Robust methods for inferring cluster structure in Single Cell RNA Sequencing data

by

Elijah Willie

B.Sc, Simon Fraser University, 2018

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Bioinformatics)

The University of British Columbia
(Vancouver)

October 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Robust methods for inferring cluster structure in Single Cell RNA Sequencing data

submitted by **Elijah Willie** in partial fulfillment of the requirements for the degree of **Master of Science** in **Bioinformatics**.

Examining Committee:

Sara Mostafavi, Statistics and Medical Genetics
Supervisor

Ryan Brinkman, Medicine and Medical Genetics
Supervisory Committee Member

Francis Lynn, Medicine and Biomedical Engineering
Supervisory Committee Member

Abstract

Single Cell RNA sequencing (SCRNA-SEQ) enables researchers to gain insights into complex biological systems not possible with previous technologies. Unsupervised machine learning, and in particular clustering algorithms, are critical to the analysis of scRNA-seq datasets, enabling investigators to systematically define cell types based on similarities in global gene expression profiles. However, in robustly applying a clustering algorithm to identify and define cell types, two critical open questions remain: i) to what extent do natural clusters exist in a given dataset? ii) what is the number of clusters best supported by the data? More specifically, most clustering algorithms will attempt to identify a fixed number of clusters without considering whether a given dataset is clusterable (e.g., natural clusters exist). However, understanding when the application of clustering algorithms is appropriate is crucial in making inferences from scRNA-seq datasets. Further, all clustering algorithms require the user to explicitly or implicitly specify the number of clusters to search for. In this thesis, we first assess the robustness of multimodality testing methods for determining whether a given set of points (or a dataset) is clusterable. Next, we utilize this framework to develop an algorithm, which we refer to as CCMT, for inferring the number of robust clusters in a given dataset. Results on simulation studies show that multimodality testing as a means for inferring cluster structure is robust and scales favorably for large datasets. This method can detect cluster structure with high statistical power in situations where there is high overlap between the clusters. We also apply our approach to real scRNA-seq datasets and show that it can accurately determine the cluster structure in both positive and negative control experiments. In the second part of this work, we apply CCMT in simulation studies and show that coupling multimodality testing with the nested

structure of hierarchical clustering and discriminant analysis provides a robust approach for determining the number of clusters in a given dataset. Results on real data also show that CCMT can recover ground truth partitions with reasonable accuracy, and it is much faster than the competing methods that have a similar accuracy range.

Lay Summary

The human body contains millions of cells, each able to perform a wide variety of tasks. It is now possible to view each of these cells individually at the molecular level. Doing so has enabled scientists to systematically define cell types based on their molecular (gene expression) properties. To identify existing cell types and potentially define new ones, scientists often use clustering algorithms. Clustering is a technique used to group similar objects based on their properties – in this context, our objects are cells, and their properties are the expression levels of genes. In this thesis, we address two questions that are critical to the success of clustering algorithms: i) to what extend "natural" clusters exist in a given dataset, ii) how many clusters can be found in a given dataset.

Preface

This dissertation is an original intellectual product of the author, Elijah Willie. The breakdown of contributions reported in the present manuscript is as follows: The implementation of all statistical methodologies as described in all of Chapter 3 has been performed by Elijah Willie. Results discussions and exploration presented in Chapters 4 and 5 has been performed by Elijah Willie. This manuscript has been written by Elijah Willie under the supervision of Dr. Sara Mostafavi.

Table of Contents

List of Tables

List of Figures

Glossary

ARI Adjusted Rand Index

CCMT Computing clusters through multimodality testing

CDF Cumulative distribution function

CSR Complete spatial randomness

DWH Soft least squares Euclidean consensus partitions described in Dimitriadou, Weingessel and Hornik

FWER Family wise error rate

PBMCS Peripheral Blood Mononuclear Cells

PCA Principal component analysis

PCS Principal components

PDF Probability density function

SCRNA-SEQ Single cell RNA sequencing

T-SNE t-distributed Stochastic Neighbor Embedding

UMAP Uniform Manifold Approximation and Projection for Dimension Reduction

Acknowledgments

Although the process of obtaining a Masters degree may seem like a solitary endeavor, mine certainly would not have been possible without the support of a plethora of individuals including mentors, colleagues, friends and family members. With all my heart, I would like to sincerely thank the following individuals:

- My supervisor Dr. Sara Mostafavi, for her continual support, guidance and wisdom throughout my studies.
- Dr. Bernard Ng, for his willingness to always help and provide constructive feedback whenever necessary.
- All of my lab-mates, who made graduate school a delightful experience. Particularly I would like to thank Gherman Novakovsky, for his willingness to always talk to me about doubts I had about my project.
- My beloved friends Divya Bafna and Figal Taho for being part of this journey with me and always being available to share a laugh even when it was hard to do so. You guys made graduate school an unforgettable experience, both in the lab and outside the lab.
- My family, for always believing in me, showing me unconditional love, nurturing and encouragement, and encouraging me to push myself beyond my limits. They are the rock through wish all my strength is built upon. All this would not be possible without them.
- Dr. Leond Chindelevitch, who cultivated my interest in research during my undergraduate studies, commitment to my research development and made so many opportunities possible for me.

- My Lord and Saviour, Jesus Christ, for his provision, unconditional love and saving grace, and through whom all things are possible.

Chapter 1

Introduction

The advent of single-cell RNA sequencing (SCRNA-SEQ) has enabled the investigation of complex tissues cellular composition with unprecedented resolution. Deciphering cell type composition is one of the primary applications of scRNA-seq, typically done using unsupervised clustering, a technique borrowed from the machine learning literature. The nature of scRNA-seq datasets has provided new challenges for unsupervised clustering methods. These challenges include high variability and noise levels do technical limitations and biological factors [5, 7, 36, 45, 110].

A fundamental challenge when clustering scRNA-seq datasets is whether the data contains inherent clusters to warrant clustering in the first place [2]. This issue is essential because a dataset should be clustered only when there exists an inherent cluster structure. The idea of clusterability, which seeks evidence for structure inherent to a dataset, should be a pivotal step in helping the user decide if clustering is appropriate for their dataset. Clustering should only be applied if a dataset contains inherent structure; else, the results would be misleading. As an example, consider a set of n = 1000 points generated from a unimodal distribution. There is no inherent cluster structure, and thus all clustering algorithms should return a single cluster. Any number ($k > 1$) of clusters returned would be purely nonsensical. See Figure 1.1.

Determining a suitable number of clusters is another concern when clustering scRNA-seq datasets. For most clustering algorithms, the number of clusters needs

to be known apriori and is often unknown. Domain knowledge is often used in the biological setting to determine a suitable number of clusters. For example, when dealing with scRNA-seq gene expression data, the data is clustered with multiple cluster numbers. The user then uses domain knowledge, such as the expression of known marker genes from the literature, to determine a suitable number of clusters [60].

In this thesis, we first assess the robustness of multimodality testing as a proxy for assessing clusterability in scRNA-seq datasets. Next, we utilize the multimodality framework to design an algorithm for computing the number of clusters in scRNA-seq datasets. We show that this approach is robust to challenges inherent to scRNA-seq datasets through an extensive simulation study. Finally, extensive comparisons using benchmarking datasets show that this approach compares favorably to methods currently used for analyzing scRNA-seq datasets.

1.1 Single cell RNA sequencing

In the late mid-2000s, RNA-sequencing (RNA-Seq) emerged as a novel approach that would eventually supersede the already successful gene expression microarrays. New protocols developed for this technology typically required bulk sampling to profile thousands to millions of cells. In 2009, [104] provided a novel protocol referred to as single-cell RNA-seq (scRNA-seq), which profiles individual cells. ScRNA-seq treats a cell as an individual entity and allows for comparing cells within a specific population. For a general review, refer to [40, 56]. A sufficient amount of research is currently devoted to developing novel protocols and technologies to increase profiling accuracy. It is now possible to profile thousands of cells in a single experiment [103]. This increase in the number of cells profiled is partially due to reduced sequencing costs, improved cell dissociation protocols, and library preparation.

1.1.1 Analysis methods

Most scRNA-seq methods are designed for unsupervised clustering, pseudo-time ordering, and network inference to gain biological insights. After preprocessing [84, 101] and quality control [51, 76] of the output from a sequencing machine,

typical analysis steps include:

1. **Data normalisation.**
   Data normalization techniques have become vital in dealing with fluctuations in reads obtained per cell for sequencing technologies with high throughput. Methods developed for bulk RNA-seq can be used on scRNA-seq data as well. These methods include counts per million (CPM) normalization [82], where each value in the count matrix is divided by the total count in cell and multiplied by a million. When applied to scRNA-seq data, this is referred to as transcripts per million (TPM). Other methods include the use of size factors [4, 7] and quantile normalization [71, 76].

2. **Unsupervised clustering of cells.**
   Finding clusters of similar cells, or characterization of cell-type composition is one of the essential applications of scRNA-seq. Characterization of cell-type composition is done using unsupervised clustering. Below are some of the most prominent clustering paradigms currently used in the scRNA-seq literature

   (a) **Partitioning models**
   Partitioning based clustering algorithms are some of the most widely used clustering methodologies. These methods try to partition a given dataset into K partitions such that objects in the same partition are more similar to each other than objects across other partitions. These methods include k-means, and k-medoids, which many scRNA-Seq clustering algorithms are based on. These methods include *SC3* [59] *SCUBA* [74], *SAIC* [121], *pcaReduce* [128] and *RaceID2* [37].

   (b) **Mixture models**
   If there exists significant prior knowledge about the data generation process, specifically the distributions generating the data, then mixture models provide an intuitive way to compute clusters where each distribution represents a cluster. Clustering involves assigning objects to the most likely distribution that generated the data using expectation maximization. ScRNA-Seq methods based on this paradigm include

BISCUIT [87], *Seurat* [17] and *TSCAN* [53].

(c) **Graph models**

Graph-based models rely on building a graph representing objects as nodes and then finding densely connected regions as clusters. Generating these graphs typically involves computing a similarity score between the objects and using these scores as edge weights between objects. Finding densely connected regions is then reduced to finding regions of high similarity in the graph. Both spectral and clique detection methods are used for finding these dense regions. These models are attractive because they make no distributional assumptions about the data. ScRNA-Seq methods based on this paradigm include *Seurat* [17], *SIMLR* [113], *SNN-Cliq* [127] and *SCANPY* [116].

(d) **Density models**

Like the intuition from graph-based clustering, density-based algorithms seek to find highly dense regions of objects without representing the objects in a graph. Most notable algorithms using this paradigm includes DBSCAN [27] and Density peak [99] clustering. These models are attractive in the scRNA-Seq domain due to their potential for finding rare cell populations. Methods include *Monocle* [108], *GiniClust* [54] and *sscClust* [88].

(e) **Ensemble models**

Borrowing from the machine learning literature where weaker classifiers are combined to form a more robust classifier, ensembles models have become particularly useful for clustering. The idea here is to cluster the set of objects using different methods, including features, similarity metrics, and clustering algorithms, and then combine them to form an ensemble. Ensemble models can help to combine diversity obtained from different clustering solutions. Ensembles have also been shown to outperform single models in terms of accuracy and robustness [34, 50].

(f) **Hierarchical models**

Over the years, hierarchical clustering has become one of the most

widely used clustering methods in the scRNA-Seq domain [117]. Their increased use is due to the lack of distributional assumptions about the data generating mechanism [90]. Hierarchical clustering algorithms can uncover possible hierarchies amongst cell types and represent them using a tree structure, which is appealing since cell types can exist in hierarchies. Hierarchical representation also makes interpreting clusters easier. ScRNA-Seq methods that employs this algorithms include *SC3* [59], *cellTree* [25], *CIDR* [68] and *DendroSplit* [125].

(g) **Multiview models**
Advances in data curation and clustering techniques have enabled researchers to gather information on an experiment from multiple views to increase the resolution about the phenomena under examination. Multiview clustering [13, 20, 122] provides a methodology for combining the information contained in these views into a common representation. There are a few advantages of multiview, including generating a complete knowledge of the data, reducing noise content, and generating a more robust clustering of the data. Multiview methods differ from ensemble methods in that they combine information across multiple views before clustering is done. Whereas in ensemble models, clustering is done for each view, and the results are then combined. There have not been many adaptations of these methods to the scRNA-seq domain. For example, [16] combines gene expression data and paired epigenetic data to infer cell types and gene regulatory networks. Also, in [98], a similar idea based on pattern fusion analysis is used to integrate multiple heterogeneous omics data. Even with these methods, the multiview clustering literature for the scRNA-seq domain remains sparse and provides a possible avenue for further research.

This is by no means an exhaustive list of all the clustering paradigms and methods available to the scRNA-seq domain. However, most methods developed fall somewhere within these paradigms. Other methods combine aspects of multiple paradigms. Most methods under these paradigms require that the number of clusters K is known apriori. This value is typically un-

known, and there exists some heuristics to compute it prospectively or retrospectively. For a more thorough review of these paradigms, their advantages, and disadvantages, refer to [85].

3. **Ordering of cells.**
   An alternate but equally significant aspect of the scRNA-seq analysis is pseudo-time analysis. Pseudo-time methods are used for the analysis of gene expression levels over a continuous axis. Since scRNA-seq datasets only provide a snapshot of cells at a point in time, pseudo-time analysis requires setting up a continuous axis and observing gene expression of cells. The task then becomes ordering these cells using this continuous axis. Pseudo-time analysis could lead to understanding differentiation trajectories for different cell types and understanding how different cell states vary. A few methods have been developed for this including *Monocle* [108], *TSCAN* [53], *SCUBA* [74] and *scVelo* [11]. This area remains an active area of research, and new methods are released routinely, see [19] for a review.

4. **Differential expression analysis.**
   After clustering the cells, the next step is the interpretation of clusters. Interpreting clusters is typically done by finding genes that are differentially expressed between clusters or marker genes that are expressed in specific clusters. There have been many methods developed to do this including *D3E* [22], *DEsingle* [79], *MAST* [30], and *SCDE* [57]. Some of these methods have been designed specifically for scRNA-seq data and do not have limitations faced by bulk RNA-seq methods. Other methods developed for bulk RNA-seq have been applied to scRNA-seq data. However, they may not be appropriate due to assumptions made, see [111]. Due to large numbers of cells obtained from scRNA-seq experiments, simple statistical methods such as the Mann-Whitney U test, Student's t-test, or logistic regression may not be limited by their statistical assumptions. These simple tests are being implemented in many scRNA-seq data analysis pipelines including *Seurat* [17] and *scater* [76]. For cells ordered using pseudo-time analysis, differential expression analysis is done by finding genes with significant relationships between expression and the continuous axis. Typically splines are fit, and co-

efficients are tested for significance. Most pseudo-time analysis methods described previously include methods for finding differentially expressed genes along a trajectory.

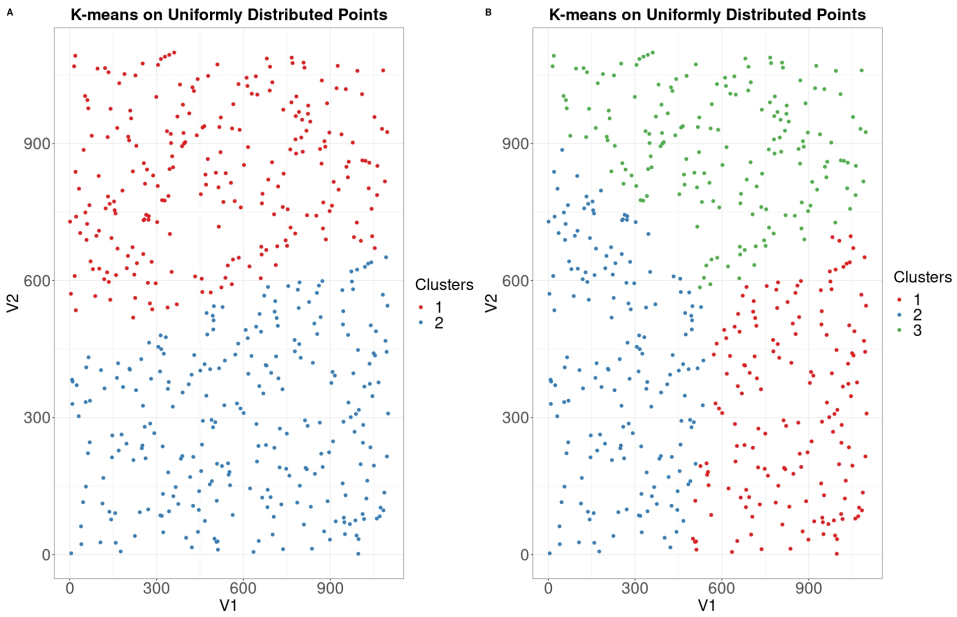

Figure 1.1: Plots of uniformly distributed points clustered by *K-means* with $k = 2$ and $k = 3$. **A)** *K-means* with $k = 2$. **B)** *K-means* with $k = 3$. Note that the cluster borders returned by *K-means* appears to be arbitrary since there is no inherent cluster structure.

1.1.2 Thesis motivation and contribution

Since clustering is a crucial aspect of most scRNA-seq analysis pipelines, great care must be taken when applying clustering algorithms. Most clustering algorithms will find clusters in a dataset, even if none exists [2]. See Figure 1.1. This is because clustering algorithms are designed to optimize an objective function that seeks to partition the dataset optimally [73]. These algorithms do not consider the possibility that there may not exist inherent clusters in the dataset. Thus, there is a risk associated with blindly applying them without proper prior analysis. In this thesis, we focused our efforts on developing a methodology to systematically

assess the level of clustering or cluster structure (Clusterability) in scRNA-seq datasets, while also estimating the possible number of clusters. To our knowledge,this question has not been addressed specifically for scRNA-Seq datasets and thus this work provides an initial framework for addressing this problem.

We perform extensive simulation studies in order to understand which problems affect clusterability analysis in scRNA-seq datasets. We consider problems such as increasing data size, complexity, sparsity, cluster size, and cluster separability. Exploring these issues in a controlled manner will illuminate the factors limiting the effectiveness of scRNA-seq clustering methods. Clusterability analysis is also done for a range of real datasets. We found that the methods developed in this thesis are robust to data sparsity, cluster size, cluster separability, and scales well with increasing data size.

We also compared methods developed in this work to other methods most often used to cluster scRNA-seq data. We used benchmarking data to evaluate these methods' running time and accuracy concretely and compared them to methods developed in this work. Our method provides a competitive running time while, on average, having a higher accuracy when clustering.

1.1.3 Detailed outline of thesis

The rest of this thesis is outlined as follows:

- Chapter 2 is dedicated to surveying the current literature on clusterability analysis and estimating the number of clusters. We discuss some of the methods most often used to assess cluster structure in a dataset. We also discuss a few ways currently used to estimate the number of clusters. These discussions are done in a broader scope without reference to the scRNA-seq domain. Finally, we provide a short discussion on how some of these methods are used for scRNA-seq analysis.

- In Chapter 3, we outline the methods developed to assess clusterability and estimate the number of clusters. First, we provide a detailed outline of the multimodality testing method for assessing clusterablity. Next, we present the CCMT procedure for estimating the number of clusters. Thirdly we outline both the simulation studies and real datasets used to evaluate both meth-

ods. Lastly, we provide details about the methods used for benchmarking and the evaluation metrics.

- In Chapter 4, we discuss the results of our methods based on simulation studies and real data. First, we provide simulation studies and real data results for assessing clusterability. Next, we provide the results on simulation studies and real data for the CCMT procedure. Finally, we provide the results for the comparative analysis for the CCMT procedure against other methods.

- In Chapter 5, general conclusions are provided about the finding of this work. Also discussed are the possible shortcomings of the proposed methods and the provision of future directions that may improve these methods.

Chapter 2

Related Works

The clustering task consists of partitioning a set of objects into k groups (possibly overlapping groups) such that members of the same groups are sufficiently similar to each other and sufficiently dissimilar to non-members. Defining similarity between members is highly dependent on the question asked, the phenomena studied, and the clustering algorithm used. This creates an inherent level of subjectivity when clustering. In any clustering task, the user needs to make some assumptions about the data being clustered, with the most implicit and necessary assumption that the data indeed contains meaningful clusters. Based on the clustering algorithm chosen, further assumptions need to be made about the data, and the efficacy of the results depends on how strongly these assumptions hold. The user also needs to decide how many clusters to compute. This is a problem because the user rarely knows beforehand how many clusters to expect, and clustering results may heavily depend on the number of clusters chosen.

In this chapter, we survey the current literature on clusterability and determining a suitable number of clusters. First, we discuss the clusterability problem and the various methods for addressing it. Second, we discuss the problem of estimating the number of clusters and the current methods for addressing it.

2.1 The clusterability problem

"Clusterability" seeks to provide a measure of the level of cluster structure in a dataset. Clusterability methods assess the potential for a dataset to form clusters without making any assumptions about the data's nature. If clustering seeks to find meaningful partitions in a dataset, then clusterability seeks to find the extent to which these partitions exist. Ideally, a clustering solution is meaningful if it captures the natural cluster structure in the data.

2.1.1 Visual assessment of clusterability

The first and most intuitive way of assessing clusterability is by visual inspection [70]. Cluster structure is assessed visually by projecting the points in a dataset to a lower dimension space (typically 2 or 3 dimensions) using linear or non-linear dimensionality reduction methods. The projected points are then visualized using a $2D$ or $3D$ plot, and the cluster structure is assessed by identifying the grouping structure in the plot. Linear dimensionality reduction methods such as Principal Component Analysis (PCA) [55] assumes that the data lies on a linear plane. Non-linear dimensionality reduction methods such as t-distributed stochastic neighbor embedding (T-SNE) [112] and Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) [77] do not make this assumption about the data and uses heuristics for projecting the points on to a non-linear plane. After projection, grouping structure is identified by human eyes by seeking regions of high density separated by regions of low density. This creates a level of subjectivity that may differ between users.

