Computational methods for large-scale single-cell RNA-seq and multimodal data

Văn Hoàn Đỗ

München 2021

Dissertation zur Erlangung des Doktorgrades
der Fakultät für Chemie und Pharmazie
der Ludwig–Maximilians–Universität München

Computational methods for large-scale single-cell RNA-seq and multimodal data

Văn Hoàn Đỗ

aus

Vinh Phuc, Vietnam

2021

Erklärung:
Diese Dissertation wurde im Sinne von §7 der Promotionsordnung vom 28. November 2011 von Herrn Dr. Stefan Canzar betreut.

Eidesstattliche Versicherung:
Diese Dissertation wurde eigenständig und ohne unerlaubte Hilfe erarbeitet.

München, 01.09.2021

Văn Hoàn Đỗ̃

Dissertation eingereicht am 08.09.2021
1. Gutachter: Dr. Stefan Canzar
2. Gutachter: Prof. Dr. Caroline Friedel
Mündliche Prüfung am 09.11.2021

Contents

Acknowledgments

First and foremost I offer my sincerest gratitude to my supervisor Dr. Stefan Canzar, who has supported me throughout my doctoral journey with his expertise and knowledge whilst providing me an opportunity to work in many exciting projects at Gene Center Munich. Dr. Canzar has been extremely supportive and has given me the freedom to pursue various projects. I will be forever grateful for his scientific guidance and the time he spent to discuss our work. I had a wonderful time as a doctoral student because of him and the lab he established.

I also would like to express my big thanks to all the previous and current members of the Canzar lab, especially Francisca, Parastou, Pablo, Shounak and Israa. I really miss our joyful lunch breaks and monthly get-together. I am also grateful to the Graduate School of Quantitative Biosciences Munich and Gene Center Munich for financial support and creating the environment to connect with other scientists and PhD students.

I will forever be thankful to my former advisors and professors for their guidance of my journey to science. In particular, I would like to thank Prof. Moshe Rosenfeld for enthusiasm and inspiration in mathematics. You are an amazing professor, and I want to let you know that you made a meaningful impact on my life and remain my best role model for a scientist and teacher. I also thank Dr. Nam-Dung Hoang and Dr. Hoang Duc Luu. Their lessons and guidance have shaped my way of doing scientific research. I am especially grateful to Prof. Dong Yen Nguyen for his recommendation letter. I truly admire him for his profession and personal life.

Last but not least, I would like to say a heartfelt thank you to my family. I could not achieve this without their endless support and encouragement from them. A special thanks goes to my dear wife, Thi Hong Tham Nguyen. Thank you for always being by my side and taking care of our kids, and I am sure that we will build a sweet home full of happiness and joys.

Summary

Emerging single cell genomics technologies such as single cell RNA-seq (scRNA-seq) and single cell ATAC-seq provide new opportunities for discovery of previously unknown cell types, facilitating the study of biological processes such as tumor progression, and delineating molecular mechanism differences between species. Due to the high dimensionality of the data produced by the technologies, computation and mathematics have been the cornerstone in decoding meaningful information from the data. Computational models have been challenged by the exponential growth of the data thanks to the continuing decrease in sequencing costs and growth of large-scale genomic projects such as the Human Cell Atlas. In addition, recent single-cell technologies have enabled us to measure multiple modalities such as transcriptome, protome, and epigenome in the same cell. This requires us to establish new computational methods which can cope with multiple layers of the data. To address these challenges, the main goal of this thesis was to develop computational methods and mathematical models for analyzing large-scale scRNA-seq and multimodal omics data. In particular, I have focused on fundamental single-cell analysis such as clustering and visualization.

The most common task in scRNA-seq data analysis is the identification of cell types. Numerous methods have been proposed for this problem with a current focus on methods for the analysis of large scale scRNA-seq data. I developed Specter, a computational method that utilizes recent algorithmic advances in fast spectral clustering and ensemble learning. Specter achieves a substantial improvement in accuracy over existing methods and identifies rare cell types with high sensitivity. Specter allows us to process a dataset comprising 2 million cells in just 26 minutes. Moreover, the analysis of CITE-seq data, that simultaneously provides gene expression and protein levels, showed that Specter is able to incorporate multimodal omics measurements to resolve subtle transcriptomic differences between subpopulations of cells.

We have effectively handled big data for clustering analysis using Specter. The question is how to cope with the big data for other downstream analyses such as trajectory inference and data integration. The most simple scheme is to shrink the data by selecting a subset of cells (the sketch) that best represents the full data set. Therefore I developed an algorithm called Sphetcher that makes use of the thresholding technique to efficiently pick representative cells that evenly cover the transcriptomic space occupied by the original data set. I showed that the sketch computed by Sphetcher constitutes a more accurate presentation of the original transcriptomic landscape than existing methods, which leads to a more balanced composition of cell types and a large fraction of rare cell types in the sketch. Sphetcher bridges the gap between the scalability of computational methods and the volume of the data. Moreover, I demonstrated that Sphetcher can incorporate prior information (e.g. cell labels) to inform the inference of the trajectory of human skeletal muscle myoblast differentiation.

The biological processes such as development, differentiation, and cell cycle can be mon-

itored by performing single cell sequencing at different time points, each corresponding to a snapshot of the process. A class of computational methods called trajectory inference aims to reconstruct the developmental trajectories from these snapshots. Trajectory inference (TI) methods such as Monocle, can computationally infer a pseudotime variable which serves as a proxy for developmental time. In order to compare two trajectories inferred by TI methods, we need to align the pseudotime between two trajectories. Current methods for aligning trajectories are based on the concept of dynamic time warping, which is limited to simple linear trajectories. Since complex trajectories are common in developmental processes, I adopted arboreal matchings to compare and align complex trajectories with multiple branch points diverting cells into alternative fates. Arboreal matchings were originally proposed in the context of phylogenetic trees and I theoretically linked them to dynamic time warping. A suite of exact and heuristic algorithms for aligning complex trajectories was implemented in a software Trajan. When aligning single-cell trajectories describing human muscle differentiation and myogenic reprogramming, Trajan automatically identifies the core paths from which we are able to reproduce recently reported barriers to reprogramming. In a perturbation experiment, I showed that Trajan correctly maps identical cells in a global view of trajectories, as opposed to a pairwise application of dynamic time warping.

Visualization using dimensionality reduction techniques such as t-SNE and UMAP is a fundamental step in the analysis of high-dimensional data. Visualization has played a pivotal role in discovering the dynamic trends in single cell genomics data. I developed j-SNE and j-UMAP as their generalizations to the joint visualization of multimodal omics data, e.g., CITE-seq data. The approach automatically learns the relative importance of each modality in order to obtain a concise representation of the data. When comparing with the conventional approaches, I demonstrated that j-SNE and j-UMAP produce unified embeddings that better agree with known cell types and that harmonize RNA and protein velocity landscapes.

