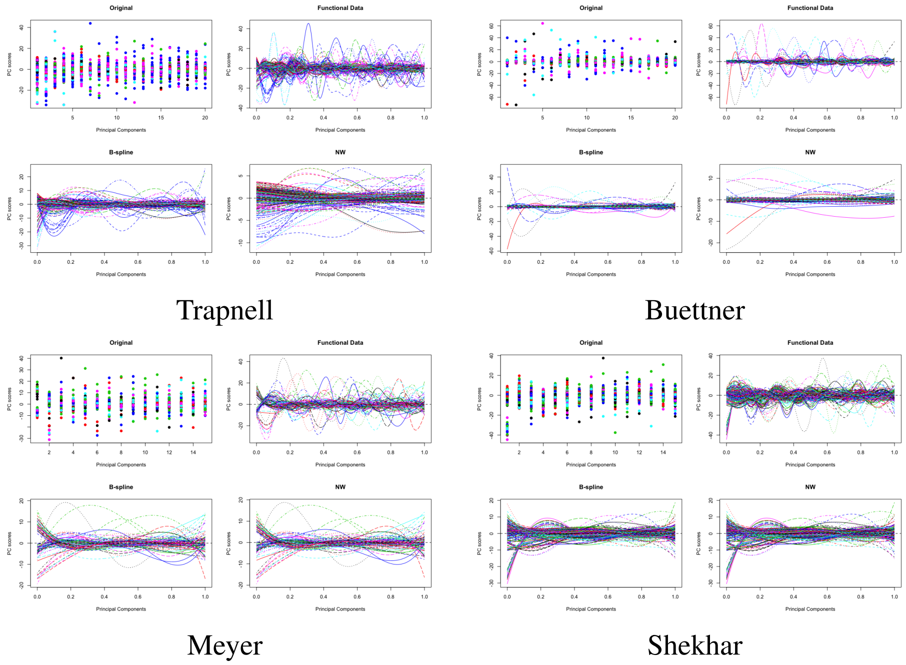
Sort by genetic spatial information Genetic spatial information is also an important factor to consider in scRNA-seq analysis. After PCA, we find the first top variable gene for each principal component, and then, we embed these genes into the components. Then, we extract the gene information using `biomaRt` [13, 12] from Bioconductor. Each ensembl gene ID has various types of information such as ensembl transcript ID, chromosome name, start and end positions of the molecule, and external gene name. Based on this information, we extract the chromosome name, start position, and end position to sort the principal components. Most chromosome names are given as numbers (1 to 21); however, some are denoted as characters such as X and Y for males and females, respectively. Hence, we assign X as 22 (for human, 20 for mouse) and Y as 23 (for human, 21 for mouse). We also assign mitochondria (MT) as 24 (or 22 for mouse). Then, we can align the gene names (embedded in the principal components) according to the chromosome numbers and molecular positions. Table 4 shows the top variable genes for each principal component for each data set. For example, **Trapnell** has the *ENSG00000149925 ensembl* gene ID for PC1. By using `biomaRt`, this refers to *ALDOA* as the HUGO Gene Nomenclature Committee (hgnc) ID with 16 chromosome names, with 30,064,164 start positions and 30,070,457 end positions. Then, we can calculate the average molecular position and sort by the genetic spatial information according to the chromosome name and average position.