There are more standard methods of assessing cluster structure visually [12, 49, 115]. These methods first compute pairwise euclidean distances between points and orders them such that any potential cluster structure in the data becomes obvious. A heat-map of the ordered dissimilarities is plotted, and a diagonal block structure provides evidence for the existence of clusters. There is another method [115] similar in flavor that uses image processing techniques to automatically count the number of diagonal blocks. However, this method has high computational complexity and becomes impractical for datasets containing many points common to scRNA-seq. Most, if not all, scRNA-seq analysis pipelines contain methods to vi-

sualize data by reducing the data on to a linear or non-linear subspace. Some established pipelines use already developed methods such as PCA [55], T-SNE [112] and UMAP [77]. Other pipelines have developed visualization methods specific to the analysis of scRNA-seq. These include methods such as ZIFA [86], ZINB-WaVE [89], scvis [24], and many more. Methods such as T-SNE and UMAP are stochastic and typically require parameter tuning by the user. Depending on the parameterization of these methods, it is possible to obtain completely different results for the same dataset, which can become problematic when determining cluster structure. This signifies the need for new methods that are deterministic and are not highly impacted by parameters.

2.1.2 Spatial randomness

Another method for assessing clusterability is through testing for complete spatial randomness (CSR). This method is one of the first and maybe the oldest methods for testing clusterability in a dataset [23, 65]. First, define a point process as a set of mathematically defined points located in an underlying space, such as the real line or the Cartesian plane. If we consider a data matrix D of n points in p dimensions as a point process, the notion of CSR can be applied. More formally, CSR methods perform a statistical test on the matrix D and draw one of three conclusions [52]:

1. The points in D are arranged randomly, meaning there is no evidence of cluster structure.

2. The points are aggregated or clustered, meaning that there is evidence for cluster structure.

3. The points are regularly spaced.

If applied to scRNA-seq, we can let D be the gene expression counts matrix, and then we can apply these types of tests and make conclusions based on the results. One of the most prominent tests for spatial randomness is the Hopkins test [47, 64]. This method tests spatial randomness by comparing a set of sampled points from D and their nearest neighbors to distances of points sampled from a null model. If D exhibits complete spatial randomness, these distances should be similar on average.

Simulation studies using this test have shown low power when the putative clusters are not well-separated [2].

Other tests for CSR includes:

1. **Scan tests** which are based on the number of points in the densest sub-region of a predefined sampling window. A large count provides evidence for the presence of cluster structure.

2. **Quadrat** analysis partitions a predefined sampling window into equally sized rectangles, and the number of points in each quadrat is counted. These counts follow a Poisson model under the assumption of CSR. A chi-square test is used for hypothesis testing.

3. **Inter-point distances** which reflect structural relationships among points. A test for randomness compares these distances with that computed from a null model.

4. **Structural graphs methods**, which defines a graph over the pairwise distances between points. A test for spatial randomness compares the distribution of this graph's edge lengths with that of a null model.

The listed methods above are described in [52]. Most of these methods suffer from expensive computations, impracticality in high dimensions, and a lack of suitable null models to be used extensively. To our knowledge, there are currently no tests of spatial randomness applied to scRNA-seq datasets.

2.1.3 Multimodality testing

Multimodality testing is another way to assess clusterability [31]. Multimodality testing tests the existence of multiple modes for a probability distribution over a set of points. Some of the well-known methods in this field are described below. However, before explaining further, a few definitions are provided.

Definition A probability density function (PDF) f is unimodal if for some value k, f is monotonically increasing for $x \leq k$ monotonically decreasing for $x \geq k$. f is multimodal if no such k exists.

Definition A cumulative distribution function (CDF) F is unimodal if a k exists such that F is convex on the interval $(-\infty, k]$ and concave on the interval $[k, \infty)$. F is multimodal if no such k exists.

Definition The modes of a multimodal probability density function are the values of x where *f(x)* it's local maximum. This maximum may be located at a single point signifying a mode or a closed interval signifying a modal interval.

See Figure 2.1 for an example of a unimodal PDF and CDF.

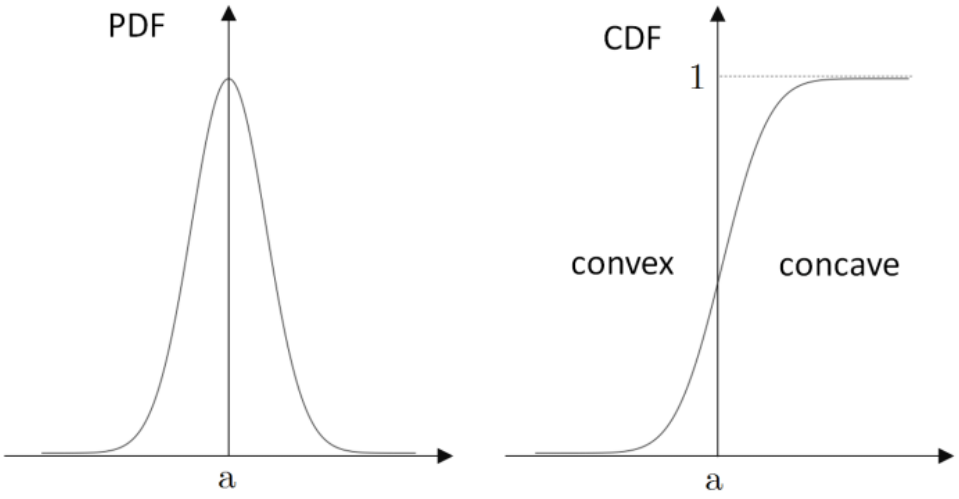

Figure 2.1: Plots of a unimodal PDF and CDF. Note that for both the PDF and CDF, the mode is located a. Figure adopted from [29].

Modality tests formally test a distribution generated from a set of points for multiple modes or multimodality. The relation can be stated as follows:

1. If a dataset D contains multiple clusters, then points in the same cluster will be closer to each other than in other clusters.

2. The distribution function of (dis)similarity between the points can be statistically tested for multimodality.

3. A significant test provides evidence for multimodality, meaning clusters are present, hence evidence for clusterability.

Below we state two of the most frequently used multimodality testing methods, namely the Dip test [42], and the Silverman test [100]. There are other tests [3, 43, 93] not described here due to their low statistical power, computational inefficiency, and lack of suitable implementations.

The Dip test

Consider a dataset $X = (x_1, ..., x_n)$, where for simplicity we assume all x_i are independent and identically distributed(i.i.d). Let f be a distribution function of (dis)similarities between a set of points, the Dip test compares these two hypothesis:

H_0: f has 1 mode vs. H_1: f has > 1 mode(s).

Now, define the Dip statistic D of a cumulative distribution function (CDF) to be:

$$D(F) = \min_{G \in U} \sup_{x} |F(x) - G(x)| \tag{2.1}$$

Where U is the class of all unimodal distributions. This value is computed using the empirical cumulative distribution (ECDF) [42]. Put another way, the Dip statistic is the maximum difference between the empirical distribution function and the unimodal distribution function that minimizes this difference. The Dip essentially measures a function's departure from unimodality. To obtain a p-value, the Dip statistic is computed for the ECDF, and this value is compared with Dip values for b samples from a uniform null distribution. Formally,

$$p - value = \frac{\sum_{b=1}^{B} \{Dip_b > Dip_X\}}{B} \tag{2.2}$$

where Dip_b is the Dip statistic for the b^{th} sample from $U(0,1)$, and Dip_X is the Dip statistic for the data. This p-value can also be computed by interpolation from a table containing empirical percentage points of the Dip statistic based on $N = 1000001$ samples of size n from $U[0,1]$.

The Silverman test

Again Consider a dataset $X = (x_1, ..., x_n)$, where for simplicity we assume all x_i are independent and identically distributed(i.i.d). The Silverman test [100] compares

these two hypothesis: H_0: f has 1 mode vs. H_1: f has > 1 mode(s).

This test employs a kernel density estimation of the form:

$$f_X(h) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{2.3}$$

where $K(\cdot)$ is the Gaussian kernel and h is the bandwidth. The Gaussian kernel is used because the number of modes k of $f_X(h)$ is a non-increasing function of h. First Let $f_X(h)$ be the kernel density estimate for X as a function of h. Next, Let h_{crit} be the maximum value of h such that $f_X(h)$ has at most k modes. The value of h_{crit} is used as a test statistic with large values indicating evidence against unimodality and small values indicating evidence for unimodality.

The test works as below:

1. Generate B samples $Z^b = \{Z_1^b, \cdots, Z_n^b\}$ with $b = 1, \cdots, B$, where $Z_i^b = (1 + \frac{h_k^2}{\sigma^2})^{\frac{-1}{2}} X_i^b)$, where σ^2 is the sample variance and X_i^b is generated from $f(h_{crit})$
2. For each sample Z^b, compute h_{crit}^b
3. Finally, to get a p-value, we compute:

$$p - value = \frac{\sum_{b=1}^{B} 1(h_{crit}^b \leq h_{crit})}{B} \tag{2.4}$$

See [3, 100] for a careful treatment of the methodology and algorithm.

2.2 Estimating the number of clusters C

There are several clustering algorithms available for use [119]. However, most of these algorithms require that the number of clusters c is specified. While it is relatively easy to point out the cluster grouping structure using $2D$ or $3D$ plots, it is more challenging to determine exactly how many groups are present. Therefore it is necessary to have robust ways of estimating the number of clusters to obtain good results. Below we discuss a few ways typically used to estimate the number of clusters when clustering is applied.

2.2.1 Conventional methods

The conventional approach to determining the number of clusters is to run a clustering algorithm with multiple values for c and use cluster validity indices or stability indices to select an optimal c [120]. Validity indices are classified into two groups; external and internal. External indices validate a clustering solution by comparing it to an external source such as known cluster labels. Internal indices, in contrast, do not rely on any such external information. Their validation is based purely on the clustering solution. Since users typically do not have prior information, such as original labels, we focus our attention on internal validation and stability indices.

Many internal indices have been proposed to estimate the number of clusters. These methods include the CH Index [18], Silhouette index [92] and the Gap index [106]. Most of these indices are distance-based and measure cluster compactness using pairwise distances. These measures also compute cluster separation by computing the distances between cluster centers. A value for c is chosen, such that cluster compactness or cluster separation is maximized. Distance-based internal indices are sensitive to noise, outliers, and the scaling of the variables of interest. This sensitivity is often circumvented when the data is processed to remove outliers and scaled to zero mean and unit variance before clustering is done. This method of estimating c can be computationally expensive. This is because a clustering algorithm is run many times, and the validation index is computed, which can be costly for large datasets.

Cluster stability methods seek to find a clustering that is robust to data perturbation and noise. The basic idea is to find a clustering of a dataset that is robust to random perturbation. According to [10], if the data is over-clustered, the clustering algorithm will need to randomly split true clusters leading to a lack of stability in the resulting clusters. Similarly, if the data is under-clustered, the clustering algorithm will need randomly merge true clusters, again leading to the lack of stability in the resulting clusters. One method which assesses cluster stability is based on resampling. This approach clusters overlapping subsets of the data and then computes a similarity score between the clustering of the subsets. A similarity score is computed as the pairwise distances between these clustering, and the stability score is the average of these pairwise distances. A value for c is chosen, such that

it maximizes the stability value over a range of different values for c. See [72] for a detailed treatment of the resampling based method for cluster stability. Another method not discussed here is based on building an ensemble. This method combines many clustering of the dataset over a range of c values. See [102] for the treatment of this approach.

2.2.2 Statistical significance methods

Another way for estimating the number of clusters is by using more formal statistical tests. This is done by finding a value of c such that it provides the most significant evidence against a null hypothesis of a single cluster. The hypothesis includes the uniformity hypothesis [95] and the unimodal hypothesis [14, 41, 52]. Under the unimodality hypothesis, the data is viewed as a random sample generated from a multivariate Gaussian distribution. The data is viewed as a random sample generated from an d-dimensional uniform distribution under the uniformity hypothesis. For both these hypotheses, evidence for or against the null can be computed using internal validation indices discussed in section 2.2.1. Since internal indices are used, multiple clustering of the dataset for different values of c is again required. See [52] for a comprehensive treatment of these methods.

Another method for estimating the number of clusters makes use of the nested nature of hierarchical clustering. The results of hierarchical clustering methods are presented using a binary tree or a dendrogram, which provides an intuitive way of viewing the hierarchical structure in the data. The number of clusters is estimated using a statistical test in a top-down manner at each node in the resulting dendrogram. These tests are designed such that the null hypothesis tests data homogeneity by computing a test statistic on the data and comparing it to a suitable null distribution. The null distribution is computed by making specific assumptions about the data. These assumptions can be both parametric and non-parametric. Methods that have implemented this approach include [44, 58]. These methods either use a Gaussian null model [58] or a Unimodal null model [44] described below.

2.2.3 The Gaussian model for statistical significance

This model assumes clusters are derived from a single Gaussian distribution parameterized by a mean vector μ and a covariance matrix Σ. One of the most notable methods implementing this model is [69]. The hypothesis tested is formally stated as

H_0 : A pair of clusters follow a single Gaussian dist.
H_a : A pair of clusters do not follow a single Gaussian dist.

Where a p-value obtained would provide evidence for or against the null.

The test statistic is the 2-means cluster index which is a tightness or compactness measure for clusters is defined as follows:

$$CI = \frac{\sum_{k=1}^{2} \sum_{j \in C_k} \parallel x_j - \bar{x}^k \parallel}{\sum_{j=1}^{n} \parallel x_j - \bar{x} \parallel} \tag{2.5}$$

Where $\bar{x}^k$ is the mean for each cluster $k \in \{1,2\}$, C_k is the sample index for the cluster k and $\bar{x}$ is the overall mean. We note that a smaller value of this function is associated with larger variation explained by a given clustering, implying a better clustering.

Lastly, significance for a pair of clusters is obtained by comparing the test statistics computed from the data with the same statistic computed from the null distribution. The null distribution is empirically estimated by computing the test statistic for many datasets generated from a single Gaussian distribution. The Gaussian distribution is estimated from the original dataset using methods described in [48, 69]. A p-value is then obtained by computing:

$$p - value = \frac{\sum_{b=1}^{B} \{CI_b > CI_{data}\}}{B} \tag{2.6}$$

B is the number of datasets generated, and CI_b is the cluster index for the null dataset generated during iteration b.

In [58], this method is extended to a hierarchical setting by a Monte-Carlo based sequential hypothesis testing framework. First, a hierarchical tree is generated using agglomerative clustering. Next, at select nodes, starting from the root,

a test for significance of clustering is done between the two clusters below the current node. To control the Family Wise Error Rate (FWER) due to multiple testing, methods described in [78] is used. This method returns the k nodes that are significant which implies $k+1$ clusters.

2.2.4 The Unimodal model for statistical significance

The unimodal model is non-parametric and assumes that clusters follow a unimodal distribution. The Gaussian model makes specific assumptions about the distribution of clusters. This assumption may decrease statistical power if clusters are non-Gaussian. The method implemented in [44] assumes a unimodal distribution for clusters.

H_0 : A pair of clusters follow a single unimodal distribution.
H_a : A pair of clusters do not follow a single unimodal distribution.

Where a p-value obtained would provide evidence for or against the null. To perform the test, a null dataset X^0 needs to be computed with the requirements that it is as close as possible to X under unimodality conditions. This is achieved by using a Gaussian kernel density estimator (KDE) to model each feature in the dataset. This can be expressed as:

$$\hat{f(t;h_j)} = (nh_j)^{-1} \sum_{i=1}^{n} K(h_j^{-1}(t - X_{ij})) \tag{2.7}$$

Where h_j is the bandwidth; $K(\cdot)$ is the Gaussian kernel function; and $X_{1j}, ..., X_{nj}$ are the entries for feature j. According to [100], there exists a critical bandwidth h_{kj} such that:

$$h_{kj} = inf\left\{h_j : \hat{f}_j(\cdot; h_j)\right\} \tag{2.8}$$

has at most k modes.

h_{1j} is then computed and $\hat{f(t;h_{1j})}$ is re-scaled to have variance equal to that of the sample variance S. Finally, using the re-scaled KDE, bootstrap samples are

generated by:

$$X_{ij}^0 = (1 + \frac{h_{1j}^2}{\sigma_j^2})^{\frac{1}{2}}(X_{I_{ij}} + h_{1j}\varepsilon_i) \tag{2.9}$$

where $\varepsilon \sim N(0,1)$, σ^2 is the sample variance for feature j, and $X_{I_{ij}}$ are sampled uniformly with replacement from the observed data for feature j [44]. Finally, to ensure the covariance structure is maintained, the data is scaled to have mean 0 and variance 1 before computing X^0. X^0 is then be multiplied by the Cholesky root of the sample covariance matrix of X.

The 2-means cluster index defined in Section 2.2.3 for the Gaussian model is used to measure the strength of the clustering obtained from both X and X^0. Again, significance for a pair of clusters is obtained by comparing the test statistics computed from the data with the same statistic computed from the null distribution. The null distribution is empirically estimated by computing the test statistic for many datasets generated using the null unimodal distribution. The unimodal distribution is estimated from the original dataset using the methods described above and formally in [44]. A p-value is then obtained by computing:

$$\frac{\sum_{b=1}^{B}\{CI_b > CI_{data}\}}{B} \tag{2.10}$$

Where B is the number of datasets generated, and CI_b is the cluster index for the null dataset generated during iteration b.

It is worth noting that the authors mention the applicability of this method to a nested setting, as seen in [58]. However, no methods for controlling the FWER due to multiple testing are presented. This is a potential drawback that can be addressed by adopting the FWER control method used in [58] to this model. Both methods require bootstrapping to generate a suitable null model, which can be very time consuming and impractical for large datasets.

2.2.5 ScRNA-seq specific methods for estimating C

A few clustering algorithms developed for scRNA-seq datasets estimate the number of clusters directly or indirectly [38, 59, 113, 127]. For example, the consensus clustering method SC3 [59] uses random matrix theory to compare the eigenval-

ues of a transformed gene expression matrix to estimate the number of clusters. However, this is highly susceptible to overfitting when the data sparsity is high. Overfitting due to data sparsity is particularly worrisome since scRNA-seq datasets have an abundance of zero entries. Another method that attempts this is SIMLR [113]. SIMLR estimates the number of clusters by optimizing heuristic functions based on network diffusion. SIMLR's optimization algorithm requires a range of the possible number of clusters, which, in turn, requires prior knowledge about the possible number of clusters. Imposing prior knowledge on the possible number of clusters renders clusterability analysis useless since there is an implicit assumption that the dataset contains clusters. However, this may not be the case.

Chapter 3

Methods

This chapter details our approach for testing for clusterability and computing the number of clusters. In Section 3.1, we describe how multimodality testing using the Dip test can be used as a measure of clusterability. Section 3.6 describes the Computing Clusters Through Multimodality Testing (CCMT) procedure for computing the number of clusters. In Section 3.7, we formally describe the simulation setup used for methods evaluation. In Section 3.8, we discuss the real datasets used for benchmarking and comparisons with other methods. In Section 3.10, we detail the metric used for evaluation on both simulated data and real benchmarking data. In Section 3.11, we provide details about the clustering methods used for comparison against the CCMT procedure. In Section 3.12, we discuss how both the multimodality testing and the CCMT procedure performance are assessed the simulation studies and real data. Finally, in Section 3.13, details of how computational running times are computed are presented.

3.1 Testing expression patterns for multimodality

To assess the cluster structure inherent to a scRNA-seq dataset, we look for multimodality in the gene expression patterns for cells. We use multimodality testing as a proxy for clusterability. Formally, this method takes as input a scRNA-seq gene expression matrix M with n rows and p columns. We denote y_{ij} as the raw expression count for gene j in cell i. We assume that the count values are integer-

valued and have not been normalized. Before any analysis is done, we filter genes by selecting genes expressed in at least 5 cells. We also select cells with a minimum of 200 genes expressed and a maximum of 2500 genes. We cap the number of genes expressed in a cell because cells with extremely high gene counts tend to be multiplets, and cells with very low gene counts tend to be the results of empty droplets. Cells with more than 5% mitochondrial content are also filtered out. This is the possible result of cells that were broken in a droplet during sequencing.

Following steps formally described below, the matrix M of counts is normalized, highly informative genes are selected, and dimensionality reduction using PCA is performed. Finally, multimodality is tested using the Dip test [42] on the cosine distances between the cells in PCA space. See Figure 3.1 for a visualization of the pipeline.

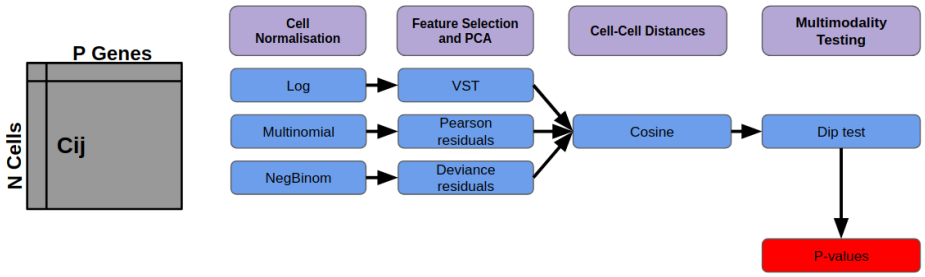

Figure 3.1: An overview of the pipeline for multimodality testing integrated. Starting with a molecular counts matrix, **Cell Normalisation** is done using three methods. Next, **feature Selection and PCA** is done for each normalisation method followed by computing **cell to cell** Cosine distances for each normalisation method. Finally **multimodality testing** is done using **Dip test** on the distribution of these cell to cell distances. The output is a table containing three p-values. One for each normalisation method.