Chapter 1

Introduction

The cell is the basic unit of all living organisms. Cells provide structure for the body, take and convert the nutrients from food to energy, and carry out specialized functions. The cells in humans and in many other organisms come in many different shapes and sizes that we can categorize into different types which carry different functions. For example, muscle cells form muscle tissue enable all bodily movement. Nerve cells are the basic unit of the nervous system, which send signals between the brain and other body organs. Despite their differences, they all have the same genome (DNA) which contains the information needed to build the entire body. A DNA molecule is divided up into functional units called genes, which are templates to make proteins through a process "central dogma of molecular biology". Proteins make up body structures as well as control chemical reactions and carry signals between cells.

The central dogma of molecular biology involves two steps: transcription and translation. In transcription, the DNA sequence of a gene is copied into an RNA molecule. In eukaryotes, this pre-messenger RNA (mRNA) will be further processed into a mature RNA, which is in turn translated into amino-acid sequences. This process of going from a gene to a functional product is known as gene expression (Crick, 1970). A gene can therefore be considered "on" if it is transcribed into RNA and only a subset of the genes in a cell are turned on at any one time. The variety of gene expression profiles seems to be the most relevant answers to the question of cell types and development. In fact, as the body develops, different sets of cells within these organisms turn specific combinations of genes on and off. Such developmental patterns are responsible for the variety of cell types in the mature organism.

Measuring gene expression level is an important problem in molecular biology. Different techniques are used to quantify gene expression level. In the past, traditional hybridization-based approaches such as microarrays allowed to measure gene expression of a single transcript at a time. More recently, high throughput methods such as bulk RNA-seq can measure tens of thousands of expressed genes and allow unbiased study of gene expression in a tissue. A fundamental research aim in many RNA-seq studies is to identify differentially expressed genes between different groups or conditions. Additionally, gene expression profiling enables us to detect allele specific expression and gene fusion events (Conesa et al., 2016).

Since bulk RNA-seq only measures the average expression level for each gene across a large population of cells in the tissue, it is not sufficient for studying heterogeneous systems such as complex tissues like the brain. To overcome the limitations of bulk RNA-seq, single-cell RNA sequencing (scRNA-seq) was developed to measure transcriptomic profiles of

the individual cell, which provides a powerful tool in decoding the heterogeneity in complex tissues as well as reconstruction of developmental processes. For example, scRNA-seq was applied to understand the complex subpopulations in healthy tissues such as lung (Treutlein et al., 2014) and brain (Pollen et al., 2014) as well as in human diseases such as breast cancer (Nguyen et al., 2018) and lung cancer (Guo et al., 2018; Stewart et al., 2020). In addition, other studies have used scRNA-seq to reconstruct novel dynamics in developmental processes within embryogenesis (Yan et al., 2013; Biase et al., 2014), hematopoiesis (Nestorowa et al., 2016) and neurogenesis (La Manno et al., 2016). Importantly, scRNA-seq also provides mechanistic insights in gene regulatory network inference (Aibar et al., 2017) and cell to cell interactions (Armingol et al., 2021).

1.1 Computational analysis of single-cell sequencing data

To enable transcriptomic profiling at a single-cell resolution, a number of high-throughput single-cell RNA-sequencing (scRNA-seq) protocols and technologies have been developed. In brief, a typical scRNA-seq experimental workflow begins with the dissociation of cells from a tissue and the isolation of single-cells with specific devices. The isolation step is performed differently depending on the protocol; the main approaches include isolation of cells on a plate, or capturing each cell in the microfluidic droplet. In the next step, mRNAs are captured for reverse transcription to generate complementary DNA (cDNA). Finally, cDNA will be amplified and undergo library preparation for sequencing. The data generated by a sequencing machine are processed to obtain an expression matrix, which further undergoes preprocessing steps such as quality control and normalization to obtain a final expression matrix. For more details we refer the reader to the papers (Svensson et al., 2017; Ziegenhain et al., 2017).

The computational analysis of scRNA-seq typically starts with an expression matrix. Conventionally, each row of the matrix corresponds to a gene and each column represents a cell. Each entry in the expression matrix represents the expression level of a gene in a given cell. The exact nature of scRNA-seq analysis depends on the biological questions at hand when performing the experiments. Despite these differences, there are a number of common downstream computational analyses that can be applied (see Figure 1.1). Here, we highlight core computational data analyses present in most of single-cell studies. For in depth review of scRNA-seq data analysis, we refer the readers to Luecken and Theis (2019).

Clustering

The most fundamental step in the scRNA-seq data analysis is to assign cell types to the cells because cell type information is not captured in scRNA-seq data. The process of labeling the data is typically done by clustering the cells based on a gene expression matrix and annotating clusters by the identity of upregulated genes associated with each cluster. The problem of grouping data into similar patterns has been studied in anthropology (Driver and Kroeber, 1932) and psychology (Zubin, 1938) almost a century ago and since then this so-called cluster analysis has become one of the most well-studied problems in unsupervised machine learning. In scRNA-seq data, we identify groups of cells based on the similarities of the gene expression profiles without knowing the prior labels. Expression profile similarity is determined via distance metrics, which often take dimensionality reduced representations