Data	# of PCs	Top Variable Gene for each PC
Trapnell	20	ENSG00000149925, ENSG00000270629, ENSG00000113356, ENSG00000104980, ENSG00000126432, ENSG00000281852, ENSG00000136874, ENSG00000121578, ENSG00000108561, ENSG00000149761, ENSG00000135372, ENSG00000101470, ENSG00000196683, ENSG00000138675, ENSG00000138134, ENSG00000092054, ENSG00000226248, ENSG00000160799, ENSG00000173436, ENSG00000103197
Buettner	20	ENSMUSG00000069083, ENSMUSG00000062997, ENSMUSG00000029580, ENSMUSG00000028837, ENSMUSG00000065990, ENSMUSG00000101111, ENSMUSG00000101249, ENSMUSG00000020608, ENSMUSG00000013236, ENSMUSG00000015290, ENSMUSG00000035506, ENSMUSG00000019942, ENSMUSG00000050107, ENSMUSG00000064356, ENSMUSG00000047675, ENSMUSG00000032415, ENSMUSG00000031292, ENSMUSG00000027007, ENSMUSG00000035673, ENSMUSG00000002489
Meyer	15	ENSMUSG00000073421, ENSMUSG00000051748, ENSMUSG00000096967, ENSMUSG00000064356, ENSMUSG00000062590, ENSMUSG00000071650, ENSMUSG00000030107, ENSMUSG00000029802, ENSMUSG00000002076, ENSMUSG00000070501, ENSMUSG00000029196, ENSMUSG00000035498, ENSMUSG00000097164, ENSMUSG00000060441, ENSMUSG00000026944
Shekhar	15	ENSMUSG00000024985, ENSMUSG00000027674, ENSMUSG00000006007, ENSMUSG00000021803, ENSMUSG00000024857, ENSMUSG00000029309, ENSMUSG00000028519, ENSMUSG00000028172, ENSMUSG00000022820, ENSMUSG00000064368, ENSMUSG00000022836, ENSMUSG00000021036, ENSMUSG00000026740, ENSMUSG00000090733, ENSMUSG00000005150

Table 4: Top variable gene for each principal component for each data set

After sorting the PCs based on these two criteria, we build the functional data using `fda` and `fda.usc` in the `R` software to convert the functional data from the original data. Then, we apply two functional smoothing methods, parametric and non-parametric, to smooth the functional data. The results of these processes are shown in Figure 4 and Figure 5. In particular, the functional data, in which the phase components are sorted by eigenvalues and by genetic spatial information, are shown in Figure 4 and Figure 5, respectively. In each figure, the first panel shows the original multivariate data after PCA. The second panel of the figure shows the functional data, and each cell is fitted using a B-spline basis. We set a sufficient number of bases to construct the functional data from the scRNA-seq data. The third and last panels display the functional data after smoothing. The first panel is using a B-spline basis with a Generalized Cross-Validation (GCV) criterion, and the other panel is a kernel smoothing [15] using a Nadaraya-Watson [37] estimator with GCV. Unlike the simulated data, which are already smooth and sufficiently simple since we use a Gaussian process, the functions from the real scRNA-seq data are messy and noisy. Therefore, the functions are smoothed after applying functional smoothing techniques. All the smoothed data using parametric and non-parametric methods show very smooth curves compared to the original functional data in Figure 4 and Figure 5. Using these results of the original and smoothed data, we applied several classification methods to classify the clusters for the scRNA-seq data.

We performed clustering analysis on real data, and the classification results and accuracy rates are shown in Table 5. In these real data experiments, we utilized three different cases of phase alignment