3.2 Counts modelling and normalization

Normalization of the counts matrix is a pivotal step in analyzing scRNA-seq data. Counts normalization is typically done to address differences in sequencing depth

from cell to cell. Normalization of counts also helps to adjust for various forms of noise or noise bias in the sequencing process, so it does not heavily confound real biological differences present. To combat this, we use three different methods(described below) for normalizing the counts matrix.

3.2.1 Log transformation of normalized counts

Log-transformed molecular counts have become widely used in the analysis of scRNA-seq data. This is because of their statistical simplicity and interpretation. These transformed values can represent log-fold changes in gene expression between cells, which is sometimes used for informative genes selection and differential expression analysis. Log-transformation also helps to reduce the severity of stochasticity in the counts for genes that are highly expressed. Formally, the log-transformed model is defined as follows: let y_{ij} be the observed molecular count for cell$_i$ and gene$_j$. Let $n_i = \sum_j y_{ij}$ be the total molecular counts in cell$_i$. Now let $\hat{\pi_{ij}} = \frac{y_{ij}}{n_i}$ be the true observed proportion of gene$_i$ in cell$_i$. We can define the log transformed values as as $z_{ij} = log_2(c + \hat{\pi}_{ij} * m)$ where $c = 1$ is a pseudo-count to deal with situations where $\hat{\pi}_{ij} = 0$, and m is a scaling factor(typically set to 10^6). The resulting z_{ij} values are used as normalised expression counts for downstream analyses. The *Seurat* package [17] was used to compute the log-transformed values.

3.2.2 Multinomial normalization

Recent studies have shown that log-transformation of molecular counts causes statistical biases, which leads to loss of power during downstream inferences [39, 46, 71, 107]. These biases include inadequate variance stabilization caused by artificial variance inflation. The second normalization method assumes the counts are derived from a multinomial model and thus models the molecular counts directly.

Formally, the multinomial model is defined as follows: let y_{ij} be the observed molecular count for cell$_i$ and gene$_j$ and let $n_i = \sum_j y_{ij}$. Now let π_{ij} be the true unknown relative abundance for gene$_j$ in cell$_i$. We can say that the vector $y_i = (y_{i1}, ..., y_{i,J})^T$ with constraint $n_i = \sum_j y_{ij}$ follows a multinomial distribution with a density function

$$f(y_i) = \binom{n_i}{y_{i1}, ..., y_{i,J}} \prod_j \pi_{ij}^{y_{ij}} \tag{3.1}$$

This model is computed using Maximum Likelihood Estimation on the raw molecular counts using a Binomial or Poisson approximation. The Pearson residuals or Deviance residuals values are used as normalized expression counts for downstream analyses. For a more thorough treatment of the multinomial model for molecular counts, see [107]. The *scry* package [107] was used for fitting the model and computing the normalised values.

3.2.3 Regularized negative binomial normalization

The third and final model used to normalize the counts is a regularized negative binomial model. This model is similar to the multinomial model in that they both models the counts directly. However, this model assumes the counts follow a negative binomial model. This is very similar to existing normalization models, such as ZIFA [86]. However, [39] showed that these models are prone to overfitting, which negatively affects downstream analyses such as clustering and differential expression analysis. To combat this, [39] proposes a regularized version of the negative binomial model. Here a generalized linear model is fitted with molecular counts for a gene$_j$ or y_{ij} as the response and sequencing depth as a covariate. Regularization is done by using kernel regression on parameters estimated from this model.

Formally, the regularized negative binomial model is defined as follows:

$$\log(E(x_i)) = \beta_0 + \beta_1 log_{10} m \tag{3.2}$$

Where x_i is a vector of molecular counts for gene$_i$, and m is a vector of count values assigned to the cells, i.e $m_j = \sum_i x_{ij}$. Here β_0 and β_1 are regularised across genes using kernel regression. For a thorough treatment of the regularized negative binomial model, see [39]. Normalised counts are computed using Pearson residuals defined using the regularized regression parameters. The normalised counts are defined as follows: $z_{ij} = \frac{x_{ij} - \mu_{ij}}{\sigma_{ij}}$, where $\mu_{ij} = \exp(\beta_{0_i} + \beta_{1_i} \log_{10} m_j)$ and $\sigma_{ij} = \sqrt{\mu_{ij} + \frac{\mu_{ij}^2}{\theta_i}}$ [39]. Here z_{ij} is the Pearson residual of gene$_i$ in cell$_j$, μ_{ij} is the expected

molecular count for gene$_i$ in cell$_j$, x_{ij} is the observed molecular count for gene$_i$ in cell$_j$ in the regression model defined earlier. Finally, β_0, β_1, and θ_i are parameters obtained from the model. The *sctransfrom* package [39] was used for fitting the model and obtaining the normalised scores.

3.3 Feature selection

A typical scRNA-seq dataset can contain expression values for thousands of genes, creating potential problems for downstream analyses. Firstly, with increasing dimensionality, most analysis methods do not scale well and dramatically increase computational costs. Secondly, the existence of a large proportion of non-informative genes will significantly increase the amount of noise in the data, which will reduce the true biological signal, thus reducing the power of statistical methods. This problem is circumvented by selecting a subset of $500 - 1500$ informative genes, and there are a few existing methods that can compute how informative genes are. For methods based on gene variability, genes are ranked based on their variability across cells, and only highly variable genes are retained [15]. For methods based on gene expression, genes are then ranked based on averaged expression across cells, and only highly expressed genes are retained [26]. Depending on the normalization method used, we select the top 500 most informative genes. The selection methods are described below.

3.3.1 Feature selection log transformation of normalized counts

To select highly informative genes for log-transformed normalized counts, first, a variance stabilizing transformation as described in [75] is performed. This is done to account for the mean-variance relationship inherent to scRNA-seq data that is not accounted for by the log transformation. Next, a line is fitted to the relationship between log(variance) and log(mean) of genes using local polynomial regression (loess). Highly informative genes are then selected by ranking the genes using the standardized residuals from the fitted line. The *Seurat* package [17] was used to compute these genes.

Feature selection for multinomalial normalization

For the multinomial model, the gene deviance residuals based on a binomial approximation to the multinomial model are defined as:

$$D_j = 2\sum_i \left[y_{ij} \log \frac{y_{ij}}{n_i \hat{\pi}_j} + (n_i - y_{ij}) \log \frac{n_i - y_{ij}}{n_i(1-\hat{\pi}_j)} \right] \tag{3.3}$$

where Di is the deviance for gene$_j$ and other parameters are defined as before in Section 3.2.2 [107]. This model assumes constant gene expression across cells, so genes that deviate significantly from this model are genes that are the most informative. Gene deviance values are sorted, and the most deviant genes are used for clusterability analyses. The *scry* package [107] was used for computing the gene deviance.

3.3.2 Feature selection for Regularized Negative Binomial Normalisation

For the negative binomial normalization, the gene residuals z_{ij} defined in Section 3.2.3 are sorted. The top genes with the highest residuals are selected and used for downstream analyses. The sctranform package [39] is used to compute the gene Pearson residuals.

3.4 Dimensionality reduction

Dimension reduction methods have become an integral part of most scRNA-seq analysis pipelines since scRNA-seq datasets often have high dimensionality. Dimensionality reduction helps make analysis methods faster and scalable by producing lower dimensional embedding of high dimensional datasets. The lower dimension projections are computed such that they preserve the majority of relevant signals in the dataset. These methods can also be used for denoising, visualizing, and data compression. Most dimension reduction methods come in two flavors, linear and non-linear. Linear methods assume that the data lies on a linear manifold and uses a linear function to project it onto a lower dimension. However, if this manifold is non-linear, a linear method will result in an insufficient lower

dimension embedding. Non-linear methods do not make this assumption and thus better able to better project non-linear data onto a lower dimension. Throughout this thesis, PCA is used for dimensionality reduction.

3.5 Multimodality testing

Testing for multimodality in gene expression patterns is an integral part of this work. As an example, consider a scRNA-seq dataset containing four distinct cell types. Cells of the same type will typically have similar gene expression patterns. Whereas, cells of different cell types will have different gene expression patterns. We can compute and plot the cosine values of gene expressions between cells. The distribution of cosine values will contain two modes. One mode centered around a large cosine value indicating cells in the same partition with similar gene expression patterns. The second mode will be centered around a small cosine value indicating cells in different partitions with different gene expression patterns. In contrast, a dataset containing a single cell type or homogeneous cell types should show roughly the same cosine values between cells indicating a single partition, see Figure 3.2.

Multimodality tests [3, 42, 100] discussed earlier in Section 2.1.3 can be used to assess this idea more formally. We use the Dip test to test the distribution of the cosine distances between the cells for multimodality. The idea of using modality tests on pairwise distances was proposed initially by [1, 2]. However, we have modified this approach in few ways for a more effective use on scRNA-seq data. Firstly, we have made the number of PCS used for computing distances to be dependent on the characteristics of the data. This is done by selecting the top most significant PCS. This helps to limit the influence of noise by selecting PCS that contribute a significant amount of signal. Secondly, we are computing distances between cells using the cosine distance defined in 3.4 instead of the Euclidean distance between the cells. The Dip test is used because of its scalability and statistical power.

$$D(\mathbf{X},\mathbf{Y}) = 1 - \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{3.4}$$

where x and y are both vectors of gene expression values.

For each normalization method, we perform this test as follows:

1. Compute the cosine distances between the top K PCS.
2. Run the Dip test on these pairwise distances with significant threshold α
3. If the p-value is $< \alpha$, the dataset shows significant evidence for clustering structure or clusterability.
4. If the p-value is $> \alpha$, the dataset does not show significant evidence for clustering structure or clusterability.

The Dip test provides a p-value representing the probability of seeing this distribution or a more extreme multimodal distribution if the data is unimodal. This p-value should be large for unimodal distributions and small for multimodal distributions. As is common practice, the significant threshold α is set to be 0.05. See Figure 3.1 for an illustration of this pipeline.

3.6 Computing clusters through multimodality testing

Below we describe the Computing Clusters Through Multimodality Testing (CCMT) procedure developed for estimating the number clusters. This method is similar to the statistical significance methods discussed in Section 2.2.2. We couple hierarchical clustering with discriminant analysis and multimodality testing to estimate the number of clusters. Below we discuss generating the hierarchical partition and discriminant coordinates. Next, the CCMT procedure is discussed in detail. Finally, we discuss how the results of the CCMT procedure are combined across normalization methods.

3.6.1 Generating a hierarchical partition

A hierarchical tree is generated by first transforming the normalized datasets by first computing the top k PCS of the cosine distances between the cells defined in equation 3.4. Next, hierarchical clustering is performed with the top k PCS using squared Euclidean distances coupled with Ward's minimum variance criterion.

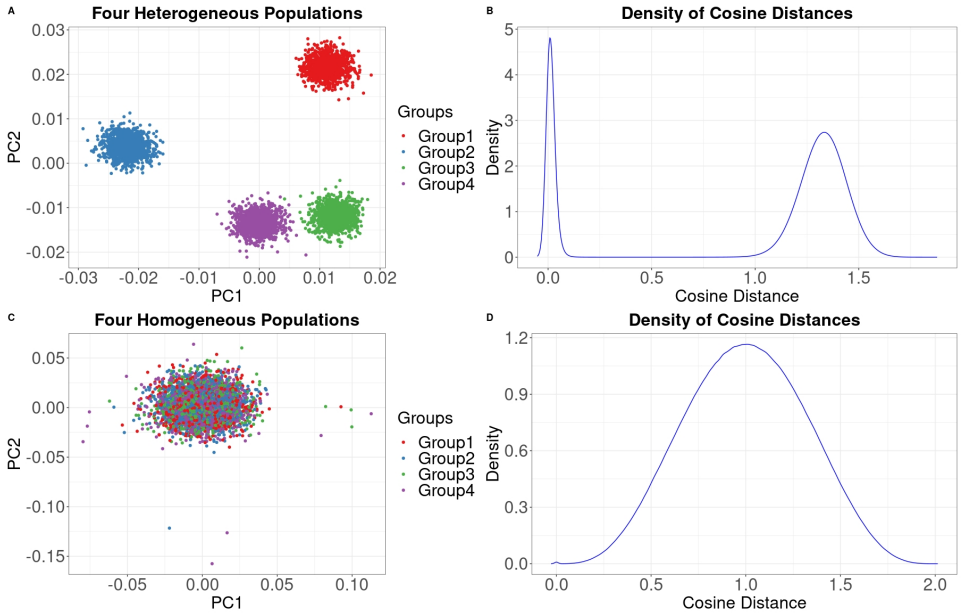

Figure 3.2: A) PCA plot of four simulated heterogeneous cell types. **B)** Density plot of the correlation distances between the cells in **A**. **C)** PCA plot of four simulated homogeneous cell types. **D)** Density plot of the cosine distances between the cells in **B**. Note that high values in **B** and **D** indicates low correlation and low values indicate high correlation.

3.6.2 Discriminant coordinates

Discriminant coordinates are often used to study clustering effects in a dataset including face recognition [28], action recognition [81] and gesture recognition [94]. One example of methods used is Fisher discriminant coordinates [32], which computes the direction that best separates two classes. Discriminant coordinates aim to find a projection of two classes on to a line that best separates both classes. See Figure 3.3. By projecting the classes on to this discriminant line, the overlap between both classes decreases, which will enable the testing of clustering strength through multimodality testing. The method of discriminant coordinates is described below.

Consider a set of n observations (in this case, normalised expression values for genes) with each n_i belonging to $\{i = 0, 1\}$ with $\sum n_i = n$. Let x_{ij} be the j^{th} observation in group i and denote

$$\bar{x}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} x_{ij} \quad \textit{and} \quad \bar{x} = \frac{1}{n_i}\sum_{i=1}^{2}\sum_{j=1}^{n_i} x_{ij} \tag{3.5}$$

We can then define two matrices, the within group covariance W:

$$W = \frac{1}{n_i}\sum_{i=1}^{2}\sum_{j=1}^{n_i}(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})' = \sum_{i=1}^{2}(n_i - 1)S_i \tag{3.6}$$

and the between group covariance matrix B:

$$B = \frac{1}{n_i}\sum_{i=1}^{2}\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)' = \sum_{i=1}^{2} n_i(x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)' = (n-g)S \tag{3.7}$$

Here S_i is the sample covariance matrix for group i, and S is the sample covariance matrix for the combined groups. A discriminant coordinate vector is defined as as the vector c that maximizes the function:

$$J(c) = \frac{c'Bc}{c'Wc} \tag{3.8}$$

$J(c)$ can be maximized computing the eigenvector associated with the largest eigenvalue of $W^{-1}B$. This eigenvector provides the direction that maximizes the between-class variance of B while minimizing the within-class variance of W. By projecting the classes on to the vector c, we decrease the overlap between both classes, which will enable us to test for separability between the classes.

3.6.3 The CCMT procedure

The CCMT procedure is based on testing for multimodality sequentially using the Dip test coupled with discriminant coordinates. Contrary to established cluster significance testing methods, this method requires no formal hypothesis testing except for the one performed by the Dip test. The details of the test are formally described below.

1. Let M denote the normalized counts matrix generated using one of the described methods and M^c be a matrix of cosine distances between points in M.

2. Let D denote the top k PCS for M^c.

3. Generate a tree T from D using the ward's minimum variance method on squared Euclidean distances.

4. For each node in T, let C_1 and C_2 be two clusters containing the children nodes in the two respective sub-trees.

   (a) Let $D_{c_{12}}$ be D subsetted to contain only the points in the two subtrees at the current node being considered.

   (b) Compute a discriminant coordinates vector w that best separates C_1 and C_2.

   (c) Generate a one dimensional projection of $D_{c_{12}}$ on to w and denote this as x.

   (d) Finally, test x for multimodality using the Dip test and return the p-value. If the classes are well separated, their projection onto the discriminant vector will be multimodal, and a significant p-value will be obtained.

5. Starting with the root node steps, $4a - 4d$ are performed on all nodes that parents were themselves significant or had a number of points in each subtree to be greater than n. For this implementation, the significance level $\alpha = 0.05$ and the minimum number of points $n = 10$.

6. Finally, a dendrogram is returned where each node's significance is labeled. See Figure 3.5. The points below the significant nodes are the significant partitions. The number of significant partitions is an estimate of the number of clusters. These significant partitions are also extracted and used as a clustering of the data.

See Figure 3.4 for an example of this method applied to a simulated dataset.

Both multimodality testing and the CCMT procedure can be integrated into a pipeline. The user would first apply multimodality testing and then proceed to run the CCMT procedure if there is significant evidence for cluster structure. See figure 3.6 for an illustration of the pipeline

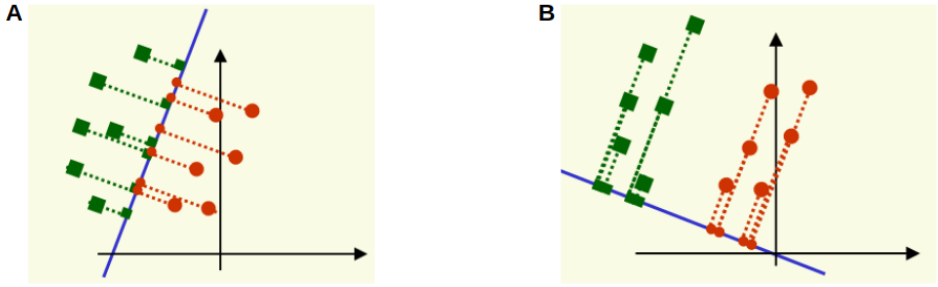

Figure 3.3: An illustration of discriminant coordinates. Fisher discriminant coordinates is the line that best separates both classes. In both panels, the red and green represents two different classes. **A)** Shows two classes that are not well separated by the discriminant line. **B)** Shows two classes that well separated by the discriminant line.

3.6.4 Combining significant partitions

For a given dataset, the CCMT is performed on each of the three normalized versions. We provide two methods to combine the results across normalized datasets. The first uses an ensemble method which combines the clustering solutions across the three normalized datasets. See Figure 3.7. The second uses a multiview approach [13, 20, 122], which combines all three normalized versions of the dataset before the CCMT procedure is performed. This approach computes a set of PCS common to all three normalized datasets [109]. The CCMT procedure is then applied to the common PCS. See figure 3.8.

3.7 Simulation studies

To test the robustness and scalability of the methods developed in this thesis, a set of synthetic datasets was generated using the splatter package [123]. This package was designed for simulating scRNA-seq count data and allows one to vary the number of cells, genes, and clusters, with varying levels of separability and varying degree of sparsity. The Splatter tool enables the possibility of generating synthetic datasets that capture different properties that the typical scRNA-seq dataset may have. This tool also enables the ability to simulate problems typically associated with the analysis of scRNA-seq datasets. These problems include sparsity due to

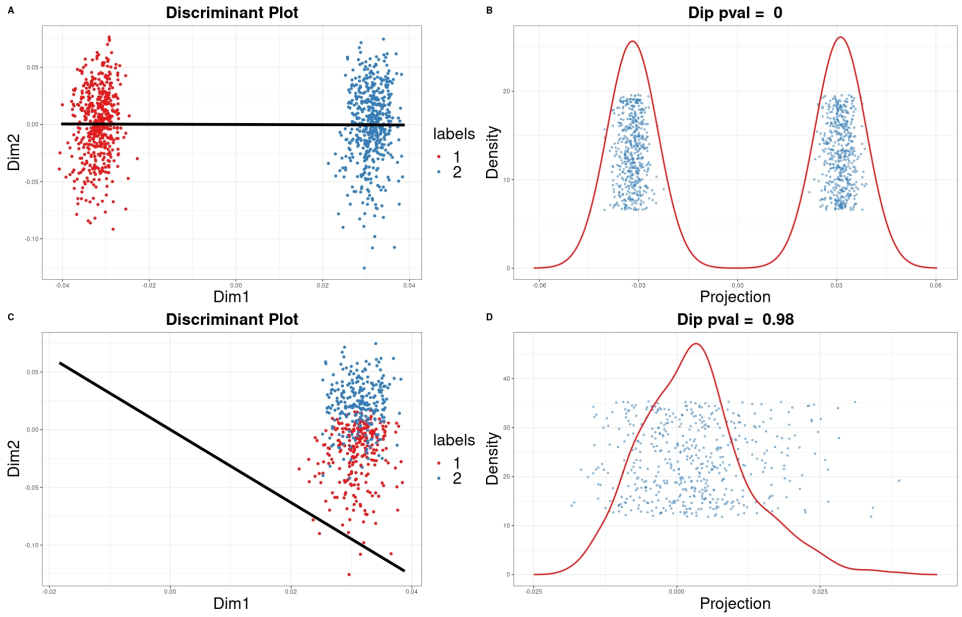

Figure 3.4: An example of discriminant coordinates significance clustering test. **A) and C)** Shows two classes that are separated by a discriminant line in black. **B) and D)** Shows the Dip test applied to the projection of the two classes onto this discriminant line. Note that for **B**, a significant p-value is obtained reflecting the separation between the classes. For **D**, the opposite is observed. P-values $< 2e^{-16}$ were rounded to 0.

dropout events, low separability between cell types, and an increasing number of cells. We adopted a similar simulation paradigm used in [62] to generate three simulation setups. We, however, changed some of the parameters used in the simulation setup for more rigorous testing. The simulation setups are described below.