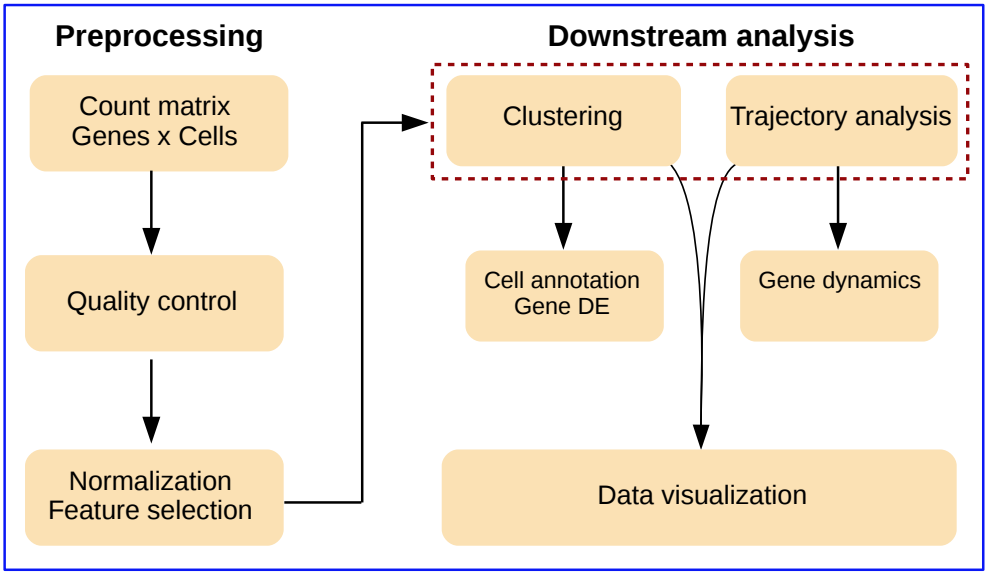

Figure 1.1: Common workflow in scRNA-seq data analysis. The workflow starts with an expression matrix, followed by preprocessing steps, which include the selection and filtration of cells based on quality control metrics, data normalization, and the selection of highly variable genes. The processed data then undergoes downstream analysis. The core analyses are the identification of cell types using clustering; reconstruction of cellular dynamic process using trajectory inference and RNA velocity; and visualization of cells in 2D using t-SNE and UMAP.

as input. The clustering algorithm which produces a partition of the data plays the most important role in success of cell type annotation. Various methods have been developed to perform clustering of scRNA-seq data, among them Seurat (Satija et al., 2015) is the most widely used. Single-cell clustering has been used in characterizing cell types in complex tissues such as the brain (Zeisel et al., 2018) as well as identification of new cell types (Cao et al., 2017; Fincher et al., 2018).

The gene signatures associated with each group are called marker genes. Marker genes are often found by performing differential expression tests between two groups: the target group and the remaining cells in the data set. Typically we are interested in genes that are upregulated in the cluster of interest. The Wilcoxon rank sum or the t-test are commonly used to rank genes based on the difference between the two groups. Clusters can be labeled by prior knowledge of the genes which determine the cell type or by comparing marker genes from the query data set and marker genes from a reference data set. The latter approach is possible thanks to many recent projects which aim to create reference single-cell atlas, e.g., the Human Cell Atlas (Regev et al., 2017).

Trajectory analysis and RNA velocity analysis

One area where scRNA-seq has been very successful is in the study of cellular dynamic processes such as development, differentiation, and cell cycle. The human body develops from a single cell, the fertilized egg, into a complex multicellular organism over time. During development cells differentiate into more specialized cell types. This process of differentiation can be monitored by taking snapshots of gene expression profiles at different time points.

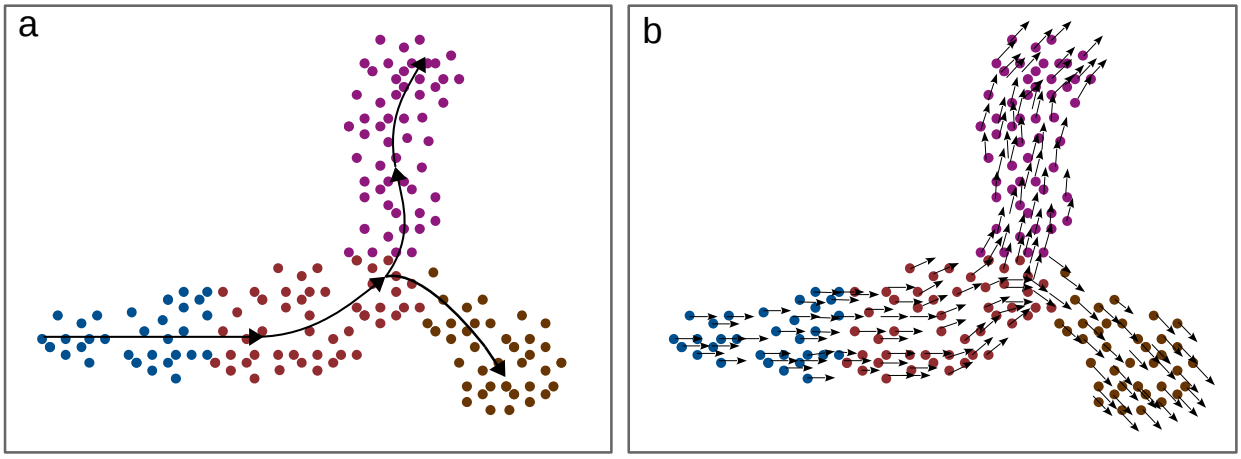

Figure 1.2: Trajectory inference (a) and RNA velocity (b) of the same process. Cells are colored by cell types.

The class of methods, which aims to capture transitions between cell types and cell lineages, is known as trajectory inference (TI). TI first identifies a trajectory which represents the underlying cellular dynamic process. Depending on the biological process, the constructed trajectories can be linear, bifurcating or complex structures such as trees and cycles (see Figure 1.2-a for an example). The order of cells along these trajectories is called a pseudotime variable, which is interpreted as a proxy for developmental time. TI provides a powerful tool in delineation of a differentiation tree as well as studying the changes of gene expression during a process. For example, Monocle (Trapnell et al., 2014; Qiu et al., 2017; Cao et al., 2019), the most widely used TI method, has been used to investigate a large variety of biological processes, including investigation of gene expression dynamics occurring during progenitor maturation towards an oligodendrocyte fate (Qiu et al., 2017), transcriptional dynamics of mouse organogenesis (Cao et al., 2019), the development of the human prefrontal cortex (Zhong et al., 2018), and the differentiation trajectory of breast cancer T cells (Savas et al., 2018). Besides Monocle, there are more than 70 computational methods for trajectory inference from single-cell transcriptomics. Among them 45 methods have been evaluated in a benchmarking paper by Saelens et al. (2019). Based on the benchmark results, they concluded that Slingshot (Street et al., 2018), TSCAN (Ji and Ji, 2016) and Monocle DDRTree (Qiu et al., 2017) are the top performing methods.

RNA velocity provides an alternative way to model dynamic process of cells (La Manno et al., 2018; Bergen et al., 2020). Instead of constructing a trajectory for the process, RNA velocity - the time derivative of the gene expression state - infers the future state of cells, i.e., predicts gene expression of individual cells on a timescale of hours. From this we can visualize the kinetic state of all cells in low-dimensional representations of the cell populations (see Figure 1.2-b). RNA velocity has been used to identify various branching lineages of the developing mouse hippocampus and to study the kinetics of transcription in the human embryonic brain. Furthermore, directionality from the RNA velocity has allowed for the identification of the root of the lineage tree of the hippocampus (La Manno et al., 2018). RNA velocity can infer putative driver genes of the dynamic process, providing an alternative to the standard differential expression analysis (Bergen et al., 2020). Although TI and RNA velocity provide complementary approaches to study single-cell dynamics, Zhang and Zhang

(2021) proposed CellPath, a trajectory inference method that infers a trajectory from the RNA velocity.