Trapnell

Buettner

Meyer

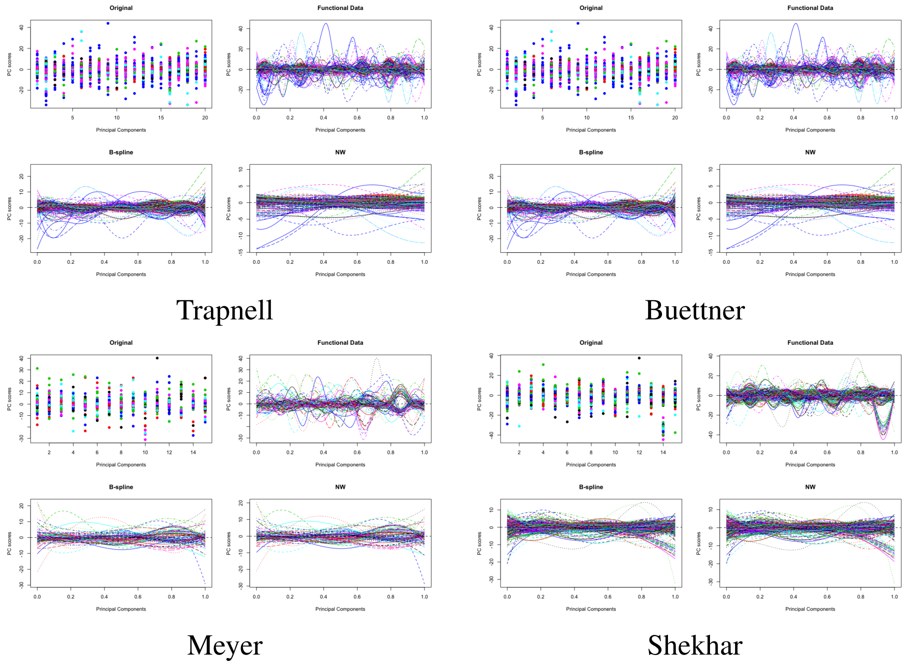
Shekhar

Figure 4: Visualization of multivariate and functional scRNA-seq data with smoothing (sorted by eigenvalue). Each subfigure shows the real scRNA-seq data after reducing the dimensionality of the gene expression data. Each color and curve (for functional data) represents the individual cell, where the x-axis shows the index of the principal components and the y-axis represents the principal component scores. The first panel shows the original data, which are considered discrete multivariate data. The second panel is the function data converted from the original discrete data. The third and fourth panels show the smoothed data with B-spline smoothing and kernel smoothing using the Nadaraya-Watson estimator, respectively.

for each data set: 1) randomly aligned (*Random*), 2) sorted by eigenvalue (*PC*), and 3) sorted by genetic spatial information (*spatial*). In this table, the shaded box represents the highest accuracy rate for each scRNA-seq data set. All of the data sets show higher accuracy rates for functional clustering algorithms. Moreover, smoothing methods enhance the classification efficiency from the results for the **Trapnell**, **Meyer**, and **Shekhar** data sets. This table also shows that randomly sorted functional data sets have lower accuracy rates compared to those sorted by our proposed criteria. **Trapnell** and **Meyer** show higher accuracy rates for *PC*, and **Buettner** and **Shekhar** perform the robust classification and show higher classification rates for *Spatial*. The classification results of randomly aligned (*Random*) phase components are similar to the results of multivariate data clustering algorithms. In particular, the accuracy rate is very similar to the MDA clustering classification rate for the **Meyer** data. In general, the functional clustering approach with two methods of aligning the phase components on the scRNA-seq data outperforms the multivariate clustering approach from these real-data experiments results.

4 Discussion

In this study, we have proposed a new framework, functional data analysis for scRNA-seq data, to identify the subpopulations of cell populations. We have demonstrated that functional data analysis can be applied to scRNA-seq data to improve the accuracy rate of classification to identify and characterize cell populations. This is another method of analyzing scRNA-seq data considering cells as a functional shape



Trapnell

Buettner

Meyer

Shekhar

Figure 5: Visualization of multivariate and functional scRNA-seq data with smoothing (sorted by genetic spatial information). Each subfigure shows the real scRNA-seq data after reducing the dimensionality of the gene expression data. Each color and curve (for functional data) represents the individual cell, where the x-axis shows the index of the principal components and the y-axis represents the principal component scores. The first panel shows the original data, which are considered discrete multivariate data. The second panel is the function data converted from the original discrete data. The third and fourth panels show the smoothed data with B-spline smoothing and kernel smoothing using the Nadaraya-Watson estimator, respectively.