3.7.1 Simulating data scalability, differing cluster number and sizes

The first simulation setup assessed scalability and differing cluster sizes. Scalability here is defined as the capacity for methods developed to adequately handle an increasing number of cells. We also require it to be able to handle an increasing number of clusters with different relative sizes. For this purpose, counts are simulated for a range of cells in $\{5000, 10000, 15000\}$, each with 1000 genes. For each possible number of cells, clusters in the set $\{4, 8, 16\}$ are generated. For

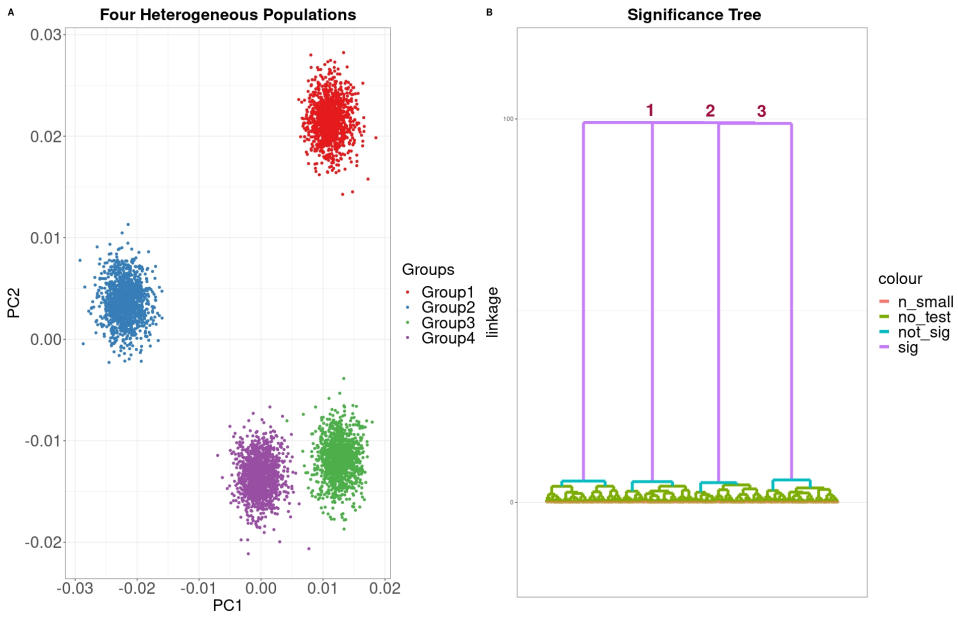

Figure 3.5: An illustration of the CCMT procedure **A)** Shows a PCA plot of four simulated heterogeneous population. **B)** Shows the significance tree after applying the CCMT procedure. Note that the tree is colored by significance and if a test was performed at a node or not.

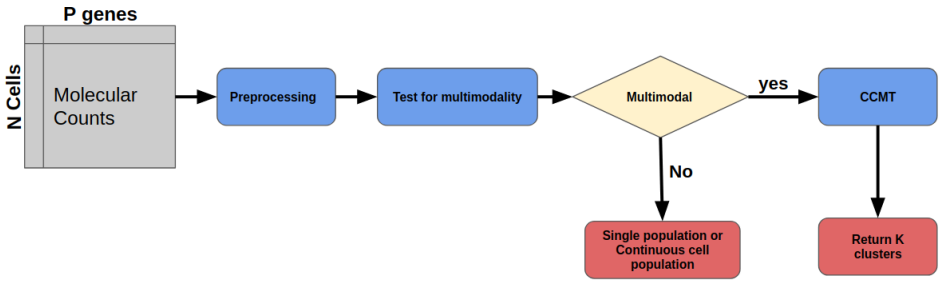

Figure 3.6: Overview of how multimodality testing and the CCMT procedure can be integrated into a pipeline. If the test for clusterability returns evidence for a cluster structure, the CCMT procedure is applied and the clusters are returned. Otherwise there is a possibility of a single cell population or the cells are on a continuum.

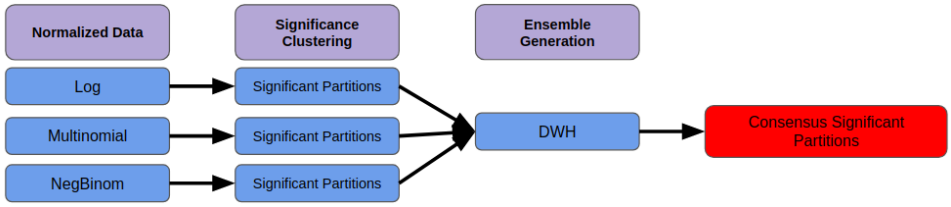

Figure 3.7: An illustration of the ensemble method for combining significant partitions across normalisation methods. An ensemble is generated using the soft least squares consensus partition method (DWH) implement in the *Clue* package.

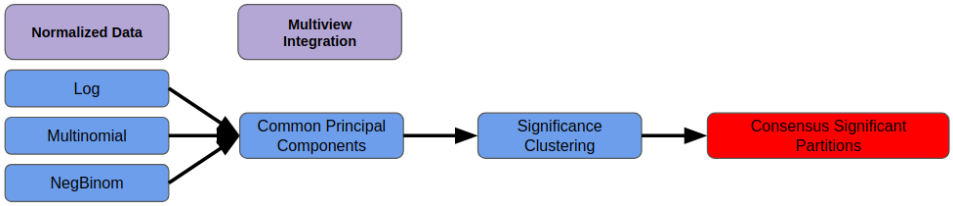

Figure 3.8: An illustration of the multiview method for combining significant partitions. A set of common principal components (CPCS) is first computed for all three normalisation methods. These CPCS are then used for significant clustering.

each number of clusters, datasets were generated containing equal proportions of cells in each cluster and a dataset containing an uneven proportion of cells in each cluster. To generate uneven cluster proportions, $p_1, p_2, ..., p_k$ numbers were simulated from a uniform distribution such that $\sum_{i=1}^{k} p_i = 1$. Here, k is the number of clusters for the current simulation. For example, a set of proportions of cells for a dataset containing four clusters generated that are unbalanced would be $\{0.20, 0.10, 0.40, 0.30\}$. In contrast, the proportions of cells for a dataset generated containing four balanced clusters would be $\{0.25, 0.25, 0.25, 0.25\}$. This setup generated a total of 18 datasets. See Figure 3.9 for an illustration of this setup.

3.7.2 Simulating cluster separability

The second simulation setup assessed the ability to detect varying degrees of separation between clusters. Here, the number of cells and genes was fixed at 1000 and

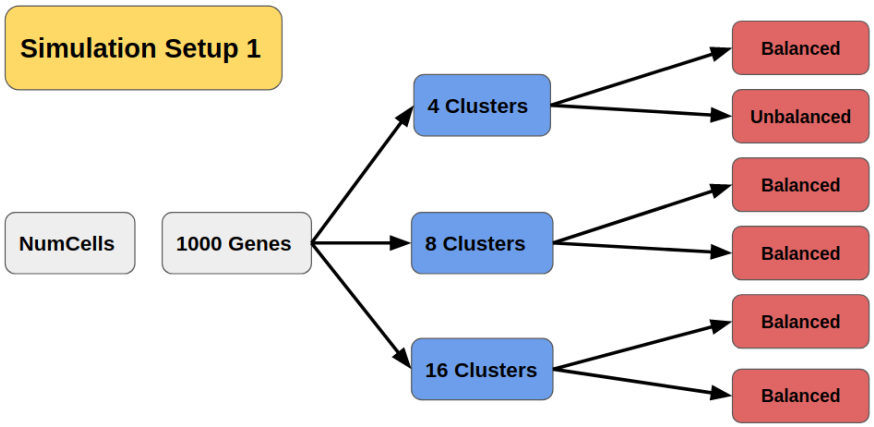

Figure 3.9: An illustration of **simulation setup 1.** A total of 18 simulated datasets were generated.

5000, respectively. The number of clusters was fixed to be 8 with relatively balanced sizes. The separability between clusters was then varied from no separation at all to well separated. To control cluster separability, the probability for a gene to be deferentially expressed between clusters was varied by generating 50 values in a range of $\{0, 0.5\}$. Figure 3.11 shows illustrations of both extremes. Separability values close to 0 produces clusters with low separability and values closer to 1 produces clusters with higher separability. This setup generated a total of 50 datasets. See Figures 3.10 and 3.11 for an illustration of this simulation setup.

3.7.3 Simulating data sparsity

In the third simulation setup, we assessed our method's ability to handle data sparsity or increasing proportions of missing information. Again the number of cells and genes is held constant at 5000 and 1000, respectively. The number of clusters remained the same again at 8, with the sizes balanced. To control the proportion sparsity in the generated datasets, the parameter(dropout.mid) controlling rate of zero counts in the logistic function that generates the counts in the Splatter package was varied. This parameter was varied in the range of $\{0, 1, 2, 3, 4, 5\}$ to generate datasets ranging in 20% to 90% sparsity. This setup generated a total of 6 datasets.

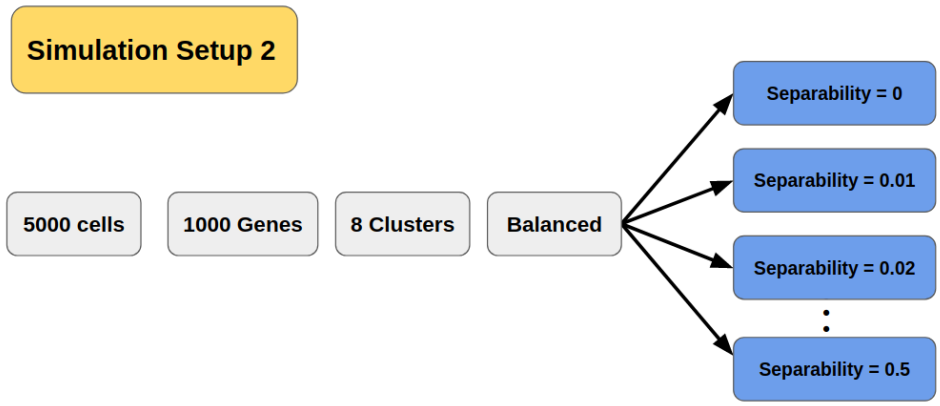

Figure 3.10: An illustration of **simulation setup 2.** A total of 50 simulated datasets were generated

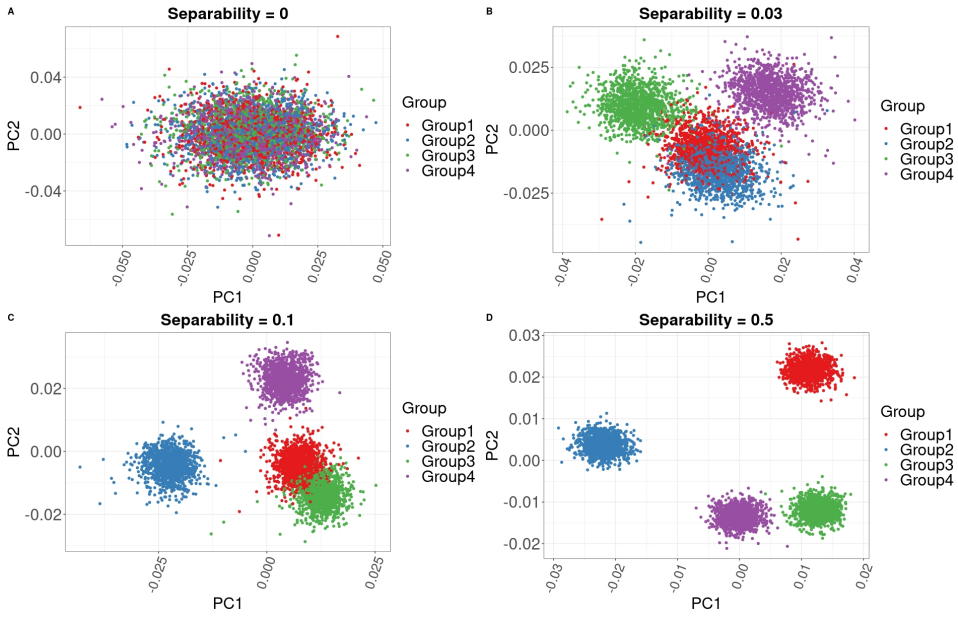

Figure 3.11: An illustration of the possible ranges cluster separability in 4 simulated clusters. **A)** A simulated dataset with no cluster separability. **B)** A simulated dataset with low cluster separability. **C)** A simulated dataset with intermediate cluster separability. **D)** A simulated dataset with high cluster separability.

See Figure 3.12 for an illustration. Figure 3.13 shows the possible ranges of data sparsity being considered and how it affects the datasets. Notice that as sparsity increases, the separability between the clusters decreases.

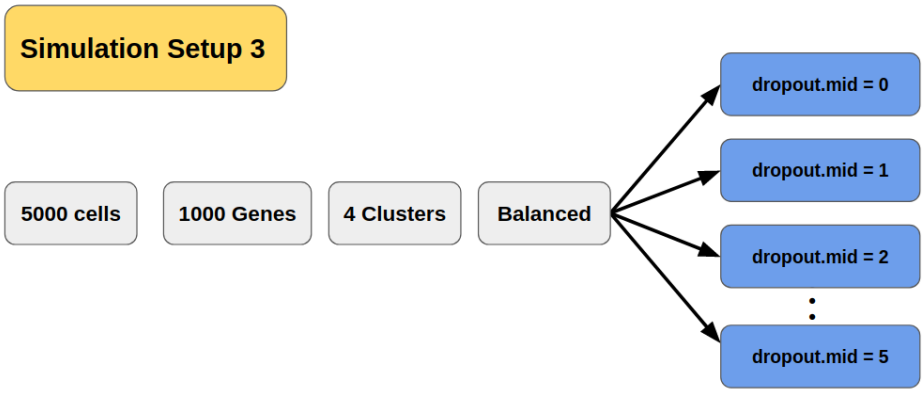

Figure 3.12: An illustration of **simulation setup 3.** A total of 6 simulated datasets were generated.

3.8 Benchmarking data

A set of published scRNA-seq datasets frequently used for analyzing scRNA-seq pipelines was used for evaluation as well. A majority of these datasets were obtained from the Hemberg lab Github repository. The Peripheral Blood Mononuclear Cells (PBMCS) datasets were obtained from the 10X genomics website. These datasets ranged in the number of cells, number clusters, cluster separability, sparsity, and overall complexity. These datasets are also derived from multiple organisms, including humans and mice, as well as multiple tissues, including blood, pancreas, brain, spleen, and retina.

The datasets used are formally described below. They contain two groups, the small scale, and the medium to large scale datasets. The small scale datasets are datasets that have less than 2500 cells. These datasets were used to benchmark CCMT against other scRNA-seq methods that do not scale very well for larger datasets. The medium to large datasets group are datasets that have greater than 2500 cells. These datasets were used to benchmark CCMT against other scRNA-

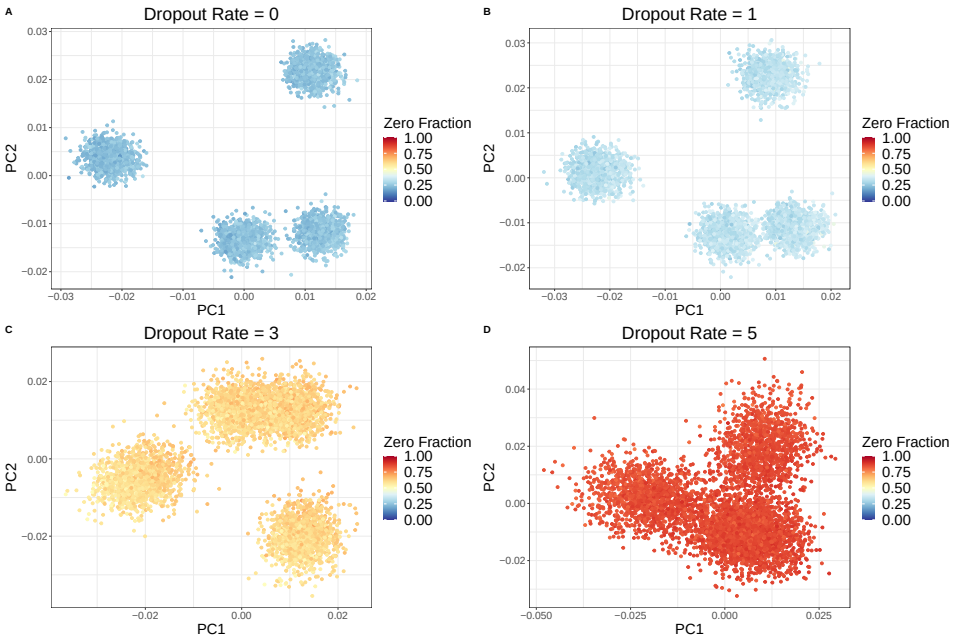

Figure 3.13: An illustration of the possible ranges of data sparsity in 4 simulated clusters. Sparsity is defined as the fraction of genes with 0 expression values in each cell. **A)** A simulated dataset with $20-30\%$ sparsity. **B)** A simulated dataset with $25-35\%$ sparsity. **C)** A simulated dataset with $50-70\%$ sparsity. **D)** A simulated dataset with $80-90\%$ sparsity.

seq methods that are well equipped to handle reasonably large datasets.

3.8.1 Small scale datasets

Seven small scale datasets described in Table 3.1 were used for benchmarking. For the CCMT procedure, these datasets were processed, as described in Chapter 3.1. For the other methods, preprocessing was done using the default settings provided by the methods. We note that with improvements in sequencing technology and reduced sequencing costs, it is now possible to profile hundreds of thousands more cells than considered in the small scale datasets. We decided to include these datasets for completeness and to provide a means for testing against other methods with computational bottlenecks such as an increasing number of cells.

Dataset	NumGenes	NumCells	NumPopulation	Sparsity
Gold [33]	58,302	925	3	0.85
Baron Mouse [8]	14,878	1886	13	0.89
Tasic [105]	24,150	1,679	18	0.69
Muraro [83]	19,127	2,126	10	0.73
Wang [114]	19,950	635	8	0.70
Xin [118]	39,851	1,600	8	0.88
Li [66]	55,186	561	9	0.79

Table 3.1: Small scale datasets used.

3.8.2 Medium to large scale datasets

Ten medium to large scale datasets described in Table 3.2 were used for benchmarking. For the CCMT procedure, these datasets were processed, as described in Chapter 3.1. For the other methods, preprocessing was done using the default settings.

Dataset	NumGenes	NumCells	NumPopulation	Sparsity
Silver5 [33]	17,043	8,352	11	0.96
Segerstolpe [96]	25,525	3,514	15	0.82
Klein [61]	24,175	2,717	4	0.66
Zheng [126]	15,568	3,994	4	0.97
Chen [21]	23,284	14,437	47	0.93
HMS [63]	28,962	11,127	9	0.97
Zeisel [124]	19,972	3,005	9	0.81
Romanov [91]	24,341	2,881	7	0.88
BaronHuman [8]	20,125	8,569	14	0.91
Shekar [97]	13,166	27,499	19	0.93

Table 3.2: Medium to large scale datasets

3.9 Obtaining ground truth clusters

For both the small and medium to large scale datasets used in this work, the cell labels provided by their respective authors are used as ground truth. We are aware that intrinsic difficulties exist when defining ground truth cell labels when evalu-

ating clustering or classification methods. This is due to the existence of multiple biologically plausible and interpretable way of clustering a scRNA-seq dataset, each representing relevant signals. The datasets used in this work were clustered with existing algorithms, and cell type labels were assigned using domain expertise. Therefore, we acknowledge that there is a risk of inherent bias in favor of the clustering method used to compute these cell type labels when making comparisons.

3.10 Evaluation metrics

To benchmark clustering solutions across methods, we use the Adjusted Rand Index (ARI). This metric has been routinely used in the clustering domain to evaluate how similar two clustering solutions are. It can also be used to compare how similar a clustering solution is to some know ground truth. The ARI score takes values between 0 and 1, with 0 being no similarity between two clustering solutions and 1 being perfect similarity between two clustering solutions.

Let D be a set containing n points. Denote a clustering of D as C, a set of non overlapping and non empty subsets $C_1, ..., C_k$. Now denote another clustering of D to be the set C' of non overlapping and non empty subsets $C^{'}_1, ..., C'_m$. Using C and C', we can create a k by m contingency table T such that the T_{ij} is the intersection of C_i and $C^{'}_j$.

We can then formally define the ARI score as:

$$ARI(C,C') = \frac{\sum_{i=1}^{k}\sum_{j=1}^{m}\binom{T_{ij}}{2} - u_3}{\frac{1}{2}(u_1+u_2) - u_3} \tag{3.9}$$

where $u_1 = \sum_{i=1}^{k}\binom{|C_i|}{2}$, $u_2 = \sum_{i=1}^{m}\binom{|C^{'}_j|}{2}$, and $u_3 = \frac{2u_1u_2}{n(n-1)}$

3.11 Clustering methods

There are many algorithms available to cluster scRNA-seq datasets. We selected three to benchmark on the small scale datasets and one to benchmark on the medium to large scale datasets. For the small scale datasets, we evaluated *Seurat* [17], *SC3*

[59] and *SIMLR* [113]. For the medium to large scale datasets, only *Seurat* was used. This provides a comparison to the CCMT procedure against methods that are currently used. Both *SC3* and *SIMLR* were used only on the small scale datasets because they do not scale well with larger datasets. *Seurat* was chosen because it is currently one of the most widely used methods for clustering scRNA-data. It also scales quite well with larger datasets. For benchmarking, we evaluated clustering accuracy and consistency using the Adjusted Rand Index described in section 3.10. We also compared the running time for all of the clustering algorithms. The default parameters provided in their respective software packages were used for all clustering methods, including how data preprocessing is done. We note that more careful consideration of these parameters may provide different results.