Visualization

The dimensionality of scRNA-seq data is large with typically at least 20,000 features (variables). Since humans are visual learners, we are interested in presenting the data in 2D or 3D plots that can capture both the overall shape and the fine granular structure of the data. Currently, t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten and Hinton, 2008) and UMAP (McInnes et al., 2018) have become standard tools for single-cell visualization since they preserve structure of data well and often produce embeddings which are consistent with cell type labels. t-SNE and UMAP aim to preserve the similarities between cells in the low embedding. In other words, if two cells are close in the original space, they must be also close in the low dimensional embedding, and vice versa. t-SNE and UMAP have become a cornerstone of scRNA-seq analysis. For example, the low dimensional plot is often colored by the expression levels of a gene of interest, which is useful in exploring the gene expression pattern over cell populations (e.g., found by clustering). Besides, RNA velocity overlays the vector field (direction of cell to its future state) on the 2D plots computed by t-SNE or UMAP, which allows us to discover the dynamic trends in single-cell data.

1.2 Thesis overview and contributions

Thanks to the continuing decrease in sequencing costs and growth of large-scale genomic projects such as the Human Cell Atlas, many scRNA-seq datasets have been generated with a recent dataset measuring the gene expression of over 4 million single-cells (Cao et al., 2020). Hence, current computational methods designed to handle hundreds to thousands of cells will need to scale to millions to match the pace of data generation. The exponential growth in the data is one of the biggest challenges in single-cell data analysis (Svensson et al., 2018). Moreover, we are able to generate scRNA-seq data sets of many different processes in the same individuals as well as in different species. This allows us to do comparative analysis between two data sets to pinpoint the differences and similarities between the two. For example, comparison between two single-cell trajectories computed by Monocle revealed molecular determinants of myogenic reprogramming outcome (Cacchiarelli et al., 2018). Currently, developed methods for the comparison of trajectories are restricted to simple, linear trajectories. A new approach needs to be developed to deal with a more realistic scenario, i.e., complex trajectories containing branching points that divert cells into different fates. Finally, along with improving throughput in single-cell experiments, recent technological innovations allow us to simultaneously measure different modalities (e.g., mRNA, protein level, chromatin accessibility) in the same cell (Cao et al., 2018; Stoeckius et al., 2017; Zhu et al., 2020). Many computational methods developed for unimodal single-cell data (e.g., scRNA-seq) are not applicable to multimodal data. Therefore, a new set of methods and techniques need to be developed to cope with multiple facets of the data. It is worth to mention that the aforementioned analyses (clustering, trajectory inference, visualization) are also fundamental for the analysis of other unimodal single cell genomics data (e.g., scATAC-seq) as well as multimodal omics data. This thesis aims to address some of these challenges in several major computational analyses including clustering and visualization.

Clustering of ultra-large scRNA-seq and multimodal data using Specter

One of the most fundamental computational tasks in the context of scRNA-seq analysis is the identification of groups of cells that are similar in their expression patterns, i.e. their transcriptomes, and which are at the same time distinct from other cells. Numerous methods have been proposed for clustering scRNA-seq data sets (Duò et al., 2018; Tian et al., 2019), with Seurat (Satija et al., 2015) and its underlying Louvain clustering algorithm (Blondel et al., 2008) being arguably the most widely used one. More recently, attempts have been made to design algorithms for the analysis of ultra-large scRNA-seq data sets, owing to the ever-increasing throughput of droplet-based sequencing technologies that allow to profile genome-wide expression for hundreds of thousands of cells at once. To address these challenges, we introduce Specter, a clustering method that adopts and extends recent algorithmic advances in fast spectral clustering. We adopt the idea of landmarks that are used to create a sparse representation of the full data from which a spectral embedding can then be computed in linear time. We exploit Specter's speed in a cluster ensemble scheme that achieves a substantial improvement in accuracy over existing methods and that is sensitive to rare cell types. Its linear time complexity allows Specter to scale to millions of cells and leads to fast computation times in practice. Furthermore, on CITE-seq data that simultaneously measures gene and protein marker expression we demonstrate that Specter is able to utilize multimodal omics measurements to resolve subtle transcriptomic differences between subpopulations of cells. The details are presented in Chapter 2, which is based on the publication: Van Hoan Do, Francisca Rojas Ringeling, and Stefan Canzar. *Linear-time cluster ensembles of large-scale single-cell RNA-seq and multimodal data*. Genome Research, 31(4):677-688, 2021.

Data summarization using Sphetcher

Sampling provides a more general framework to deal with big data. Although sampling might overlook parts of the data, this problem can be alleviated if a subset of cells is carefully chosen. The most common subsampling strategy is random sampling, however, it ignores the gene expression patterns of single-cells and thus risks overlooking rare cell states. Spatial random sampling (SRS) (Rahmani and Atia, 2017) and k-means++ (Arthur and Vassilvitskii, 2007) on the other hand, take into account the structure of the data when sampling the data. Experiments performed in Hie et al. (2019b), however, demonstrated that these data-dependent methods do not scale efficiently to large datasets and provide unbalanced samples that hamper downstream analyses. Hie et al. (2019b) introduced geometric sketching as an alternative approach that efficiently samples cells evenly across gene expression space rather than proportional to the abundance of cells that are in a similar state. For purely computational reasons, however, Hie et al. (2019b) approximate the transcriptomic space of single-cells by equal-sized boxes rather than spheres, within which cells are randomly selected as representatives into the sketch. Here, we propose an algorithm Sphetcher that makes use of the thresholding technique originally proposed for the design of approximation algorithms for bottleneck problems to efficiently pick representative cells within spheres of a fixed size into a spherical sketch of different metric spaces. We provide theoretical guarantees for the spherical sketch computed by Sphetcher and demonstrate through experiments on 6 single-cell datasets that these theoretical guarantees are indeed reflected in a more accurate representation of the original transcriptomic space which in turn benefits downstream anal-

yses such as clustering, and which allowed us to detect a rare population of inflammatory macrophages. Furthermore, our optimization scheme naturally allows to include fairness aspects that require to include cells of each pre-defined category which can encode prior biological or experimental knowledge such as cell type or collection time point. We demonstrate how our fairness-inspired model can help to incorporate the collection time point of cells in a time series experiment into the reconstruction of their developmental trajectory. Carefully combined with a prior grid sampling strategy that is orders of magnitude faster than geometric sketching, Sphetcher requires only 16 minutes to compute a sketch for a mouse embryonic dataset comprising 2 million cells. The details are presented in Chapter 3. The contents of this chapter are based on the publication: Van Hoan Do, Khaled Elbassioni, and Stefan Canzar. *Sphetcher: Spherical thresholding improves sketching of single-cell transcriptomic heterogeneity.* iScience, 23(6):101126, 2020.