Type	**Smoothing**	**Clustering**	Trapnell			Buettner			Meyer			Shekhar		
			Random	PC	Spatial	Random	PC	Spatial	Random	PC	Spatial	Random	PC	Spatial
MDA		k-means	39.58			45.83			57.29			30.81		
		hierarchical	36.11			33.68			51.04			25.59		
		Mclust	40.28			53.13			53.13			31.07		
FDA	No Smoothing	funHDDC	52.43	53.13	52.08	48.96	61.46	64.58	56.25	52.08	57.29	30.29	28.98	30.81
		k-means	34.38	35.76	38.89	33.68	33.68	33.68	54.17	59.38	51.04	29.50	26.89	29.42
		funFEM	39.58	52.08	49.65	45.49	47.22	46.88	58.33	58.33	60.42	30.81	28.98	30.81
	Parametric (b-spline)	funHDDC	43.06	53.47	48.61	40.28	35.42	46.88	50.00	62.50	60.42	27.94	34.99	29.50
		k-means	36.81	37.50	52.43	44.10	36.46	51.04	51.04	50.00	51.04	33.16	31.59	32.64
		funFEM	39.93	50.35	42.36	44.10	36.46	51.04	50.00	66.67	55.21	27.94	31.85	30.29
	Non-parametric (N.W.)	funHDDC	36.11	35.76	42.36	41.67	34.38	44.10	51.04	63.54	60.42	28.20	34.79	31.07
		k-means	43.40	36.81	50.00	41.67	41.67	45.83	50.00	52.08	52.08	29.77	36.03	37.08
		funFEM	46.53	39.93	44.44	44.79	40.97	48.61	50.00	51.04	54.17	30.03	31.85	28.46

Table 5: Classification results (%) for real data. Each gray box shows the highest classification rate for each data set.

rather than discrete vectors. This framework improves the classification rates in scRNA-seq analysis, in particular, when the biological data may not represent the strongest signal present in the data. This was one of the major problems in multivariate data analysis since any evaluation is based on typical inference by clustering the cells using MDA clustering algorithms, and these clustering label risks are biased. In

MDA, most bioinformatics techniques and methods are focused on examining the discrete genomic units of genes, and this approach might ignore spatial information and eventually loses important information such as the functional nature of the gene expression dynamics. In this sense, an approach based on FDA also plays a major role in scRNA-seq analysis.

We noted that there are two main strategies in this paper for building functional data from scRNA-seq data. One strategy is considering how to handle the number of spikes that are considered dropouts. scRNA-seq analysis allows us to reveal rare and complex cell populations and uncover regulatory relationships between genes. However, the computational analyses are more complicated due to the high variability, low capture efficiency and high rates of the zero-inflated values of the scRNA-seq assays. To solve this problem, we first perform PCA to condense the gene expression data for formatting for functional data. In this way, we can not only reduce the dimensionality of the data but also remove the dropouts such that the function can be easily fitted to the data. The other problem of building functional data is that the phase components of the function do not allow permutations, which would affect the analysis. To obtain the best classification results, we investigated the order of the phase components of the function. We give two methods of handling the order of the phase components: using the PCs from PCA, which are sorted by eigenvalues, and sorting by genetic spatial information since each gene is located in different chromosomes and at different molecular positions. We approach the clustering analysis using this spatial information to increase the classification accuracy.

There is still the shortcoming of applying functional data analysis in that, due to the removal of noise several times, some of the important facts may end up ignored in the analysis. For example, we first normalize and scale the gene expression data to reduce biological errors, such as batch effects; then, we perform PCA to remove the dropouts of the data to fit the function. Then, we fit the function on discretized gene expression data using a known basis. We also apply smoothing techniques to smooth the functional data. When implementing these several steps, we might remove crucial information about the gene expression data.

Although there are disadvantages in applying functional data analysis, this new statistical technique enhances the classification performance and ultimately improves the understanding of stochastic biological processes. Therefore, this new framework should not be considered as a replacement for conventional MDA methods. However, it can be truly effective when current MDA methods cannot detect or uncover the hidden nature of the gene expression dynamics due to weak signals. Moreover, this study enables the conversion of functional data from gene expression data, and any further functional statistical analysis is applicable to scRNA-seq analysis. This is a critical step for scRNA-seq analysis as well as functional data analysis.

Acknowledgements

We would like to thank the members of the Fujiwara laboratory for valuable discussion.

Funding

This work was supported by RIKEN intramural funding and RIKEN Single Cell Project.

Conflict of Interest: none declared.

References

[1] M. R. Anderberg. *Cluster analysis for applications: probability and mathematical statistics: a series of monographs and textbooks*, volume 19. Academic press, 2014.