3.12 Performance assessment

3.12.1 Simulation Studies

To assess the performance of multimodality testing under various simulated conditions. We look at the p-values returned from the Dip test on each of the simulation studies. For the first simulation setup, all the datasets show cluster structure. Therefore multimodality testing should return a significant p-value for all simulations. For the second setup, which simulates cluster separability, we expect the Dip test to be very sensitive to very low cluster separability. For the third setup, which simulates data sparsity, all the datasets have a cluster structure. Therefore, multimodality testing should return a significant p-value for all levels of sparsity.

For the CCMT procedure, these simulation studies provided a way to measure how well CCMT can recover the simulated partitions in each setup. The ARI score is used to measure how well the simulated partitions are recovered.

3.12.2 Positive control

The small scale and the medium to large scale datasets discussed earlier are used as positive controls datasets. These datasets have been used routinely to evaluate new clustering algorithms for scRNA-seq datasets. Therefore, there is an implicit assumption that these datasets exhibit significant cluster structure. These datasets

also provide a more realistic example of the type of datasets often encountered. It is expected that multimodality testing will return a significant p-value for all these datasets.

3.12.3 Negative control

To test how multimodality testing behaves when there is a single cluster, we selected the *Gold Standard* dataset used in [33]. This dataset is composed of three different cell lines cultured separately from human lung adenocarcinoma. To generate a set of three negative control datasets, we isolated each of the cell lines separately. See Figure 3.14 for an example of the isolated *HCC827* cell line.

The same three isolated cell lines from the gold standard dataset was used as negative controls for the clustering methods, including the CCMT procedure. The ARI score was again used to evaluate how well the clustering methods recovered the single cluster present.

3.13 Run time assessment

All clustering methods, including the CCMT procedure, were run in R programming language. To compare running times, the *Microbenchmark* R package was used. All methods were run with 12 threads with 32GB of ram on an Intel® Core™ i7-8750H CPU with 2.20GHz. All timing measurements include preprocessing steps.

3.14 Summary

In summary, we proposed a method for assessing the cluster structure level by testing the gene expression patterns between cells for multimodality. We also coupled multimodality testing with hierarchical clustering and discriminant analysis to estimate the number of clusters. We also developed various simulation studies to assess the reliability of the methods developed. Real datasets and methods used for benchmarking were presented as well as the metrics used for performance evaluation.

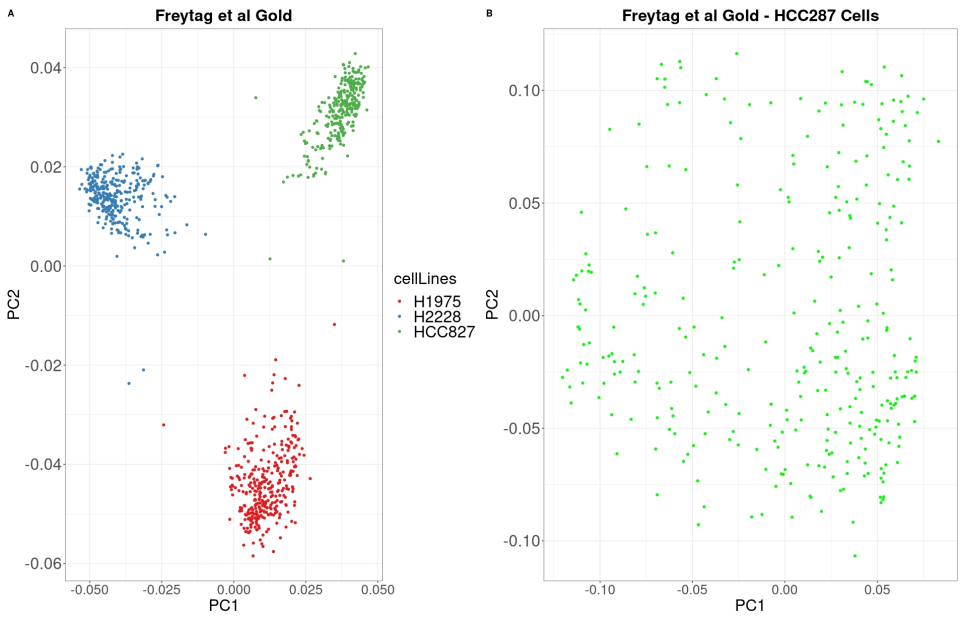

Figure 3.14: PCA plots of the dataset used for negative controls. This example shows the *HCC287* cells that were isolated and used as a one of the negative controls for both mulitmodality testing and benchmarking the CCMT procedure. **A)** shows a PCA plot of the three cell lines. **B)** shows a PCA plot of the isolated *HCC287* cell line.

Chapter 4

Results

In this chapter, we summarize the results of multimodality testing and the CCMT procedure. In section 4.1, results for multimodality testing are presented. First we show results of multimodality testing on the simulation studies discussed in section 3.7 and the positive control datasets (Section 3.12.2) as well as the negative control dataset (Section 3.12.3). Next, we show the results of the CCMT procedure applied to the simulation studies. For the CCMT procedure, we also show the results based on the positive (Section 3.12.2) control dataset as well as the negative control dataset (Section 3.12.3). Next, we present comparative results of the CCMT produce against clustering methods discussed in Section 3.11. Comparative results include the ARI score and running time assessment for both the small scale datasets (Section 3.8.1) and the medium to large scale data (Section 3.8.2).

To test the robustness and scalability of inferring clusterability through modality testing and using the CCMT procedure estimate the number of robust clusters, we generated a set of synthetic datasets discussed in 3.7. Each simulation setup was designed to simulate problems inherent to scRNA-seq data. This includes increasing data and cluster sizes, cluster separability, and data sparsity. We expect the Dip test to capture the simulated datasets cluster structure with high accuracy for multimodality testing. This implies that an insignificant p-value should be returned for datasets with no significant cluster structure. In contrast, a significant p-value should be returned for datasets with significant cluster structure. For the CCMT procedure, we expect a high ARI score for all the simulation setups implying

that this procedure can recover the simulated partition with high accuracy.

4.1 Evaluation of multimodality testing

4.1.1 Simulation studies

Tables 4.1, 4.2, 4.3 and Figure 4.1, shows the results of the clusterability analysis on all three simulation setups presented as heatmaps. Values in the tables and the heatmaps are p-values obtained after running the modality testing pipeline.

The first simulation setup results, which assesses data scalability, shows perfect performance across cluster sizes and the number of cells. This implies that clusterability testing using the Dip test scales well. This also implies that multimodality testing is not affected adversely by the relative sizes of clusters present in the dataset. This is an essential feature since it is rarely the case that clusters are perfectly balanced in scRNA-seq datasets. For the second simulation, which assesses cluster separability, all normalization methods perform reasonably well. This also implies that the Dip test is quite sensitive to cluster separability since it can find significant cluster structure even when clusters show low separability. Results for the third simulation setup, which assesses the effect data sparsity, are quite good as well. Multimodality testing can detect cluster structure in the presence of high data sparsity.

Dataset	**Log**	**NegBinom**	**Multinom**
5K Cells, 4 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
5K Cells, 8 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
5K Cells, 16 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
10K Cells, 4 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
10K Cells, 8 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
10K Cells, 16 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
15K Cells, 4 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
15K Cells, 8 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
15K Cells, 16 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$

Table 4.1: Table of p-values (p) obtained for multimodality testing done for simulating data scalability on balanced datasets.

Dataset	Log	NegBinom	Multinom
5K Cells, 4 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
5K Cells, 8 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
5K Cells, 16 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
10K Cells, 4 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
10K Cells, 8 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
10K Cells, 16 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
15K Cells, 4 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
15K Cells, 8 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
15K Cells, 16 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$

Table 4.2: Table of p-values (p) obtained for multimodality testing done for simulating data scalability on unbalanced datasets.

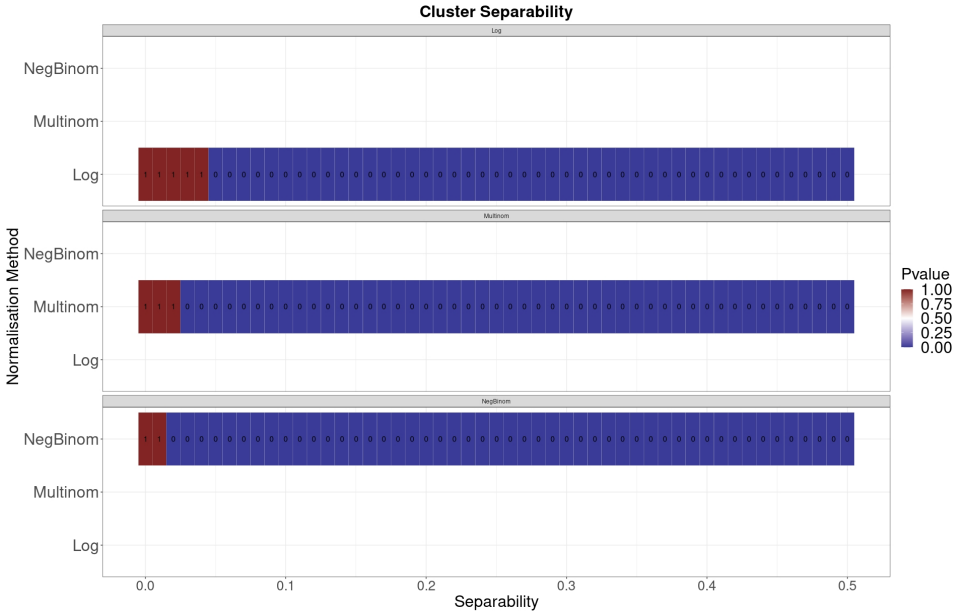

Figure 4.1: Heatmap of p-values obtained for multimodality testing done for simulating cluster separability. The x-axis is the cluster separability (higher values indicated higher separability between clusters) generated and the y-axis the normalisation method used. Note that p-values $< 2e^{-16}$ were rounded to 0 for visualization purposes.

Dropout Rate	Log	NegBinom	Multinom
0	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
1	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
2	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
3	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
4	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
5	1	$p < 2e^{-16}$	$p < 2e^{-16}$

Table 4.3: Table of p-values (p) obtained for multimodality testing done when simulating data sparsity. Higher dropout rate values indicates higher sparsity

4.1.2 Positive control

Table 4.4 shows the results of multimodality testing applied to the benchmarking datasets used as positive controls. Multimodality testing finds significant evidence of cluster structure in all datasets.

4.1.3 Negative control

Figure 4.2 shows the result of multimodality applied to the isolated *HCC287* cell line from the negative control dataset. There is no evidence of cluster structure (Dip p-value > 0.05) returned by multimodality testing across all the normalization methods. This shows that multimodality testing can correctly identify when there is no apparent cluster structure present. Similar results were obtained for the remaining two cell lines.

4.2 Factors affecting multimodality testing

4.2.1 The effect of data normalisation

Overall, multimodality testing performed well across all the simulation setups, positive control datasets, and the negative control dataset. However, taking a more in-depth look at the simulation studies results, we can make a few observations.

Consider the second simulation setup that addresses cluster separability; compared to the other two normalization methods; the log normalization method is not

Dataset	Log	NegBinom	Multinom
BaronHuman	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
BaronMouse	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Chen	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Gold	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
HMS	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Klein	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Li	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Maccosko	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Muraro	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Romanov	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Segerstolpe	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Shekar	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Silver5	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Tasic	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Wang	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Xin	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Zeisel	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
Zheng	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$

Table 4.4: Table of p-values (p) obtained for multimodality testing done for the benchmarking data. The table contains both the small and medium to large scale datasets.

as sensitive to cluster separability. This can be seen in Figure 4.1, where a significant p-value is returned after separability value of 0.05. For reference, Figure 3.11C is an example of how a dataset looks for separability value 0.03. In Figure 3.11B, the clustering structure is quite obvious. As such, both the multinomial and negative binomial normalization methods are able provide enough evidence for the Dip test to find significant cluster structure evidence. However, the Dip test applied to the log normalized data fails to find evidence for cluster structure. This implies that the log normalization typically done may not always be the best way of data normalization. It may obscure possible cluster structure by failing to find highly overlapping clusters.

Next, consider the third simulation setup, which assesses the effect data sparsity; all normalization methods show good performance for low to moderate spar-

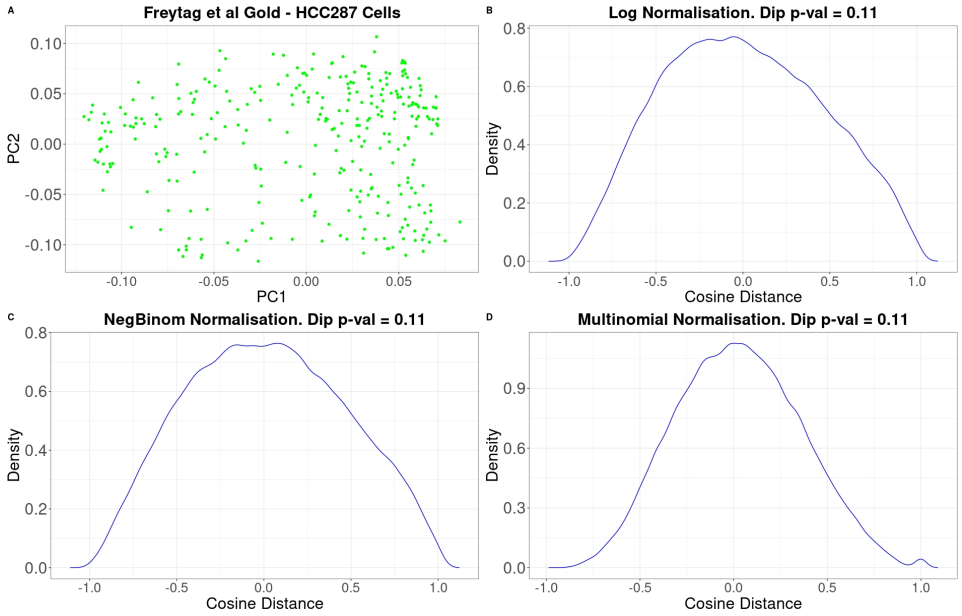

Figure 4.2: **A)** shows a PCA plot of the isolated *HCC287* cells. **B)** shows the density of the of the cosine distances between the top k PCS for the **log** normalisation method. **C)** shows the density of the of the cosine distances between the top k PCS for the **Negative Binomial** (NegBinom) normalisation method. **D)** shows the density of the cosine distances between the top k PCS for the **Multinomial** normalisation method. Similar results not shown here were obtained for the remaining two cell lines.

sity levels. See Table 4.3. However, for sparsity, $> 90\%$, the Dip test applied to the log normalized data fails to find evidence for cluster structure. The multinomial normalized and negative binomial normalized data have no trouble dealing with increasing sparsity levels. As noted in Figure 3.13, as sparsity increases, the separability between the clusters decreases. Since the log normalization method showed poor performance for low separability cases in the second simulation setup, it is not surprising to observe similar performances for higher data sparsity. Again, this is a cause for concern since log normalization is most often used for its statistical simplicity. Other normalization methods such as the negative binomial or the multinomial may be more appropriate.

4.2.2 The limitations of multimodality testing

The multimodality testing framework has performed well across simulation studies and real benchmarking data, including the negative controls datasets. However, to understand the situations where multimodality testing does not perform well, we constructed a fourth simulation setup, which combines the first two simulation setups from Section 3.7. To do this, the cluster separability was set to be 0.3 (see Figure 3.11B), while we varied the size and proportions of cells in each cluster as done in simulation setup 2 (see Figure 3.9). See Figures 4.3 and 4.4 for an illustration of and an example of this simulation setup.

Tables 4.5 and 4.6 show the results of multimodality testing applied to this simulation setup. Multimodality testing cannot capture the presence of cluster structure in all of the simulation datasets for both the balanced and unbalanced datasets. With high cluster overlap and an increasing number of clusters, multimodality testing struggles to find evidence for cluster structure. This pattern is evident for both the balanced and unbalanced datasets. For the balanced datasets (Table 4.5), the log normalization performs the worst while the multinomial normalization performs the best. For the unbalanced datasets (Table 4.6), all three normalization methods performs equally well. These results show that multimodality testing is limited when the cluster sizes are relatively balanced and highly overlapping. However, for the unbalanced datasets, multimodality testing is not as limited since it can capture the presence of cluster structure in over half of the datasets.

Dataset	**Log**	**Multinom**	**NegBinom**
5K Cells, 4 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
5K Cells, 8 Clusters	0.10	$p < 2e^{-16}$	0.06
5K Cells, 16 Clusters	0.10	$p < 2e^{-16}$	0.99
10K Cells, 4 Clusters	$p < 2e^{-16}$	0.10	$p < 2e^{-16}$
10K Cells, 8 Clusters	0.10	0.10	$p < 2e^{-16}$
10K Cells, 16 Clusters	0.10	0.10	1
15K Cells, 4 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
15K Cells, 8 Clusters	0.10	0.10	1
15K Cells, 16 Clusters	0.10	0.10	1

Table 4.5: Table of p-values (p) obtained for multimodality testing done for simulating data scalability with high overlap on balanced datasets.

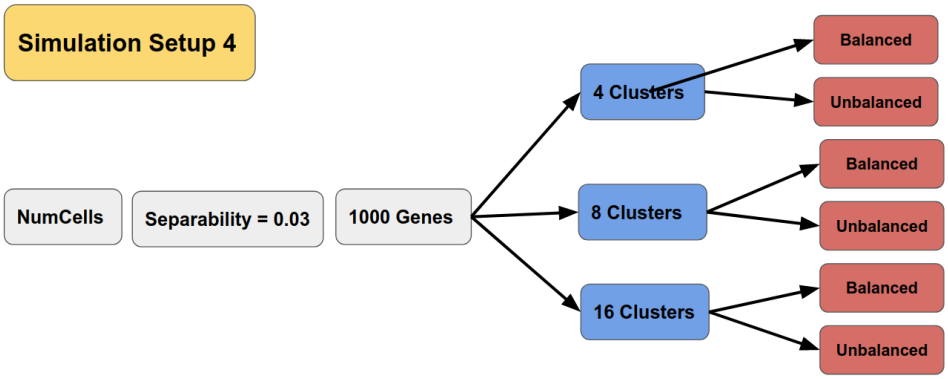

Figure 4.3: An illustration of **simulation setup 4.** A total of 18 simulated datasets were generated.

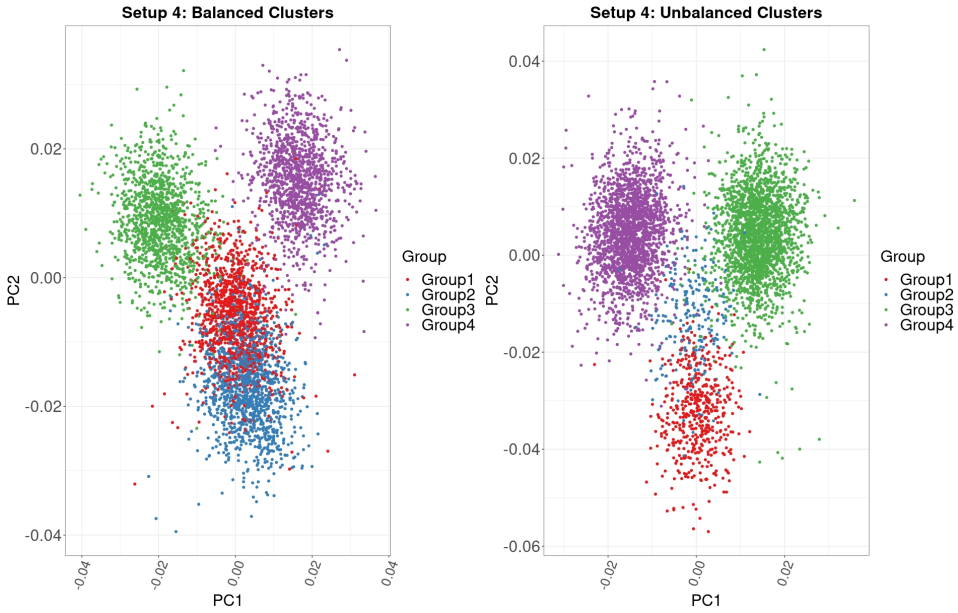

Figure 4.4: An illustration of the balanced and unbalanced datasets simulated.**A)** A simulated balanced dataset with 4 clusters. **B)** A simulated unbalanced dataset with 4 clusters. Each dataset was generated to have overlapping clusters.

Dataset	Log	Multinom	NegBinom
5K Cells, 4 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
5K Cells, 8 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
5K Cells, 16 Clusters	0.10	$p < 2e^{-16}$	1
10K Cells, 4 Clusters	$p < 2e^{-16}$	0.10	$p < 2e^{-16}$
10K Cells, 8 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
10K Cells, 16 Clusters	0.10	0.10	1
15K Cells, 4 Clusters	$p < 2e^{-16}$	$p < 2e^{-16}$	$p < 2e^{-16}$
15K Cells, 8 Clusters	0.10	0.10	$p < 2e^{-16}$
15K Cells, 16 Clusters	0.10	0.10	1

Table 4.6: Table of p-values (p) obtained for multimodality testing done for simulating data scalability with high overlap on unbalanced datasets.