Alignment of single-cell trajectories using Trajan

Single-cell RNA-seq has enabled the reconstruction of cellular lineages of biological processes such as differentiation, development and cell reprogramming. The trajectory may be used to infer the dynamic changes in gene expression along a pseudotime axis. Much can be learned from the comparative analysis of single-cell trajectories. Comparing gene expression dynamics along trajectories from two conditions can aid in elucidating the key differences between them and the regulatory programs underpinning the process. For example, comparing the trajectories underlying a given differentiation process in two species would shed light onto the evolutionary differences between these organisms. Recently, methods have been developed for this purpose, which make use of dynamic time warping (dtw). Dynamic time warping is a class of algorithms for comparing two time series that advance at different speeds. Similar to a pairwise sequence alignment that allows for insertions and deletions, dtw finds a mapping (warping) between similar elements in the two sequences to overcome locally stretched and compressed sections. In single-cell trajectories, cells are ordered along pseudo-time and can be aligned based on the expression values of (a subset of) their genes to establish a common pseudotime axis along which expression kinetics become comparable between different conditions.

Dynamic time warping can only compare two time series at a time, and thus current methods for comparing single-cell trajectories are limited to linear trajectories or rely on picking the correct path from a complex trajectory. Complex cell trajectories are common in developmental processes and also arise in response to genetic perturbations (Qiu et al., 2017). In these cases, prior information such as a set of defined markers would be necessary to pick the most relevant path, but this information is often not available. Another potential caveat of dtw is that it ignores cells that lie on alternative paths and could potentially amplify the signal used to infer the mapping between trajectories.

We present Trajan, a novel method to compare and align complex trajectories with multiple branch points diverting cells into alternative fates. Trajan automatically identifies the correspondence between biological processes in two trajectories and aligns all of them simultaneously, taking into account their overlap. Given that cells that are diverted into different fates share a common ancestry, they cannot be treated as independent from each other. Trajan adopts arboreal matchings (Böcker et al., 2013) to capture globally consistent similarities between trajectories. Arboreal matchings were originally proposed in the context of phylogenetic trees and here we theoretically link them to dynamic time warping.

When aligning single-cell trajectories describing human muscle differentiation and myogenic reprogramming, Trajan automatically identifies the core paths from which we are able to reproduce recently reported barriers to reprogramming. In a perturbation experiment, Trajan correctly maps identical cells in a global view of trajectories, as opposed to a pairwise application of dtw. The details are presented in Chapter 4. Our manuscript was presented at RECOMB 2019: Van Hoan Do, Mislav Blažević, Pablo Monteagudo, Luka Borozan, Khaled Elbassioni, Soeren Laue, Francisca Rojas Ringeling, Domagoj Matijevic and Stefan Canzar. *Dynamic pseudo-time warping of complex single-cell trajectories*. RECOMB 2019. Lecture Notes in Computer Science, 2019.

Visualization of multimodal omics data using Jvis

Emerging single-cell technologies assay multiple modalities such as transcriptome, genome, epigenome, and proteome at the same time (Cao et al., 2018; Stoeckius et al., 2017; Zhu et al., 2020). The joint analysis of multiple modalities has allowed to resolve subpopulations of cells at higher resolution (Do et al., 2021; Kim et al., 2020), has helped to infer the "acceleration" of RNA dynamics (Gorin et al., 2020) and to extend time periods over which cell states can be predicted (Qiu et al., 2019), and has linked dynamic changes in chromatin accessibility to transcription during cell-fate determination (Argelaguet et al., 2019). A fundamental step in the analysis of high dimensional single-cell data is their visualization in two dimensions. Arguably the most widely used nonlinear dimensionality reduction techniques are t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten and Hinton, 2008) and uniform manifold approximation and projection (UMAP) (McInnes et al., 2018). Currently, these techniques are applied to each modality one at a time (Cao et al., 2018; Argelaguet et al., 2019; Chen et al., 2019), and separate views of the data need to be reconciled by manual inspection. Here, we generalize t-SNE and UMAP to the joint visualization of multimodal single-cell measurements. While t-SNE and UMAP seek a low-dimensional embedding of cells that preserves similarities in the original (e.g. gene expression) space as well as possible, we propose j-SNE and j-UMAP that simultaneously preserve similarities across all modalities. Through Python package JVis they will combine different views of the data into a unified embedding that can help to uncover previously hidden relationships among them. At the same time, our joint embedding schemes learn the relative importance of each modality from the data to reveal a concise representation of cellular identity. The details are presented in Chapter 5, which is based on the publication: Van Hoan Do and Stefan Canzar. *A generalization of t-SNE and UMAP to single-cell multimodal omics*. Genome Biology, 22(1):130, 2021.

In summary, my contributions to the field of single-cell genomics data analysis are summarized in Figure 1.3.

1.3 List of peer-reviewed articles

- **Van Hoan Do** and Stefan Canzar. *A generalization of t-SNE and UMAP to single-cell multimodal omics*. Genome Biology, 22(1):130, 2021.
- **Van Hoan Do**, Francisca Rojas Ringeling, and Stefan Canzar. *Linear-time cluster ensembles of large-scale single-cell RNA-seq and multimodal data*. Genome Research, 31(4):677-688, 2021.