[2] T. S. Andrews and M. Hemberg. Identifying cell populations with scrnaseq. *Molecular aspects of medicine*, 59:114–122, 2018.

[3] Z. Bar-Joseph, G. K. Gerber, D. K. Gifford, T. S. Jaakkola, and I. Simon. Continuous representations of time-series gene expression data. *Journal of Computational Biology*, 10(3-4):341–356, 2003.

[4] R. Becker. *The new S language*. CRC Press, 2018.

[5] C. Bouveyron, E. Côme, J. Jacques, et al. The discriminative functional mixture model for a comparative analysis of bike sharing systems. *The Annals of Applied Statistics*, 9(4):1726–1760, 2015.

[6] C. Bouveyron and J. Jacques. Model-based clustering of time series in group-specific functional subspaces. *Advances in Data Analysis and Classification*, 5(4):281–300, 2011.

[7] F. Buettner, K. N. Natarajan, F. P. Casale, V. Proserpio, A. Scialdone, F. J. Theis, S. A. Teichmann, J. C. Marioni, and O. Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology*, 33(2):155, 2015.

[8] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411, 2018.

[9] N. C. Chung and J. D. Storey. Statistical significance of variables driving systematic variation in high-dimensional data. *Bioinformatics*, 31(4):545–554, 2014.

[10] N. Coffey, J. Hinde, and E. Holian. Clustering longitudinal profiles using p-splines and mixed effects models applied to time-course gene expression data. *Computational Statistics & Data Analysis*, 71:14–29, 2014.

[11] A. Duò, M. D. Robinson, and C. Soneson. A systematic performance evaluation of clustering methods for single-cell rna-seq data. *F1000Research*, 7, 2018.

[12] S. Durinck, Y. Moreau, A. Kasprzyk, S. Davis, B. De Moor, A. Brazma, and W. Huber. Biomart and bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*, 21(16):3439–3440, 2005.

[13] S. Durinck, P. T. Spellman, E. Birney, and W. Huber. Mapping identifiers for the integration of genomic datasets with the r/bioconductor package biomart. *Nature protocols*, 4(8):1184, 2009.

[14] M. Febrero-Bande and M. Oviedo de la Fuente. Statistical computing in functional data analysis: the r package fda. usc. *Journal of Statistical Software*, 51(4):1–28, 2012.

[15] F. Ferraty and P. Vieu. *Nonparametric functional data analysis: theory and practice*. Springer Science & Business Media, 2006.

[16] E. Forgey. Cluster analysis of multivariate data: Efficiency vs. interpretability of classification. *Biometrics*, 21(3):768–769, 1965.

[17] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631, 2002.

[18] C. Fraley and A. E. Raftery. Bayesian regularization for normal mixture estimation and model-based clustering. *Journal of classification*, 24(2):155–181, 2007.

[19] C. Fraley, A. E. Raftery, T. B. Murphy, and L. Scrucca. mclust version 4 for r: normal mixture modeling for model-based clustering, classification, and density estimation. Technical report, Technical report, 2012.

[20] A. Ghiglietti, F. Ieva, and A. M. Paganoni. Statistical inference for stochastic processes: two-sample hypothesis tests. *Journal of Statistical Planning and Inference*, 180:49–68, 2017.

[21] A. Ghiglietti and A. M. Paganoni. Statistical inference for functional data based on a generalization of mahalanobis distance. *Mox Report 39/2014, Department of Mathematics, Politecnico di Milano*, 6, 2014.

[22] A. D. Gordon. A review of hierarchical classification. *Journal of the Royal Statistical Society: Series A (General)*, 150(2):119–137, 1987.

[23] W. Härdle. *Applied nonparametric regression*. Number 19. Cambridge university press, 1990.

[24] J. A. Hartigan. Clustering algorithms. 1975.

[25] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[26] B. Hwang, J. H. Lee, and D. Bang. Single-cell rna sequencing technologies and bioinformatics pipelines. *Experimental & molecular medicine*, 50(8):96, 2018.