4.3 Evaluation of the CCMT procedure

4.3.1 Simulation studies

Tables 4.7, 4.8, 4.9, and Figure 4.5 shows the results of the CCMT procedure applied to the three simulation setups. For the first simulation setup, the CCMT procedure does a good job of recovering the ground truth partitions simulated. Both the ensemble (Figure 3.7) and multiview (Figure 3.8) models do a good job of recovering the simulated partitions. The CCMT procedure is robust to the relative proportions of cluster sizes and different numbers of clusters. Results on the second simulation setup show that the CCMT procedure is generally sensitive to cluster overlap. However, the ensemble method appears to be more sensitive and stable compared to the multiview model. The multiview model is not always able to fully recapture the simulated partitions for clusters having very high overlaps. For example, in Figure 4.5, the clusters are relatively well separated for a separability value of 0.1. See Figure 3.11C. However, for this value, the multiview model fails to perfectly recover the simulated partition (ARI score = 0.5). See Section 3.10 for a discussion of the ARI score. Results on the final simulation setup (Table 4.9) shows that the CCMT procedure is also robust to increasing data sparsity. Both the multiview and the ensemble models are fully able to recapture the simulated clusters for increasing levels of data sparsity. Overall the CCMT procedure performs well across all

simulation studies.

4.3.2 Positive control

The CCMT procedure applied to the positive control benchmarking datasets shows good accuracy for most datasets. Table 4.10 and Figure 4.6, shows a table and a boxplot of the ARI score for all the benchmarking datasets. Figure 4.6 shows an average ARI score of 0.70, which implies that the CCMT procedure recovers the correct partitions approximately 70% of the time across all datasets. The CCMT procedure also obtains a maximum ARI score of 0.99 and a minimum of 0.19. Both the ensemble and the multiview models show similar performance across all datasets. However, the ensemble model has more variability, as seen by the long tail in Figure 4.6. Figure 4.7 shows scatter plots of the ARI score as a function of the predicted number of clusters by the CCMT procedure for the benchmarking datasets. In both plots, the red values indicate the correct number of clusters. In these plots, we judge performance by both the ARI scores and the predicted number of clusters. For a high ARI score, we would expect to see a closer agreement between the true number of clusters (x-axis) and the predicted number of clusters (red values). Both the ensemble and the multiview models tend to underestimate the number of clusters. Compared to the ensemble model, the multiview model generally returns a smaller number of clusters. However, both models, on average, return partitions with high overlap with the ground truth partitions, as seen by the high average ARI scores in Figures 4.7A and 4.7B.

4.3.3 Negative control

On the negative control datasets, the CCMT procedure fails to return a single cluster for the ensemble mode across all of the isolated cell lines (Table 4.11). The multiview model, in contrast, returns only a single cluster for two (*HCC827* and *H2228*) of the isolated cell lines. This implies that the ensemble model is more sensitive to the over-partitioning of the data compared to the multiview model. The multiview model is conservative when partitioning a dataset. This is similar to what is observed for the predicted number of clusters for the positive control datasets.

Dataset	Multiview ARI Score	Ensemble ARI Score
5K Cells, 4 Clusters	1	1
5K Cells, 8 Clusters	1	1
5K Cells, 16 Clusters	0.83	1
10K Cells, 4 Clusters	1	1
10K Cells, 8 Clusters	1	1
10K Cells, 16 Clusters	0.8	1
15K Cells, 4 Clusters	1	1
15K Cells, 8 Clusters	0.81	1
15K Cells, 16 Clusters	0.83	1

Table 4.7: Table of the ARI scores obtained from the CCMT procedure applied when simulating data scalability on balanced datasets.

Dataset	Multiview ARI Score	Ensemble ARI Score
5K Cells, 4 Clusters	1	1
5K Cells, 8 Clusters	0.97	1
5K Cells, 16 Clusters	0.87	1
10K Cells, 4 Clusters	1	0.98
10K Cells, 8 Clusters	0.99	1
10K Cells, 16 Clusters	0.91	1
15K Cells, 4 Clusters	1	1
15K Cells, 8 Clusters	0.97	1
15K Cells, 16 Clusters	0.9	1

Table 4.8: Table of the ARI scores obtained from the CCMT procedure applied when simulating data scalability on unbalanced datasets.

4.4 Factors affecting the CCMT procedure

Overall, the CCMT procedure provides a robust and accurate way of finding significant partitions in a dataset. However, there are a few parameters that may affect the performance of the CCMT procedure. These parameters include selecting highly informative genes and the number of principal components to use when running the CCMT procedure.

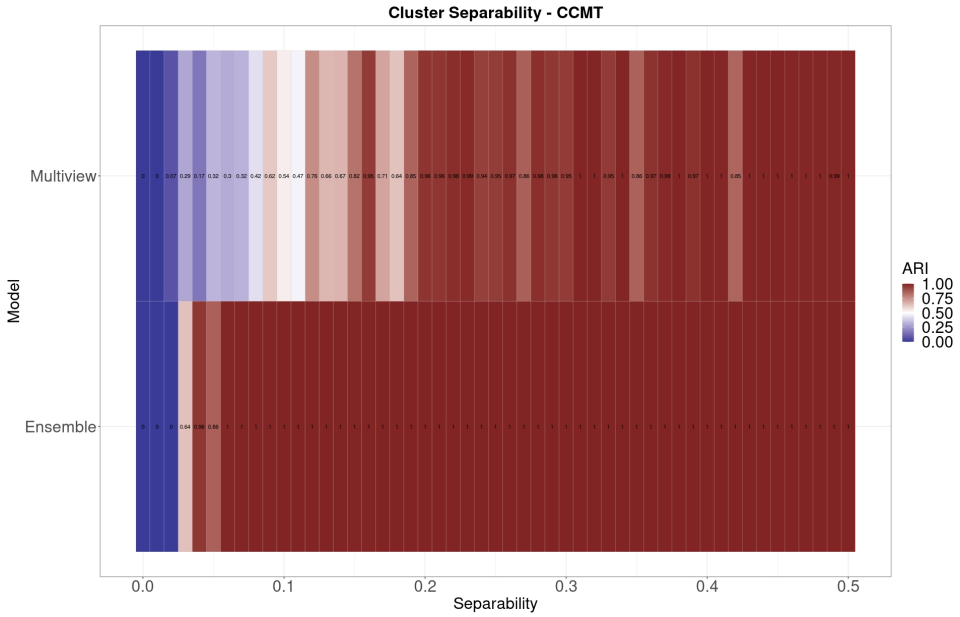

Figure 4.5: Heatmap of the ARI scores obtained from the CCMT procedure applied to simulating cluster separability. The x-axis is the cluster separability (higher values indicated higher separability between clusters) generated and the y-axis is the model used to combine the partitions across the three normalisation methods.

Dropout Rate	Multiview ARI Score	Ensemble ARI Score
0	1	1
1	1	1
2	1	1
3	1	1
4	1	1
5	0.98	0.99

Table 4.9: Table of ARI scores obtained for the CCMT procedure applied when simulating data sparsity. Higher dropout rate values indicates higher sparsity

4.4.1 The effect of the number of informative genes

In this thesis, we set the number of highly informative genes to 500. To test how the selection of highly informative genes affects the performance of the CCMT pro-

Dataset	Multiview ARI Score	Ensemble ARI Score
BaronHuman	0.89	0.9
BaronMouse	0.92	0.92
Chen	0.65	0.64
Gold	0.99	0.85
HMS	0.83	0.84
Klein	0.83	0.80
Li	0.59	0.74
Maccosko	0.87	0.9
Muraro	0.93	0.92
Romanov	0.67	0.19
Segerstolpe	0.58	0.53
Shekar	0.51	0.89
Silver5	0.55	0.50
Tasic	0.29	0.30
Wang	0.41	0.49
Xin	0.89	0.61
Zeisel	0.54	0.75
Zheng	0.97	0.7

Table 4.10: Table of ARI scores obtained for the CCMT procedure applied to the benchmarking data. The table contains both the small and medium to large scale datasets.

	Ensemble	Multiview	Cell Line
Number of clusters	4	1	HCC827
Number of clusters	7	1	H2228
Number of clusters	5	3	H1975

Table 4.11: CCMT applied to the negative control datasets.

cedure, for all the benchmarking datasets, we varied the number of highly informative genes from 500 to 2000 in increments of 500. Figure 4.8 shows the results of applying the CCMT procedure to a varying number of genes. There appears to be no significant increase in performance for the multiview model when increasing the number of genes. However, for the ensemble model, increasing the number of informative genes does increase the overall performance. From Figure 4.8, we see that setting the number of informative genes to 500 achieves the highest average

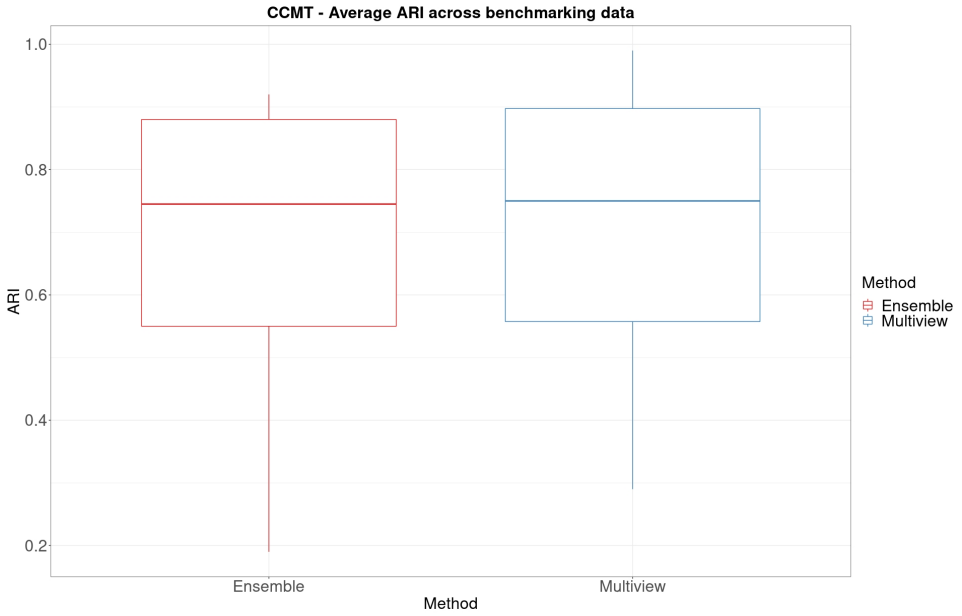

Figure 4.6: Boxplots of the average ARI score across all benchmarking datasets ($n = 18$) for both the ensemble and multiview model.

ARI score for the multiview model. The opposite is observed for the ensemble model. In contrast, using 2000 informative genes results in the lowest overall ARI score for the multiview model. Again, the opposite is observed for the ensemble model.

4.4.2 The effect of the number of PCS

In this thesis, we set the number of PCS by automatically finding the knee point of the PCA scree plot. To test the effect of the number of PCS selected for clustering, we varied the number of PCS from 5 to 25 in increments of 5. Figure 4.9 shows the results of applying the CCMT procedure to a varying number of PCS. For both the ensemble and multiview models, increasing the number of PCS negatively impacts the performance. The ensemble model is more affected by an increasing amount of PCS compared to the multiview model. The performance of the CCMT procedure when using a heuristic to select the number of PCS (Figure 4.6) is much better compared to when the number of PCS is fixed. This is most likely because increas-

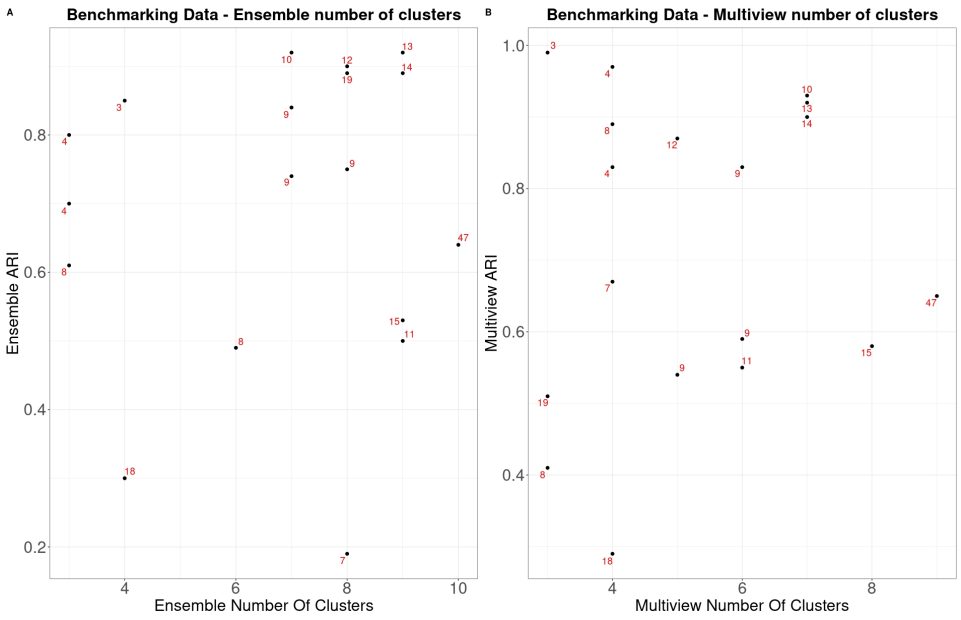

Figure 4.7: A) Scatter plot of the predicted number of clusters vs ARI score for the ensemble model. The x-axis is the predicted number of clusters and the y-axis ARI score. **B)**Scatter plot of the predicted number of clusters vs ARI score for the multiview model. The x-axis is the predicted number of clusters and the y-axis ARI score. For both scatter plots, the red values are the true number of clusters.

ing the number of PCS does not necessarily increase the data's signal. It's possible that the extra PCS included introduces significantly more noise content, which adversely affects the model's performance. This is most likely the reason the heuristic method works much better than fixing the number of PCS. The heuristic method can better identify the number of PCS to use based on the dataset's properties.

4.4.3 The limitations of the CCMT procedure

To assess the cases where the CCMT procedure significantly fails to recover ground truth partitions, we used the fourth simulation setup outlined in Section 4.2.2. Both the balanced and unbalanced datasets were passed to the CCMT procedure, and performance was assessed using the ARI score (Section 3.10) as before.

Tables 4.12 and 4.13 shows the ARI scores of the CCMT procedure applied to

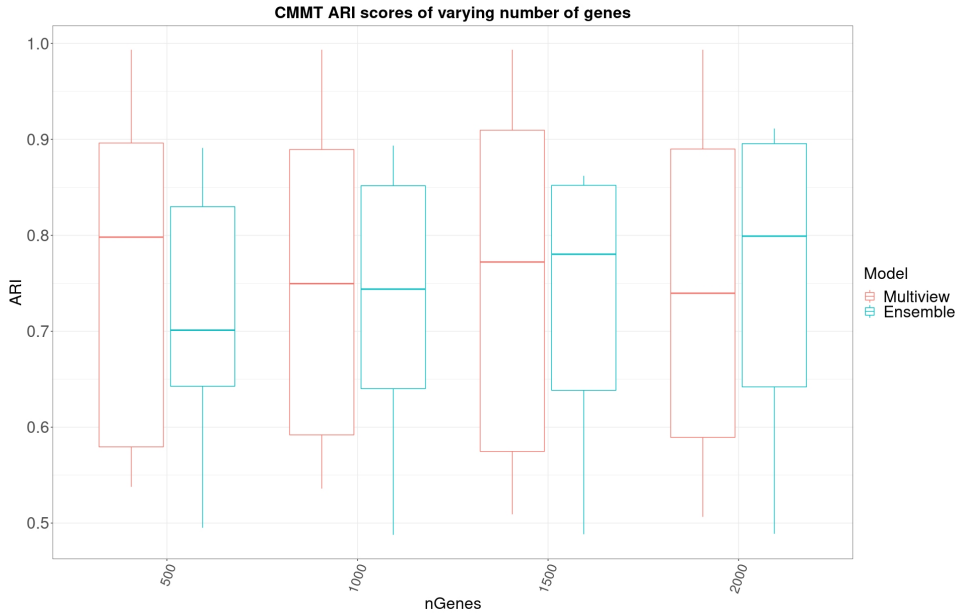

Figure 4.8: Box plot of ARI scores for varying number of genes for the both the Multiview and the Ensemble model. The x-axis is the number of genes and the y-axis is the ARI score.

the balanced and unbalanced datasets. The results show that the CCMT procedure is limited at recovering the ground truth clusters when there is high cluster overlap and an increasing number of clusters. This trend is more pronounced in the balanced dataset compared to the unbalanced datasets. For both the balanced and unbalanced datasets, the ensemble (Section 3.6.4) model, on average, does a better job at recovering the ground truth partitions compared to the multiview (Section 3.6.4) model.

4.5 Comparing CCMT to other clustering methods

The small scale and the medium to large scale datasets was used to compare the CCMT procedure against other clustering methods often used when analysing scRNA-seq data. We compared the ARI score and the running time across all methods.

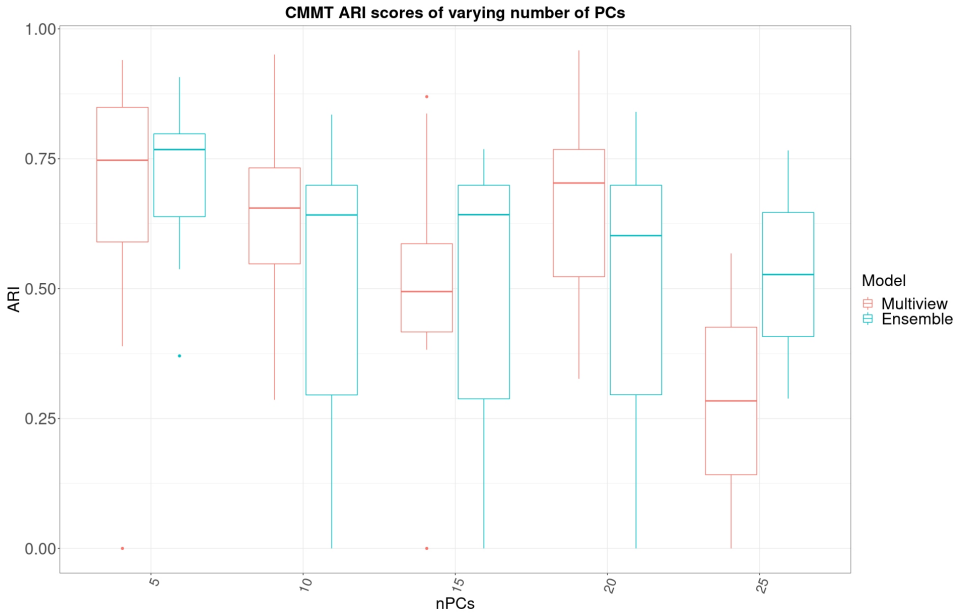

Figure 4.9: Box plot of ARI scores for varying number of PCS for the both the Multiview and the Ensemble model. The x-axis is the number of PCs and the y-axis is the ARI score.

Dataset	Ensemble ARI Score	Multiview ARI Score
5K Cells, 4 Clusters	0.97	0.60
5K Cells, 8 Clusters	0.29	0.17
5K Cells, 16 Clusters	0	0
10K Cells, 4 Clusters	0	0.55
10K Cells, 8 Clusters	0.08	0.17
10K Cells, 16 Clusters	0	0
15K Cells, 4 Clusters	0.98	0.71
15K Cells, 8 Clusters	0.46	0.18
15K Cells, 16 Clusters	0	0

Table 4.12: Table of the ARI scores obtained from the CCMT procedure applied when simulating data scalability with high overlap on balanced datasets.

Dataset	Ensemble ARI Score	Multiview ARI Score
5K Cells, 4 Clusters	0.94	0.94
5K Cells, 8 Clusters	0.88	0.40
5K Cells, 16 Clusters	0	0.09
10K Cells, 4 Clusters	0.86	0.82
10K Cells, 8 Clusters	0.12	0.23
10K Cells, 16 Clusters	0.03	0.08
15K Cells, 4 Clusters	0.76	0.75
15K Cells, 8 Clusters	0.29	0.29
15K Cells, 16 Clusters	0.02	0.07

Table 4.13: Table of the ARI scores obtained from the CCMT procedure applied when simulating data scalability with high overlap on unbalanced datasets.

4.5.1 Small scale datasets

For the small scale datasets, we compared the CCMT procedure against *Seurat*, *SIMLR* and *SC3*. *SIMLR* and *SC3* were specifically used for the small scale datasets because these methods have high computational complexity and thus do not scale well with larger datasets. Figure 4.10 shows boxplots of the ARI score and the running time of the methods applied to the small scale datasets. The running time (Figure 4.10A) was computed in nanoseconds, and values are presented on a log scale. Running time varied substantially between all the methods. *Seurat* was the fastest and *SIMLR* was the slowest. The CCMT models (ensemble and multiview) were the second and third fastest, respectively, and *SC3* was the fourth fastest. For the ARI score versus running time (Figure 4.10B), both the CCMT models have the highest average overall ARI score. *Seurat* has the second-highest overall average, with *SC3* coming third and *SIMLR* coming last. Even though Seurat has the fastest running time, it is not as accurate as the CCMT procedures. The CCMT methods may be slower compared to Seurat but are better able to recover ground truth partitions.

4.5.2 Medium to large scale datasets

For the medium to large scale datasets, we compared the CCMT models against *Seurat*. *Seurat* was explicitly used for these datasets because of its speed and ac-

curacy. *Seurat* scales quite well for larger datasets, which made comparisons to CCMT easier and efficient. Figure 4.11 shows boxplots of the ARI score and the running time of the methods applied to the medium to large scale datasets. On the medium to large scale datasets, *Seurat* is again the fastest method. Again, both the CCMT models have a similar running time. For the ARI score (Figure 4.11B), the CCMT models again have the highest overall average ARI score. Again, we see that the CCMT models do a better job of recovering the ground truth partitions than currently used methods.