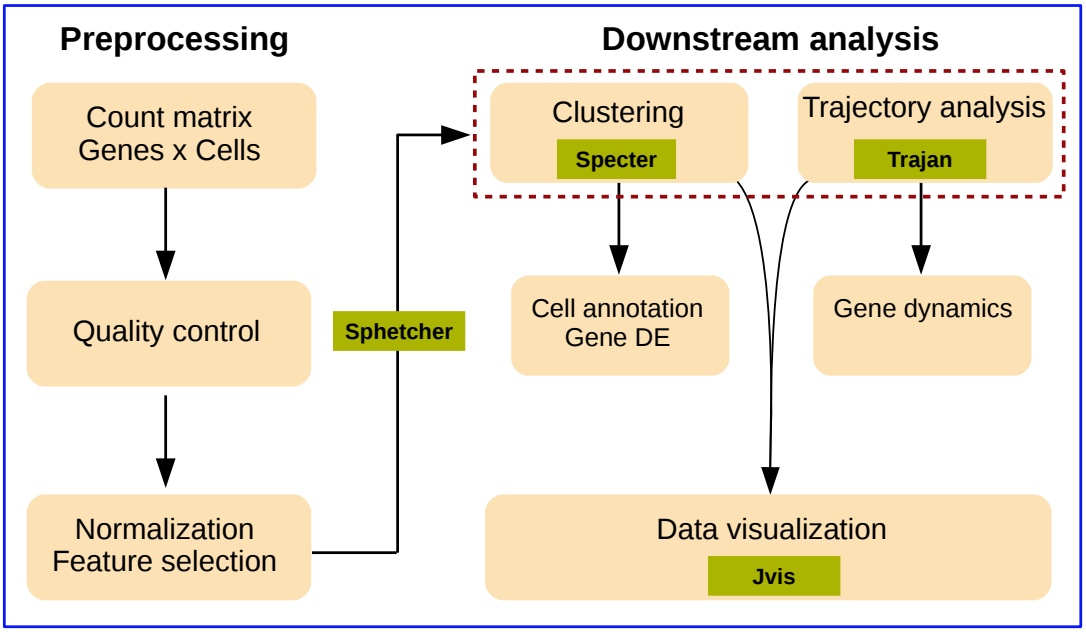

Figure 1.3: Contributed computational methods in single-cell genomics field. Contributed tools in this thesis are highlighted in green boxes. Sphetcher: geometric subsampling of big data; Specter: clustering of large-scale single cell genomics data; Trajan: alignment of single-cell trajectories; JVis: dimensionality reduction and visualization for multimodal omics data.

- **Van Hoan Do**, Khaled Elbassioni, and Stefan Canzar. *Sphetcher: Spherical thresholding improves sketching of single-cell transcriptomic heterogeneity.* iScience, 23(6):101126, 2020.
- **Van Hoan Do***, Mislav Blažević*, Pablo Monteagudo, Luka Borozan, Khaled Elbassioni, Soeren Laue, Francisca Rojas Ringeling, Domagoj Matijevic and Stefan Canzar. *Dynamic pseudo-time warping of complex single-cell trajectories.* RECOMB 2019. Lecture Notes in Computer Science, 2019.

*indicates equal contribution.

Chapter 2

Clustering of large-scale single-cell genomics data

This chapter is adapted from the publication: Van Hoan Do, Francisca Rojas Ringeling, and Stefan Canzar. *Linear-time cluster ensembles of large-scale single-cell RNA-seq and multimodal data*. Genome Research, 31(4):677-688, 2021.

One of the most fundamental computational tasks in the context of scRNA-seq analysis is the identification of groups of cells that are similar in their expression patterns, i.e. their transcriptomes, and which are at the same time distinct from other cells. Numerous methods have been proposed for clustering scRNA-seq data sets (Duò et al., 2018; Tian et al., 2019), with Seurat (Satija et al., 2015) and its underlying Louvain clustering algorithm (Blondel et al., 2008) being arguably the most widely used one. More recently, attempts have been made to design algorithms for the analysis of ultra-large scRNA-seq data sets, owing to the ever-increasing throughput of droplet-based sequencing technologies that allow to profile genome-wide expression for hundreds of thousands of cells at once. At the heart of such methods often lies a sampling technique that reduces the size of the data analyzed by a clustering algorithm. Cluster labels of cells in this so-called sketch are subsequently transferred to the remaining cells using, e.g., a nearest neighbor algorithm. dropClust (Sinha et al., 2018), for example, includes a structure preserving sampling step, but initially picks a small set of cells simply at random. Similarly, Seurat applies random subsampling prior to its nearest neighbor search.

The quality of the final clustering, however, strongly depends on how well the data sketch represents the overall cluster structure and how accurate the cluster labels of cells in the sketch can be inferred from incomplete data. Inaccurate labels of subsampled cells will likely lead to an inaccurate labeling of the full data. In addition, sampling cells proportional to their abundance might render rare cell types invisible to the algorithm. Geometric sketching was therefore recently proposed as an alternative sampling method that selects cells according to the transcriptomic space they occupy rather than their abundance. Nevertheless, labels need to be inferred from partial data.

Spectral methods for clustering have been applied with great success in many areas such as computer vision, robotics, and bioinformatics. They make few assumptions on cluster shapes and are able to detect clusters that form non-convex regions. On a variety of data types, this flexibility has allowed spectral clustering methods to produce more accurate clusterings than competing methods (Shi and Malik, 2000). The high computational complexity,

however, renders its application to large-scale problems infeasible. For n data points, spectral clustering computes eigenvectors of a $n \times n$ affinity matrix, which incurs a computational cost of $\mathcal{O}(n^3)$. For scRNA-seq data sets with n in the order of ten thousands up to the millions this presents a prohibitive cost which has thus prevented the application of spectral clustering to large-scale single-cell data sets.

Furthermore, spectral clustering methods are sensitive to the right choice of parameters used to model the similarity between data points (von Luxburg, 2007), i.e. RNA expression measurements of single cells. Data sets derived from different biological samples exhibiting different cell population structures obtained using different sequencing technologies typically require a different set of parameter values to achieve accurate clustering results. We introduce a new method, Specter, which addresses the challenges of computational complexity and parameter sensitivity to allow a tailored version of spectral clustering to be utilized in the analysis of large scRNA-seq data sets.

2.1 Preliminaries

We begin this chapter with an overview on clustering methods and evaluation metrics. Cluster analysis aims to group data points (cells) based on the similarity/distance among data points. The goal is that the points in the same group are more similar to each other than the points in the other groups. A group of similar points is called a cluster and a collection of all clusters is called a clustering. In addition, the goal is sometimes to put the clusters into a hierarchy by successively grouping the clusters so that at each level of the hierarchy, clusters in the same branch are more similar to each other than those in different branches.

There are numerous methods for performing cluster analysis, each method differs significantly in assumptions of what constitutes a cluster and how to find them. Notably, k-means, hierarchical clustering, and spectral clustering are the most popular methods. In this section we review the three clustering methods and then we introduce several metrics for evaluation of a clustering.