[27] V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, A. R. Green, et al. Sc3: consensus clustering of single-cell rna-seq data. *Nature methods*, 14(5):483, 2017.

[28] R. R. Klevecz and D. B. Murray. Genome wide oscillations in expression–wavelet analysis of time series data from yeast expression arrays uncovers the dynamic architecture of phenotype. *Molecular Biology Reports*, 28(2):73–82, 2001.

[29] X. Leng and H-G Müller. Classification using functional data analysis for temporal gene expression data. *Bioinformatics*, 22(1):68–76, 2005.

[30] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[31] Y. Luan and H. Li. Clustering of time-course gene expression data using a mixed-effects model with b-splines. *Bioinformatics*, 19(4):474–482, 2003.

[32] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[33] A. Martino. Classification algorithms for multivariate functional data. 2016.

[34] V. 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.

[35] S. E. Meyer, T. Qin, D. E. Muench, K. Masuda, M. Venkatasubramanian, E. Orr, L. Suarez, S. D. Gore, R. Delwel, E. Paietta, et al. Dnmt3a haploinsufficiency transforms flt3itd myeloproliferative disease into a rapid, spontaneous, and fully penetrant acute myeloid leukemia. *Cancer discovery*, 6(5):501–515, 2016.

[36] F. Murtagh and P. Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 31(3):274–295, 2014.

[37] E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

[38] R. Opgen-Rhein and K. Strimmer. Learning causal networks from systems biology time course data: an effective model selection procedure for the vector autoregressive process. *BMC bioinformatics*, 8(2):S3, 2007.

[39] X. Qiu, A. Hill, J. Packer, D. Lin, Y-A Ma, and C. Trapnell. Single-cell mrna quantification and differential analysis with census. *Nature methods*, 14(3):309, 2017.

[40] X. Qiu, Q. Mao, Y. Tang, L. Wang, R. Chawla, H. A. Pliner, and C. Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nature methods*, 14(10):979, 2017.

[41] J. O. Ramsay. When the data are functions. *Psychometrika*, 47(4):379–396, 1982.

[42] J. O. Ramsay. Functional data analysis. *Encyclopedia of Statistics in Behavioral Science*, 2005.

[43] J. O. Ramsay and C. J. Dalzell. Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):539–572, 1991.

[44] J. O. Ramsay, G. Hooker, and S. Graves. Functional data analysis with r and matlab, vol. 66, 2010.

[45] J. O. Ramsay and B. W. Silverman. *Applied functional data analysis: methods and case studies*, volume 77. Citeseer, 2002.

[46] J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, 2nd edition, 2005.

[47] J. O. Ramsay and B. W. Silverman. *Applied functional data analysis: methods and case studies*. Springer, 2007.

[48] J. A. Rice and B. W. Silverman. Estimating the mean and covariance structure nonparametrically when the data are curves. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(1):233–243, 1991.

[49] L. Scrucca, M. Fop, T. B. Murphy, and A. E. Raftery. mclust 5: clustering, classification and density estimation using gaussian finite mixture models. *The R journal*, 8(1):289, 2016.

[50] K. Shekhar, S. W. Lapan, I. E. Whitney, N. M. Tran, E. Z. Macosko, M. Kowalczyk, X. Adiconis, J. Z. Levin, J. Nemesh, M. Goldman, et al. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell*, 166(5):1308–1323, 2016.

[51] C. Soneson and M. D. Robinson. Bias, robustness and scalability in single-cell differential expression analysis. *Nature methods*, 15(4):255, 2018.

[52] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology*, 32(4):381, 2014.

[53] S. Wesolowski, D. Vera, and W. Wu. Srsf shape analysis for sequencing data reveal new differentiating patterns. *Computational biology and chemistry*, 70:56–64, 2017.

[54] P-S Wu and H-G Müller. Functional embedding for the classification of gene expression profiles. *Bioinformatics*, 26(4):509–517, 2010.

[55] L. Zappia, B. Phipson, and A. Oshlack. Exploring the single-cell rna-seq analysis landscape with the scrna-tools database. *PLoS computational biology*, 14(6):e1006245, 2018.