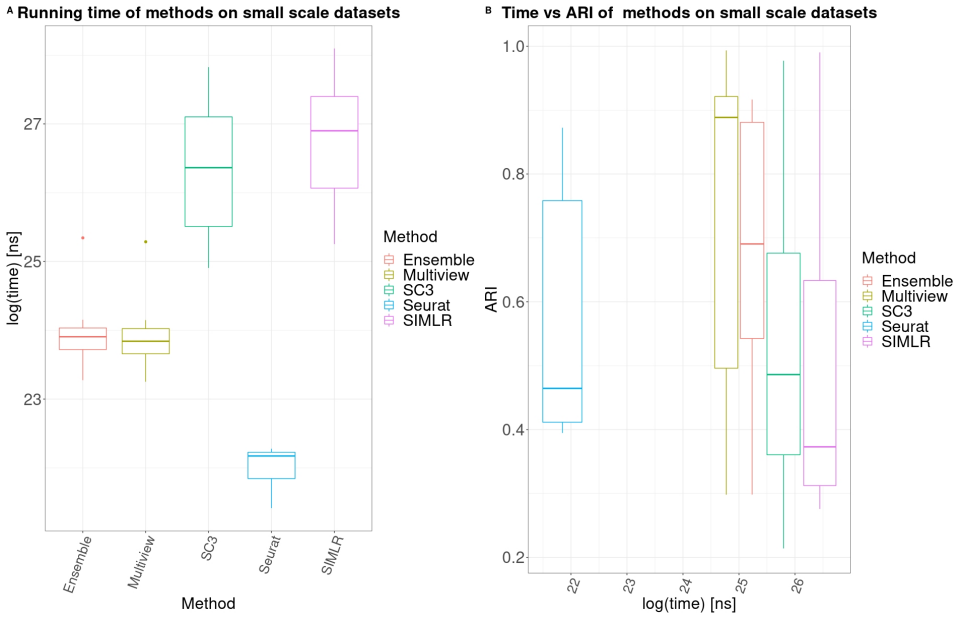

Figure 4.10: A) Box plot of running times of methods on the small scale datasets ($n = 7$). The x-axis is the method used and the y-axis natural log of the computational time in nano seconds. **B)**Box plot of running times of methods vs ARI score on the small scale datasets ($n = 7$). The x-axis is the method used and the y-axis ARI score.

4.6 Summary

To summarize, through simulation studies and real data, we showed that the multimodality testing is able to infer cluster structure accurately. This method also is

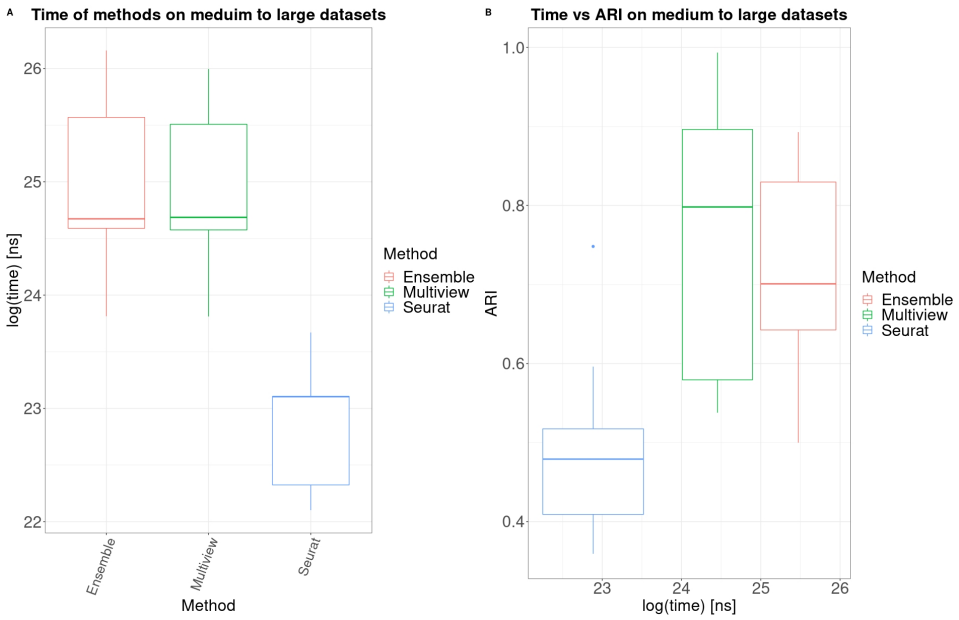

Figure 4.11: A) Box plot of running times of methods on the medium to large scale datasets ($n = 11$). The x-axis is the method used and the y-axis natural log of the computational time in nanoseconds. **B)**Box plot of running times of methods vs ARI score on the medium to large scale datasets ($n = 11$). The x-axis is the method used and the y-axis ARI score.

fast and sensitive to cluster size, separability, and data sparsity. We also showed that multimodality testing performs well on both the negative and positive control datasets. Further, we showed that for the CCMT procedure, both the multiview and the ensemble models works well for all the simulation studies, as measured by the ARI score. On real datasets, we compared both the computational time and ARI score of the CCMT procedure to well well known and often used methods. We showed that it is faster than some of the current methods and, on average, more accurate. Finally, to see the effects of the number of genes and PCS on the CCMT procedure, we performed an experiment where the number of genes and PCS were varied. The results showed that increasing the number of genes generally increased the overall performance of the CCMT procedure. However, the opposite is observed when increasing the number of PCS.

Chapter 5

Conclusions

5.1 Summary

In this thesis, we developed a method that assesses the cluster structure inherent to a dataset. We used multimodality testing coupled with three different data normalization and gene selection methods. The cosine distance was used to calculate the distribution of gene expression patterns between cells, and the Dip test was used to test this distribution for multimodality. The cosine distance was used because it is computationally efficient to compute. This distance metric also showed higher average accuracy and sensitivity than other distance metrics such as Euclidean and Manhattan on simulation studies and real data. Next, we used extensive simulation studies to show that this method is robust to the challenges inherent to scRNA-seq datasets. These challenges include high cluster overlap, high sparsity, an increasing number of clusters, and increasing data size. Using real datasets as positive and negative controls, we showed that this method performs as expected in the presence of clusters and the absence of cluster structure.

The second method developed in this thesis addressed finding the number of clusters and returning the clusters. To do this, we developed the CCMT procedure. The CCMT procedure couples multimodality testing with hierarchical clustering and discriminant analysis. The CCMT procedure assumes that clusters are derived from unimodal distribution. This assumption makes the CCMT procedure flexible enough to accommodate datasets with different distributions since many known

distributions have a unimodal variant. Using extensive simulations, we showed that the CCMT procedure is able to recover simulated clusters accurately. It is also sensitive to cluster overlap, meaning that it can detect clusters even when they are highly overlapping. Using real datasets as positive and negative controls, we demonstrated the CCMT procedure ability to accurately recover ground truth partitions. We separated the real datasets into two groups and used them to benchmark the CCMT procedure against other methods currently used to cluster scRNA-seq datasets. We showed that for both the small and medium to large scale datasets, the CCMT procedure is more accurate than other methods. The CCMT procedure is, however, slower than *Seurat*.

In the last part of the thesis, we attempted to understand the factors affecting multimodality testing and the CCMT procedure. We showed that the log normalization method is less sensitive than the other two when assessing cluster structure. For the CCMT procedure, we showed that increasing the number of highly informative genes used increases the ensemble model's overall performance. However, for the multiview model, the performance remains relatively constant. We also showed that for the CCMT procedure setting, the number of PCS constant for all datasets negatively affects both the ensemble and multiview models' performance. Lastly, we showed that modality testing and the CCMT procedure are limited in situations with increasing number of clusters and high cluster overlaps. However, these effect is more pronounced in situations where the clusters sizes are relatively balanced compared to the cases where sizes are unbalanced.

5.2 Discussion

Notably, for multimodality testing, the log normalization method proved to be less sensitive for the simulation studies. There could be a few reasons for this. Firstly, the other two normalization methods may be more similar to the simulation mechanism used in the *Splatter* package. The *Splatter* package uses a gamma-Poisson model to simulate molecular counts. Since both the negative binomial and the multinomial distributions can be estimated using a Poisson distribution, both of these normalization methods would perform better overall on simulation studies. However, this is not seen on real data both for the positive and negative control.

The data generating mechanism for the benchmarking datasets may not be completely Poisson. There may also be enough differences in gene expression patterns between cells in each of the benchmarking datasets that all the normalization methods can pick up, resulting in significant cluster structure evidence.

Multimodality testing is no stranger to the scRNA-seq domain. It has been used multiple times when analyzing scRNA-seq datasets. In [6], the authors used the Dip test to show the continuous nature of the distribution of T-cells activation states. The Dip test was also used in [35] to show that separation between cells could consistently be found when the cells are represented as a time series. However, to the best of our knowledge, this is the first time that multimodality testing has been used to the extent shown in this work.

We investigated the effect of an increased number of genes on the performance of the CCMT procedure. The results showed that increasing the number of genes positively impacted the ensemble model and had no significant impact on the multiview model. See Figure 4.8. This difference is partly due to how the clustering results are combined between the ensemble and the multiview model. For the ensemble model, a clustering solution is generated independently for each normalization method. Next, these clustering results are combined to form an ensemble. The multiview model combines the result from all three normalization methods before generating a clustering solution. By default, we select the top 500 most informative genes. The multiview model has an upper bound of 1500 on the total number of genes when clustering, which happens when there are no overlaps between the most informative genes across the normalization methods. The multiview model has a higher gene pool before clustering, so we see no significant performance increase. In contrast, the ensemble is limited to 500 genes in each normalization method. Thus increasing the number of genes improves each of the independent clustering solutions before the ensemble generation, which improves the ensemble clusters.

We also investigated the effect of an increased number of PCS used for clustering on the CCMT procedure performance. The results showed that increasing the number of PCS had a negative impact on both the ensemble and multiview models. See Figure 4.9. This decrease in performance is largely due to more PCS decreasing the signal to noise ratio in the dataset, which negatively impacts clustering. We

currently select the number of PCS for each dataset based on automatically finding the knee point on a PCA scree plot. This knee point is found by first sorting the PCS in decreasing order. Next, we find the point with the largest distance to the first and last scree plot points. Since each dataset has different characteristics and thus behaves differently, the knee point method is better able to take this in to account. The CCMT procedure results when using the knee point method are better when compared to holding PCS constant for every dataset.

The CCMT procedure heavily relies on the projection of two classes on a line that best separates their centers. The Fisher discriminant function that is used to compute the projections assumes that the classes are linearly separable. This means that a single straight line can separate both classes. A straight line may not always be the case capable of separating the classes. If the classes are not linearly separable, the projection may fail, causing the CCMT procedure to fail subsequently. We did not explore this topic in this work because there is currently no way to simulate scRNA-seq datasets with linearly not separable classes fully. The results for the CCMT procedure on benchmarking data justify our assumption that the clusters or cell types present can be separated using the Fisher linear discriminant function.

A critical decision when clustering scRNA-seq data is how many cell types to identify. There is generally no accepted way of choosing an adequate number of clusters or cell types when clustering scRNA-seq datasets. It depends on the resolution at which the user wants to view the dataset. If the user decides on a smaller number, this will result in identifying more distinct cell types. However, if the user decides on a larger number, this results in less distinct cell types. This work has served to provide some automated guidance on this issue. We hope this work will help relieve some of the headaches associated with deciding how many possible cell types are present in a dataset. The CCMT procedure developed in this work tends to underestimate the correct number of clusters. However, based on the ARI scores, it seems that even though the CCMT procedure underestimates the correct number of clusters, the clusters have a higher agreement with the ground truth.

5.3 Future Work

Much of this work depends on computing distances between cells. However, for relatively large datasets, this can become computationally expensive. A possible future direction for handling quite large datasets would be to use machine learning methods to reduce computational costs. For the CCMT procedure, it would be possible to use a subset of the dataset to train a model and then use it to predict the other cells. One approach would be to randomly sample the dataset and then run the multimodality testing algorithm on the random sample for assessing clusterability. If a large enough sample size is chosen, it should capture the general cluster structure present in the dataset.

Another avenue for future work is integrating a non-linear projection method. Currently, the Fisher discriminant coordinates projection method is used for projecting the classes. However, this is a linear projection method and may fail when the classes are not linearly separable. A kernelized version of the Fisher projection method can handle cases where there is no clear linear separation between the classes. A kernelized version of Fisher discriminant coordinates is discussed in [9, 67, 80]. Combining both the linear and the kernel versions of this projection method will make the CCMT procedure well rounded and more flexible in handling different datasets.

Bibliography

[1] M. Ackerman, A. Adolfsson, and N. Brownstein. An effective and efficient approach for clusterability evaluation, 2016. → page 29

[2] A. Adolfsson, M. Ackerman, and N. C. Brownstein. To cluster, or not to cluster: An analysis of clusterability methods. *Pattern Recognition*, 88: 13–26, Apr 2019. ISSN 0031-3203. doi:10.1016/j.patcog.2018.10.026. URL http://dx.doi.org/10.1016/j.patcog.2018.10.026. → pages 1, 7, 13, 29

[3] J. Ameijeiras-Alonso, R. Crujeiras, and A. Casal. Mode testing, critical bandwidth and excess mass. *TEST*, 09 2016. doi:10.1007/s11749-018-0611-5. → pages 15, 16, 29

[4] S. Anders and W. Huber. Differential expression analysis for sequence count data. *Nature Precedings*, 5, 04 2010. doi:10.1038/npre.2010.4282.2. → page 3

[5] T. S. Andrews, V. Yu. Kiselev, and M. Hemberg. *Statistical Methods for Single-Cell RNA-Sequencing*, chapter 26, pages 735–20. John Wiley Sons, Ltd, 2019. ISBN 9781119487845. doi:10.1002/9781119487845.ch26. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119487845.ch26. → page 1

[6] E. Azizi, A. Carr, G. Plitas, A. Cornish, C. Konopacki, S. Prabhakaran, J. Nainys, K. Wu, V. Kiseliovas, M. Setty, K. Choi, R. Fromme, P. Dao, P. McKenney, R. Wasti, K. Kadaveru, L. Mazutis, A. Rudensky, and D. Pe'er. Single-cell map of diverse immune phenotypes in the breast tumor microenvironment. *Cell*, 174, 06 2018. doi:10.1016/j.cell.2018.05.060. → page 69

[7] R. Bacher, L.-F. Chu, N. Leng, A. Gasch, J. Thomson, R. Stewart, M. Newton, and C. Kendziorski. Scnorm: robust normalization of

single-cell rna-seq data. *Nature Methods*, 14, 04 2017. doi:10.1038/nmeth.4263. → pages 1, 3

[8] M. Baron, A. Veres, S. Wolock, A. Faust, R. Gaujoux, A. Vetere, J. Ryu, B. Wagner, S. Shen-Orr, A. Klein, D. Melton, and I. Yanai. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell systems*, 3:346–360, 10 2016. doi:10.1016/j.cels.2016.08.011. → page 42

[9] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12:2385–404, 11 2000. doi:10.1162/089976600300014980. → page 71

[10] S. Ben-David, U. Luxburg, and D. Pál. A sober look at clustering stability. pages 5–19, 06 2006. doi:10.1007/11776420_4. → page 17

[11] V. Bergen, M. Lange, S. Peidli, F. Wolf, and F. Theis. Generalizing rna velocity to transient cell states through dynamical modeling. 10 2019. doi:10.1101/820936. → page 6

[12] J. Bezdek and R. Hathaway. Vat: A tool for visual assessment of (cluster) tendency. volume 3, pages 2225 – 2230, 02 2002. ISBN 0-7803-7278-6. doi:10.1109/IJCNN.2002.1007487. → page 11

[13] S. Bickel and T. Scheffer. Multi-view clustering. pages 19– 26, 12 2004. ISBN 0-7695-2142-8. doi:10.1109/ICDM.2004.10095. → pages 5, 34

[14] H.-H. Bock. On some significance tests in cluster analysis. *Journal of Classification*, 2:77–108, 12 1985. doi:10.1007/BF01908065. → page 18

[15] P. Brennecke, S. Anders, J. Kim, A. A. Kolodziejczyk, X. Zhang, V. Proserpio, B. Baying, V. Benes, S. Teichmann, J. Marioni, and M. Heisler. Accounting for technical noise in single-cell rna-seq experiments (vol 10, pg 1093, 2013). *Nature Methods*, 11:210–210, 02 2014. doi:10.1038/nmeth0214-210b. → page 27

[16] C. Burdziak, E. Azizi, S. Prabhakaran, and D. Pe'er. A nonparametric multi-view model for estimating cell type-specific gene regulatory networks. 02 2019. → page 5

[17] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 2018. ISSN 1087-0156.

doi:10.1038/nbt.4096. URL https://www.nature.com/articles/nbt.4096. → pages 4, 6, 25, 27, 43

[18] T. Caliński and H. JA. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3:1–27, 01 1974. doi:10.1080/03610927408827101. → page 17

[19] R. Cannoodt, W. Saelens, and S. Yvan. Computational methods for trajectory inference from single-cell transcriptomics. *European Journal of Immunology*, 46, 09 2016. doi:10.1002/eji.201646347. → page 6

[20] G. Chao, S. Sun, and J. Bi. A survey on multi-view clustering. 12 2017. → pages 5, 34

[21] R. Chen, X. Wu, L. Jiang, and Y. Zhang. Single-cell rna-seq reveals hypothalamic cell diversity. *Cell Reports*, 18:3227–3241, 03 2017. doi:10.1016/j.celrep.2017.03.004. → page 42

[22] M. Delmans and M. Hemberg. Discrete distributional differential expression (d3e) - a tool for gene expression analysis of single-cell rna-seq data. *BMC Bioinformatics*, 17, 12 2016. doi:10.1186/s12859-016-0944-6. → page 6

[23] P. Diggle. *The Statistical Analysis of Spatial Point Patterns*. 01 2003. → page 12

[24] J. Ding, A. Condon, and S. Shah. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature Communications*, 9, 12 2018. doi:10.1038/s41467-018-04368-5. → page 12

[25] D. duVerle, S. Yotsukura, S. Nomura, H. Aburatani, and K. Tsuda. Celltree: An r/bioconductor package to infer the hierarchical structure of cell populations from single-cell rna-seq data. *BMC Bioinformatics*, 17: 363, 09 2016. doi:10.1186/s12859-016-1175-6. → page 5

[26] A. Duò, M. Robinson, and C. Soneson. A systematic performance evaluation of clustering methods for single-cell rna-seq data. *F1000Research*, 7:1141, 09 2018. doi:10.12688/f1000research.15666.2. → page 27

[27] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. volume 96, pages 226–231, 01 1996. → page 4

[28] K. Etemad. Discriminant analysis for recognition of human. 11 2003. → page 31

[29] F. Faridafshin, B. Grechuk, and A. Naess. Calculating exceedance probabilities using a distributionally robust method. *Structural Safety*, 67: 132–141, 07 2017. doi:10.1016/j.strusafe.2017.02.003. → page 14

[30] G. Finak, A. McDavid, M. Yajima, J. Deng, V. Gersuk, A. Shalek, C. Slichter, H. Miller, M. McElrath, M. Prlic, P. Linsley, and R. Gottardo. Mast: A flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell rna sequencing data. *Genome Biology*, 16, 12 2015. doi:10.1186/s13059-015-0844-5. → page 6

[31] N. Fischer, E. Mammen, and J. Marron. Testing for multimodality. *Computational Statistics Data Analysis*, 18:499–512, 12 1994. doi:10.1016/0167-9473(94)90080-9. → page 13

[32] R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7:179–188, 01 1936. → page 31

[33] S. Freytag, L. Tian, I. Lönnstedt, M. Ng, and M. Bahlo. Comparison of clustering tools in r for medium-sized 10x genomics single-cell rna-sequencing data. *F1000Research*, 7:1297, 08 2018. doi:10.12688/f1000research.15809.1. → pages 42, 45

[34] T. Geddes, T. Kim, L. Nan, J. Burchfield, J. Yang, D. Tao, and P. Yang. Autoencoder-based cluster ensembles for single-cell rna-seq data analysis. 09 2019. doi:10.1101/773903. → page 4

[35] W. Gong, I.-Y. Kwak, N. Koyano-Nakagawa, W. Pan, and D. Garry. Tcm visualizes trajectories and cell populations from single cell data. *Nature Communications*, 9, 12 2018. doi:10.1038/s41467-018-05112-9. → page 69

[36] D. Grün, L. Kester, and A. Oudenaarden. Validation of noise models for single-cell transcriptomics. *Nature methods*, 11, 04 2014. doi:10.1038/nmeth.2930. → page 1

[37] D. Grün, M. Muraro, J.-C. Boisset, K. Wiebrands, A. Lyubimova, G. Dharmadhikari, M. van den Born, J. Es, E. Jansen, H. Clevers, E. de Koning, and A. Oudenaarden. De novo prediction of stem cell identity using single-cell transcriptome data. *Cell Stem Cell*, 19, 06 2016. doi:10.1016/j.stem.2016.05.010. → page 3

[38] M. Guo, H. Wang, S. Potter, J. Whitsett, and Y. Xu. Sincera: a pipeline for single-cell rna-seq profiling analysis. *PLOS Computational Biology*, 11: e1004575, 11 2015. doi:10.1371/journal.pcbi.1004575. → page 21

[39] C. Hafemeister and R. Satija. Normalization and variance stabilization of single-cell rna-seq data using regularized negative binomial regression. *Genome Biology*, 20, 12 2019. doi:10.1186/s13059-019-1874-1. → pages 25, 26, 27, 28

[40] A. Haque, J. Engel, S. Teichmann, and T. Lönnberg. A practical guide to single-cell rna-sequencing for biomedical research and clinical applications. *Genome Medicine*, 9, 12 2017. doi:10.1186/s13073-017-0467-4. → page 2

[41] J. Hartigan. Asymptotic distributions for clustering criteria. *Annals of Statistics*, 6, 01 1978. doi:10.1214/aos/1176344071. → page 18