2.1.1 Clustering methods

k-means

We begin with a discussion of k-means, which is one of the most well-studied clustering methods due to its simplicity and scalability to large data sets. The idea of k-means is to partition the data into k groups and represent each group (cluster) by a cluster center (centroid). We measure the *distortion* of a clustering by the sum of the squares of the distances of each data point to its cluster center. The goal is to find a partition of points such that the distortion is minimum. Formally, given n data points $x_1, x_2, \ldots, x_n \in \mathbb{R}^m$, we define a cost function (distortion measure) of k clusters $C_1, C_2, \ldots, C_k$ as follows

$$\sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2, \tag{2.1}$$

where μ_i is the center of cluster C_i, that is, $\mu_i = \sum_{x \in C_i} x/|C_i|$. The k-means algorithm aims to find a partition $C_1, C_2, \ldots, C_k$ of the data such that the distortion measure (2.1) is minimum. Unfortunately, the problem of minimizing the k-means distortion is NP-hard. In

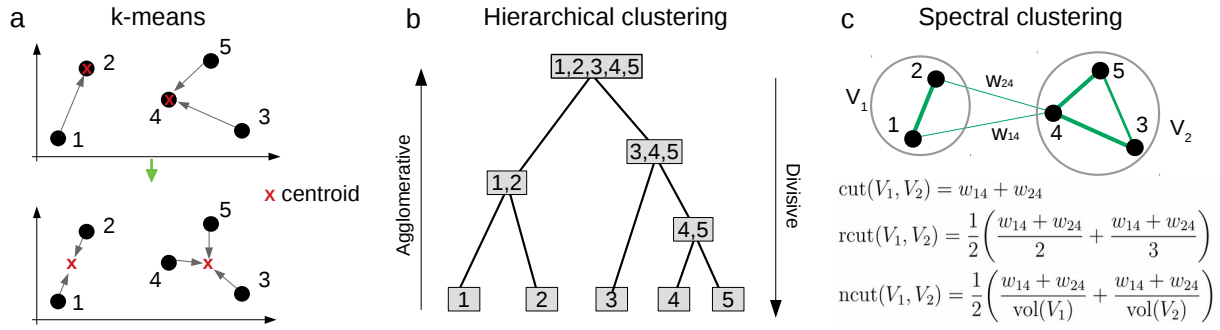

Figure 2.1: Illustrations of the three clustering methods. (a) Illustration of k-means on five data points with $k = 2$. The algorithm first selects two points (here 2 and 4) as initial centroids (marked by x), next every point is assigned to its closest centroid, resulting in two clusters $C_1 = \{1, 2\}, C_2 = \{3, 4, 5\}$ (top). Next the algorithm recomputes the cluster centers and assigns each point to its closest centroid (bottom). The algorithm terminates since the clustering is unchanged. (b) A dendrogram produced by hierarchical clustering of five data points in (a). (c) Spectral clustering first constructs a graph of five vertices, the weights of the graph representing the similarity between two vertices are indicated by the width of the corresponding edge. The thicker an edge, the more similarity two points are. The exact spectral clustering seeks for a partition of vertices (e.g., $V = V_1 \cup V_2, V_1 \cap V_2 = \emptyset$) which minimizes a graph cut described by one of the three objective functions (2.2)−(2.4). An example of the three cuts between V_1 and V_2 of the graph (top) is given in the bottom, where $|V_1| = 2, |V_2| = 3, \text{vol}(V_1) = w_{12} + w_{14} + w_{21} + w_{24} = 2w_{12} + w_{14} + w_{24}, \text{vol}(V_2) = 2w_{34} + 2w_{35} + 2w_{45} + w_{14} + w_{24}$. Here w_{ij} denotes the similarity between vertex i and vertex j.

practice, k-means is solved by an iterative procedure: (1) pick an initial set of k points (e.g., at random) as cluster centers, (2) assign each point x_i to its closest center, (3) change the cluster center to the average of its assigned points. The algorithm iterates between (2) and (3) until convergence (see Figure 2.1-a for an illustration). This algorithm is also referred to as Lloyd's algorithm. It can be shown that the algorithm will converge but it is not guaranteed to converge to the global minimum (Hartigan and Wong, 1979). Moreover, the quality of the k-means clustering strongly depends on the initialization of cluster centers in the step (1). Various schemes have been proposed for initializing the cluster centers in k-means (Celebi et al., 2013). Another challenging problem of k-means is how to determine the number of clusters k. The choice for the number of clusters depends on the goal of analysis. For single cell analysis k is usually defined as the number of cell types or subtypes in a sample, which is often unknown and in practice one has to try several k and picks one which fits well with prior biological knowledge, e.g., when clustering a scRNA-seq data set of blood tissues, we know roughly what cell types are and how many in the sample.

Solving k-means clustering problem is NP-hard, thus heuristic algorithms are generally used. Among them Floyd's algorithm is still the most widely used and its computational complexity is $\mathcal{O}(nkmi)$, where i is the number of iterations (Hartigan and Wong, 1979).

Hierarchical clustering

Hierarchical clustering is another popular clustering method. It does not necessarily require to provide the number of clusters as k-means. Hierarchical clustering can be divided into two paradigms: *agglomerative* (bottom-up) and *divisive* (top-down). Agglomerative strategy

starts with many clusters and iteratively merges a selected pair of clusters into a single cluster. Divisive paradigm on the other hand starts with a single cluster and breaks it down into smaller clusters. The hierarchy of the clusters is represented as a dendrogram (see Figure 2.1-b for an example). Here we focus on agglomerative clustering because it is more common than its counterpart. The agglomerative clustering starts with each data point as a separate cluster, it then successively merges the two most similar clusters. This process stops when all clusters merge together or it reaches the number of predefined clusters k. The pair of groups chosen for merging is the one with the smallest distance. There are several ways to measure distance between two clusters and they are called linkage methods. Some of the common linkage methods are given below.

Single linkage. The distance between two clusters is the shortest distance between two points in each cluster, i.e., $d(A, B) = \min_{x \in A, y \in B} d(x, y)$, where A, B are two clusters.

Complete linkage. The distance is defined by the largest distance between two points in each cluster, $d(A, B) = \max_{x \in A, y \in B} d(x, y)$.

Average linkage. The distance is defined by the average distance between two points in each clusters, $d(A, B) = \sum_{x \in A, y \in B} d(x, y)/(|A| \cdot |B|)$.

Centroid linkage. The distance between clusters A and B is $\|\mu_A - \mu_B\|^2$, where μ_A, μ_B are centroids of clusters A and B, respectively.

The overall complexity of the agglomerative clustering is $\mathcal{O}(n^3)$. However, optimal efficient agglomerative methods for single linkage and complete linkage is $\mathcal{O}(n^2)$ (Sibson, 1973; Defays, 1977).

Spectral clustering

Traditional clustering methods such as k-means often produce clusters, each of them having spherical or elliptical shape. Hence they will not work well when the shape of the clusters are non-convex, which is often the case in genomics data. Spectral clustering is designed for these situations. Spectral clustering uses eigenvectors of a matrix derived from the distance between points as a low-dimensional representation of the original data, which it then partitions using a method such as k-means. More precisely, given n data points $x_1, x_2, \ldots, x_n \in \mathbb{R}^m$ and a similarity matrix (affinity matrix) $W = (w_{ij})_{n \times n}$, where w_{ij} measures the similarity between points x_i and x_j, the graph Laplacian is defined as $L = D - W$ (unnormalized Laplacian) or $L = I - D^{-1/2} W D^{-1/2}$ in case of a (symmetric) normalized Laplacian. Here, D is a diagonal matrix whose entries are column sums (equivalently row sums) of W. Spectral clustering then uses the top k eigenvectors of L to partition the data into k clusters using the k-means algorithm. Why spectral clustering works is not clear at first, von Luxburg (2007) provided a justification of spectral clustering via graph partitioning. We briefly review it below and refer readers to the paper for details.