[42] J. Hartigan and P. Hartigan. The dip test of unimodality. *The Annals of Statistics*, 13, 03 1985. doi:10.1214/aos/1176346577. → pages 15, 24, 29

[43] J. Hartigan and S. Mohanty. The runt test for multimodality. *Journal of Classification*, 9:63–70, 02 1992. doi:10.1007/BF02618468. → page 15

[44] E. Helgeson and E. Bair. Non-parametric cluster significance testing with reference to a unimodal null distribution. 10 2016. → pages 18, 20, 21

[45] S. Hicks, F. W. Townes, M. Teng, and R. Irizarry. Missing data and technical variability in single-cell rna-sequencing experiments. *Biostatistics (Oxford, England)*, 19, 11 2017. doi:10.1093/biostatistics/kxx053. → page 1

[46] S. Hicks, F. W. Townes, M. Teng, and R. Irizarry. Missing data and technical variability in single-cell rna-sequencing experiments. *Biostatistics (Oxford, England)*, 19, 11 2017. doi:10.1093/biostatistics/kxx053. → page 25

[47] B. HOPKINS and J. SKELLAM. A new method for detecting the type of distribution of plant individuals. *Annals of Botany*, 18, 04 1954. doi:10.1093/oxfordjournals.aob.a083391. → page 12

[48] H. Huang, Y. Liu, M. Yuan, and J. Marron. Statistical significance of clustering using soft thresholding. *Journal of Computational and Graphical Statistics*, 24, 05 2013. doi:10.1080/10618600.2014.948179. → page 19

[49] J. Huband, J. Bezdek, and R. Hathaway. Bigvat: Visual assessment of cluster tendency for large data sets. *Pattern Recognition*, 38:1875–1886, 11 2005. doi:10.1016/j.patcog.2005.03.018. → page 11

[50] R. Huh, Y. Yang, Y. Jiang, Y. Shen, and Y. Li. SAME-clustering: Single-cell Aggregated Clustering via Mixture Model Ensemble. *Nucleic Acids Research*, 48(1):86–95, 11 2019. ISSN 0305-1048. doi:10.1093/nar/gkz959. URL https://doi.org/10.1093/nar/gkz959. → page 4

[51] T. Ilicic, J. Kim, A. A. Kolodziejczyk, F. Bagger, D. McCarthy, J. Marioni, and S. Teichmann. Classification of low quality cells from single-cell rna-seq data. *Genome Biology*, 17, 12 2016. doi:10.1186/s13059-016-0888-1. → page 2

[52] A. Jain and R. Dubes. *Algorithms for Clustering Data*, volume 32. 01 1988. doi:10.2307/1268876. → pages 12, 13, 18

[53] Z. Ji and H. Ji. Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. *Nucleic Acids Research*, 44:gkw430, 05 2016. doi:10.1093/nar/gkw430. → pages 4, 6

[54] L. Jiang, H. Chen, L. Pinello, and G.-C. Yuan. Giniclust: Detecting rare cell types from single-cell gene expression data with gini index. *Genome biology*, 17:144, 07 2016. doi:10.1186/s13059-016-1010-4. → page 4

[55] I. Jolliffe and J. Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374:20150202, 04 2016. doi:10.1098/rsta.2015.0202. → pages 11, 12

[56] T. Kalisky and S. Quake. Single-cell genomics. *Nature methods*, 8:311–4, 04 2011. doi:10.1038/nmeth0411-311. → page 2

[57] P. Kharchenko, L. Silberstein, and D. Scadden. Bayesian approach to single-cell differential expression analysis. *Nature methods*, 11, 05 2014. doi:10.1038/nmeth.2967. → page 6

[58] P. Kimes, Y. Liu, D. Hayes, and J. Marron. Statistical significance for hierarchical clustering. *Biometrics*, 73, 11 2014. doi:10.1111/biom.12647. → pages 18, 19, 21

[59] V. Kiselev, K. Kirschner, M. Schaub, T. Andrews, A. Yiu, T. Chandra, K. Natarajan, W. Reik, M. Barahona, A. Green, and M. Hemberg. Sc3: consensus clustering of single-cell rna-seq data. 05 2017. doi:10.17863/CAM.9872. → pages 3, 5, 21, 44

[60] V. Kiselev, T. Andrews, and M. Hemberg. Publisher correction: Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20:1, 01 2019. doi:10.1038/s41576-019-0095-5. → page 2

[61] A. Klein, L. Mazutis, I. Akartuna, N. Tallapragada, A. Veres, V. Li, L. Peshkin, D. Weitz, and M. Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161:1187–1201, 05 2015. doi:10.1016/j.cell.2015.04.044. → page 42

[62] M. Krzak, Y. Raykov, A. Boukouvalas, L. Cutillo, and C. Angelini. Benchmark and parameter sensitivity analysis of single-cell rna sequencing clustering methods. *Frontiers in Genetics*, 10:1253, 12 2019. doi:10.3389/fgene.2019.01253. → page 35

[63] I. labs. *Whole CD45+ splenocytes from B6 mice*, 2019 (accessed 2020). URL https://singlecell.broadinstitute.org/single_cell/study/SCP306/whole-cd45-splenocytes-from-b6-mice-10x-hms#study-summarys. → page 42

[64] R. Lawson and P. Jurs. New index for clustering tendency and its application to chemical problems. *Journal of Chemical Information and Computer Sciences*, 30:36–41, 02 1990. doi:10.1021/ci00065a010. → page 12

[65] I. Lengyel and P. Derish. Ripley, b. d. 1981. spatial statistics. john wiley sons, new york. 09 2002. → page 12

[66] H. Li, E. Courtois, D. Sengupta, Y. Tan, K. Chen, J. Goh, S. Kong, C. Chua, L. Hon, W. S. Tan, M. Wong, P. Choi, L. Wee, A. Hillmer, I. Tan, P. Robson, and S. Prabhakar. Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors. *Nature Genetics*, 49, 03 2017. doi:10.1038/ng.3818. → page 42

[67] Y. Li, S. Gong, and H. Liddell. Recognising trajectories of facial identities using kernel discriminant analysis. *Image and Vision Computing*, 21: 1077–1086, 01 2004. doi:10.1016/j.imavis.2003.08.010. → page 71

[68] P. Lin, M. Troup, and J. Ho. Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome Biology*, 18, 12 2017. doi:10.1186/s13059-017-1188-0. → page 5

[69] Y. Liu, D. Hayes, A. Nobel, and J. Marron. Statistical significance of clustering for high-dimension, low-sample size data. *Journal of the American Statistical Association*, 103:1281–1293, 09 2008. doi:10.1198/016214508000000454. → page 19

[70] S. Lovie. *Exploratory Data Analysis*, volume 27. 03 2008. ISBN 9780470061572. doi:10.1002/9780470061572.eqr222. → page 11

[71] A. Lun, K. Bach, and J. Marioni. Pooling across cells to normalize single-cell rna sequencing data with many zero counts. *Genome Biology*, 17, 12 2016. doi:10.1186/s13059-016-0947-7. → pages 3, 25

[72] U. Luxburg. Clustering stability: An overview. *Found Trends Machine Learn*, 2, 07 2010. doi:10.1561/2200000008. → page 18

[73] U. Luxburg and S. Ben-David. Towards a statistical theory of clustering. *PASCAL Workshop on Statistics and Optimization of Clustering*, 01 2005. → page 7

[74] E. Marco, R. Karp, G. Guo, P. Robson, A. Hart, L. Trippa, and G.-C. Yuan. Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape. *Proceedings of the National Academy of Sciences of the United States of America*, 111, 12 2014. doi:10.1073/pnas.1408993111. → pages 3, 6

[75] C. Mayer, C. Hafemeister, R. Bandler, R. Machold, R. Batista-Brito, X. Jaglin, K. Allaway, A. Butler, G. Fishell, and R. Satija. Developmental diversification of cortical inhibitory interneurons. *Nature*, 555, 03 2018. doi:10.1038/nature25999. → page 27

[76] D. McCarthy, K. Campbell, A. Lun, and Q. Wills. Scater: Pre-processing, quality control, normalization and visualization of single-cell rna-seq data in r. *Bioinformatics (Oxford, England)*, 33, 01 2017. doi:10.1093/bioinformatics/btw777. → pages 2, 3, 6

[77] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. → pages 11, 12

[78] N. Meinshausen, L. Meier, and P. Bühlmann. P-values for high-dimensional regression, 2008. → page 20

[79] Z. Miao and X. Zhang. Desingle: A new method for single-cell differentially expressed genes detection and classification. *bioRxiv*, 2017. doi:10.1101/173997. URL https://www.biorxiv.org/content/early/2017/09/08/173997. → page 6

[80] S. Mika, G. Rätsch, J. Weston, B. Scholkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. volume 9, pages 41 – 48, 09 1999. ISBN 0-7803-5673-x. doi:10.1109/NNSP.1999.788121. → page 71

[81] M. Mokari, H. Mohammadzade, and B. Ghojogh. Recognizing involuntary actions from 3d skeleton data using body states. *Scientia Iranica*, 27: 1424–1436, 06 2020. doi:10.24200/sci.2018.20446. → page 31

[82] A. Mortazavi, B. Williams, K. Mccue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. *Nature methods*, 5: 621–8, 08 2008. doi:10.1038/nmeth.1226. → page 3

[83] M. Muraro, G. Dharmadhikari, D. Grün, N. Groen, T. Dielen, E. Jansen, L. Gurp, M. Engelse, F. Carlotti, E. de Koning, and A. Oudenaarden. A single-cell transcriptome atlas of the human pancreas. *Cell Systems*, 3, 09 2016. doi:10.1016/j.cels.2016.09.002. → page 42

[84] F. Perraudeau, D. Risso, K. Street, E. Purdom, and S. Dudoit. Bioconductor workflow for single-cell rna sequencing: Normalization, dimensionality reduction, clustering, and lineage inference. *F1000Research*, 6:1158, 07 2017. doi:10.12688/f1000research.12122.1. → page 2

[85] R. Petegrosso, Z. Li, and R. Kuang. Machine learning and statistical methods for clustering single-cell rna-sequencing data. *Briefings in bioinformatics*, 06 2019. doi:10.1093/bib/bbz063. → page 6

[86] E. Pierson and C. Yau. Zifa: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biology*, 16, 12 2015. doi:10.1186/s13059-015-0805-z. → pages 12, 26

[87] S. Prabhakaran, E. Azizi, A. Carr, and D. Pe'er. Dirichlet process mixture model for correcting technical variation in single-cell gene expression data. *Proc. 33nd Int. Conf. Mach. Learn., ICML 2016*, 48:1070–1079, 01 2016. → page 4

[88] X. Ren, L. Zheng, and Z. Zhang. Sscc: A novel computational framework for rapid and accurate clustering large-scale single cell rna-seq data. *Genomics, Proteomics Bioinformatics*, 17, 06 2019. doi:10.1016/j.gpb.2018.10.003. → page 4

[89] D. Risso, F. Perraudeau, S. Gribkova, S. Dudoit, and J.-P. Vert. A general and flexible method for signal extraction from single-cell rna-seq data. *Nature Communications*, 9, 12 2018. doi:10.1038/s41467-017-02554-5. → page 12

[90] M. Rodriguez, C. Comin, D. Casanova, O. Bruno, D. Amancio, F. Rodrigues, and L. da F. Costa. Clustering algorithms: A comparative approach. *PLOS ONE*, 14, 12 2016. doi:10.1371/journal.pone.0210236. → page 5

[91] R. Romanov, A. Zeisel, J. Bakker, F. Girach, A. Hellysaz, R. Tomer, A. Alpar, J. Mulder, F. Clotman, E. Keimpema, B. Hsueh, A. Crow, H. Martens, C. Schwindling, D. Calvigioni, J. Bains, Z. Máté, G. Szabo, Y. Yanagawa, and T. Harkany. Molecular interrogation of hypothalamic organization reveals distinct dopamine neuronal subtypes. *Nat Neurosci*, 20:176–188, 02 2017. doi:10.1038/nn.4462. → page 42

[92] P. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 11 1987. doi:10.1016/0377-0427\%2887\%2990125-7. → page 17

[93] G. Rozal and J. Hartigan. The map test for multimodality. *Journal of Classification*, 11:5–36, 02 1994. → page 15

[94] A.-A. Samadani, E. Kubica, R. Gorbet, and D. Kulic. Perception and generation of affective hand movements. *International Journal of Social Robotics*, 5, 01 2012. doi:10.1007/s12369-012-0169-4. → page 31

[95] W. Sarle and S. Institute. *Cubic Clustering Criterion*. SAS technical report. SAS Institute, 1983. URL https://books.google.ca/books?id=YynlGAAACAAJ. → page 18

[96] Segerstolpe, A. Palasantza, P. Eliasson, E.-M. Andersson, A.-C. Andréasson, X. Sun, S. Picelli, A. Sabirsh, M. Clausen, M. Bjursell, D. Smith, M. Kasper, C. Ammala, and R. Sandberg. Single-cell transcriptome profiling of human pancreatic islets in health and type 2

diabetes. *Cell Metabolism*, 24:1–15, 10 2016. doi:10.1016/j.cmet.2016.08.020. → page 42

[97] K. Shekhar, S. Lapan, I. Whitney, N. Tran, E. Macosko, M. Kowalczyk, X. Adiconis, J. Levin, J. Nemesh, M. Goldman, S. Mccarroll, C. Cepko, A. Regev, and J. Sanes. Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell*, 166:1308–1323.e30, 08 2016. doi:10.1016/j.cell.2016.07.054. → page 42

[98] Q. Shi, C. Zhang, M. Peng, X. Yu, T. Zeng, J. Liu, and L. Chen. Pattern fusion analysis by adaptive alignment of multiple heterogeneous omics data. *Bioinformatics (Oxford, England)*, 33, 05 2017. doi:10.1093/bioinformatics/btx176. → page 5

[99] S. Sieranoja. Fast and general density peaks clustering. *Pattern Recognition Letters*, 128, 10 2019. doi:10.1016/j.patrec.2019.10.019. → page 4

[100] B. Silverman. Using kernel density estimates to investigate multimodality. *J. Roy. Stat. Soc., Ser. B., Volume 43, p. 97-99*, 43, 09 1981. doi:10.1111/j.2517-6161.1981.tb01155.x. → pages 15, 16, 20, 29

[101] T. Smith, A. Heger, and I. Sudbery. Umi-tools: Modeling sequencing errors in unique molecular identifiers to improve quantification accuracy. *Genome Research*, 27:gr.209601.116, 01 2017. doi:10.1101/gr.209601.116. → page 2

[102] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 01 2002. doi:10.1162/153244303321897735. → page 18

[103] V. Svensson, R. Vento-Tormo, and S. Teichmann. Exponential scaling of single-cell rna-seq in the past decade. *Nature Protocols*, 13:599–604, 03 2018. doi:10.1038/nprot.2017.149. → page 2

[104] F. Tang, C. Barbacioru, Y. Wang, E. Nordman, C. Lee, N. Xu, X. Wang, J. Bodeau, B. Tuch, A. Siddiqui, K. Lao, and M. Surani. mrna-seq whole-transcriptome analysis of a single cell. *Nature methods*, 6:377–82, 05 2009. doi:10.1038/nmeth.1315. → page 2

[105] B. Tasic, V. Menon, T. N. Nguyen, S. Kim, T. Jarsky, Z. Yao, B. Levi, L. Gray, S. Sorensen, T. Dolbeare, D. Bertagnolli, J. Goldy, N. Shapovalova, S. Parry, C. Lee, K. Smith, A. Bernard, L. Madisen, S. Sunkin, and H. Zeng. Adult mouse cortical cell taxonomy revealed by

single cell transcriptomics. *Nature neuroscience*, 19, 01 2016. doi:10.1038/nn.4216. → page 42

[106] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society Series B*, 63:411–423, 02 2001. doi:10.1111/1467-9868.00293. → page 17

[107] F. W. Townes, S. Hicks, M. Aryee, and R. Irizarry. Feature selection and dimension reduction for single-cell rna-seq based on a multinomial model. *Genome Biology*, 20, 12 2019. doi:10.1186/s13059-019-1861-6. → pages 25, 26, 28

[108] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. Lennon, K. Livak, T. Mikkelsen, and J. Rinn. Pseudo-temporal ordering of individual cells reveals dynamics and regulators of cell fate decisions. *Nature biotechnology*, 32, 03 2014. doi:10.1038/nbt.2859. → pages 4, 6

[109] N. Trendafilov. Stepwise estimation of common principal components. *Comput. Stat. Data Anal.*, 54, 12 2010. doi:10.1016/j.csda.2010.03.010. → page 34

[110] P.-Y. Tung, J. Blischak, J. Hsiao, D. Knowles, J. Burnett, J. Pritchard, and Y. Gilad. Batch effects and the effective design of single-cell gene expression studies. *Scientific Reports*, 7, 01 2017. doi:10.1038/srep39921. → page 1

[111] K. Van den Berge, F. Perraudeau, C. Soneson, M. Love, D. Risso, J.-P. Vert, M. Robinson, S. Dudoit, and L. Clement. Observation weights unlock bulk rna-seq tools for zero inflation and single-cell applications. *Genome Biology*, 19, 12 2018. doi:10.1186/s13059-018-1406-4. → page 6

[112] L. van der Maaten and G. Hinton. Viualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 11 2008. → pages 11, 12

[113] B. Wang, D. Ramazzotti, L. De Sano, J. Zhu, E. Pierson, and S. Batzoglou. Simlr: A tool for large-scale genomic analyses by multi-kernel learning. *PROTEOMICS*, 18, 03 2017. doi:10.1002/pmic.201700232. → pages 4, 21, 22, 44

[114] J. Wang, J. Schug, K. J. Won, C. Liu, A. Naji, D. Avrahami, M. Golson, and K. Kaestner. Single-cell transcriptomics of the human endocrine pancreas. *Diabetes*, 65:db160405, 06 2016. doi:10.2337/db16-0405. → page 42

[115] L. Wang, U. Nguyen, J. Bezdek, C. Leckie, and K. Ramamohanarao. ivat and avat: Enhanced visual analysis for cluster tendency assessment. volume 6118, pages 16–27, 06 2010. doi:10.1007/978-3-642-13657-3_5. → page 11

[116] F. Wolf, P. Angerer, and F. Theis. Scanpy: Large-scale single-cell gene expression data analysis. *Genome Biology*, 19, 12 2018. doi:10.1186/s13059-017-1382-0. → page 4

[117] Z. Wu and H. Wu. Accounting for cell type hierarchy in evaluating single cell rna-seq clustering. *Genome Biology*, 21, 12 2020. doi:10.1186/s13059-020-02027-x. → page 5

[118] Y. Xin, J. Kim, H. Okamoto, M. Ni, Y. Wei, C. Adler, A. Murphy, G. Yancopoulos, C. Lin, and J. Gromada. Rna sequencing of single human islet cells reveals type 2 diabetes genes. *Cell metabolism*, 24, 09 2016. doi:10.1016/j.cmet.2016.08.018. → page 42

[119] D. Xu and Y. Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2, 08 2015. doi:10.1007/s40745-015-0040-1. → page 16

[120] S. Xu, X. Qiao, L. Zhu, Y. Zhang, C. Xue, and L. Li. Reviews on determining the number of clusters. *Applied Mathematics Information Sciences*, 10:1493–1512, 07 2016. doi:10.18576/amis/100428. → page 17

[121] L. Yang, J. Liu, Q. Lu, A. Riggs, and X. Wu. Saic: An iterative clustering approach for analysis of single cell rna-seq data. *BMC Genomics*, 18, 10 2017. doi:10.1186/s12864-017-4019-5. → page 3

[122] F. Ye, Z. Chen, H. Qian, R. Li, C. Chen, and Z. Zheng. *New Approaches in Multi-View Clustering*. 08 2018. ISBN 978-1-78923-526-5. doi:10.5772/intechopen.75598. → pages 5, 34

[123] L. Zappia, B. Phipson, and A. Oshlack. Splatter: Simulation of single-cell rna sequencing data. *Genome Biology*, 18, 12 2017. doi:10.1186/s13059-017-1305-0. → page 34

[124] A. Zeisel, A. Manchado, S. Codeluppi, P. Lonnerberg, G. La Manno, A. Juréus, S. Marques, H. Munguba, L. He, C. Betsholtz, C. Rolny, G. Castelo-Branco, J. Hjerling Leffler, and S. Linnarsson. Brain structure. cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347, 03 2015. doi:10.1126/science.aaa1934. → page 42

[125] J. Zhang, J. Fan, H. Fan, D. Rosenfeld, and D. Tse. An interpretable framework for clustering single-cell rna-seq datasets. *BMC Bioinformatics*, 19, 12 2018. doi:10.1186/s12859-018-2092-7. → page 5

[126] G. Zheng, J. Terry, P. Belgrader, P. Ryvkin, Z. Bent, R. Wilson, S. Ziraldo, T. Wheeler, G. McDermott, J. Zhu, M. Gregory, J. Shuga, L. Montesclaros, J. Underwood, D. Masquelier, S. Nishimura, M. Schnall-Levin, P. Wyatt, C. Hindson, and J. Bielas. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8:14049, 01 2017. doi:10.1038/ncomms14049. → page 42

[127] X. Zhu, J. Zhang, Y. Xu, J. Wang, X. Peng, and H.-D. Li. Single-cell clustering based on shared nearest neighbor and graph partitioning. *Interdisciplinary Sciences: Computational Life Sciences*, 12, 02 2020. doi:10.1007/s12539-019-00357-4. → pages 4, 21

[128] J. Žurauskienė and C. Yau. pcareduce: Hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, 17, 03 2016. doi:10.1186/s12859-016-0984-y. → page 3