We encode the data in the form of a graph $G = (V, E)$, where V is the set of vertices (cells) and E is the set of edges. A vertex v_i in V represents the data point x_i and two vertices are connected by an edge if the similarity w_{ij} between the data points x_i and x_j is positive or larger than a certain threshold. The edge between v_i and v_j is weighted by the similarity w_{ij}. A common choice of the similarity between x_i and x_j is the Gaussian kernel $w_{ij} = \exp(-\|x_i - x_j\|^2/\sigma^2)$ for a user-defined parameter σ. We want to partition V into k groups $V_1, V_2, \ldots, V_k$. A good clustering should favor strongly connected nodes to end up in the same group, and nodes that are far apart (or disconnected) should be placed in different

groups (see Figure 2.1-c). This intuition can be modeled as a graph cut problem in which we minimize the *cut* defined as follows

$$cut(V_1, V_2, \cdots, V_k) = \frac{1}{2}\sum_{i=1}^{k} w(V_i, V \backslash V_i), \tag{2.2}$$

where $w(V_i, V\backslash V_i) = \sum_{s\in V_i, t\in V\backslash V_i} w_{st}$ is the total weight one needs to cut in order to disconnect V_i from the remaining vertices of the graph. In other words we favor the clustering with few edges between clusters. The *cut* objective function (2.2) often produces imbalanced clustering in which one cluster contains most of the data points and some clusters contain just a single vertex. This problem can be alleviated by using alternative objective functions: the *ratio cut* (Hagen and Kahng, 1992) and *normalized cut* (Shi and Malik, 2000) cost functions, which are respectively defined as:

$$rcut(V_1, V_2, \cdots, V_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{w(V_i, V\backslash V_i)}{|V_i|}, \tag{2.3}$$

and

$$ncut(V_1, V_2, \cdots, V_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{w(V_i, V\backslash V_i)}{\mathrm{vol}(V_i)}, \tag{2.4}$$

where $|V_i|$ is the number of vertices in V_i and $\mathrm{vol}(V_i) = \sum_{s\in V_i}\sum_{t\in V} w(s,t)$ is the volume of V_i. An example of the three cuts are given in Figure 2.1-c. The problems of finding a clustering (partition) with the minimum ratio cut and normalized cut are NP-hard. The relaxed version of the two problems lead to spectral clustering of unnormalized and normalized Laplacian we introduced above (von Luxburg, 2007).

Due to computational burden of spectral clustering in the eigendecomposition step ($\mathcal{O}(n^3)$), several methods have been proposed to accelerate the spectral clustering algorithm (Fowlkes et al., 2004; Shinnou and Sasaki, 2008; Cai and Chen, 2011). In particular, Landmark-based Spectral Clustering (LSC) has been shown to perform well in terms of efficiency and effectiveness compared to state-of-the-art methods across a large number of data sets (Cai and Chen, 2011). In short, LSC picks a small set of p representative data points $u_1, u_2, \ldots, u_p \in \mathbb{R}^m$, i.e. the landmarks, which it then uses to create a representation matrix $Z \in \mathbb{R}^{p\times n}$ whose columns represent the original data with respect to the landmarks according to $X \approx UZ$. Here, columns i of $U \in \mathbb{R}^{m\times p}$ contain landmarks u_i and columns i of X contain the original input points x_i. Let the Gaussian kernel $K(x,y) = \exp(-\|x-y\|^2/2\sigma^2)$ measure the similarity between two points x and y, then matrix $Z = (z_{ji})_{p\times n}$ is computed using Nadaraya-Watson kernel regression (Härdle, 1990) as

$$z_{ji} = \begin{cases} \frac{K(x_i,u_j)}{\sum_{j'\in U_{\langle i\rangle}} K(x_i,u_{j'})} & \text{if } j \in U_{\langle i\rangle} \\ 0 & \text{otherwise,} \end{cases} \tag{2.5}$$

where $U_{\langle i\rangle}$ is the set of r nearest landmarks of x_i. That is, z_{ji} is set to zero if u_j is not among the r nearest neighbors of x_i, which naturally leads to a sparse representation of the data. Motivated by non-negative matrix factorization that uses k (i.e. number of clusters) basis vectors to represent each data point (Xu et al., 2003). Then each original point x_i can be

approximated by

$$\hat{x_i} = \sum_{j=1}^{p} z_{ji} u_j.$$

From this landmark-based representation of the *complete* data it computes the Laplacian matrix $L = \hat{Z}^T \hat{Z}$, where $\hat{Z} = D^{-1/2} Z$ and D is the diagonal matrix whose (i,i)-entry equals the sum of the ith row of Z. Then, this graph Laplacian L admits a fast eigendecomposition in time $\mathcal{O}(n)$ as oppose to $\mathcal{O}(n^3)$ in the general case, which is described in more detail in Cai and Chen (2011). The LSC algorithm is summarized in Algorithm 1.

Algorithm 1: LSC

1 **Input:** Cells $x_1, \ldots, x_n$; number of clusters k
2 Compute p landmarks using random selection or k-means.
3 Construct a sparse similarity matrix $Z \in \mathbb{R}^{p \times n}$ between data points and landmarks as in (2.5).
4 Compute the first k eigenvectors $b_1, b_2, \ldots, b_k$ of the Laplacian $L = \hat{Z}^T \hat{Z}$ where $\hat{Z} = D^{-1/2} Z$ and let $B = [b_1, b_2, \ldots, b_k]$.
5 Apply k-means on B to obtain k clusters.

2.1.2 Clustering evaluation

In this section we introduce several popular metrics for evaluation of a clustering. We begin with the most widely used metric in scRNA-seq clustering analysis.

Given a set of n data points $X = \{x_1, x_2, \ldots, x_n\}$, and let $A = \{A_1, A_2, \ldots, A_r\}$ and $B = \{B_1, B_2, \ldots, B_s\}$ be two clusterings of X. Here A often represents the ground truth labels and B is a clustering produced by a particular clustering method. Sometimes A and B correspond to outcomes of different clustering methods/runs. In this case we want to measure the similarity between the two clusterings. Note that the number of clusters in A and B can be different.

The overlap between A and B is summarized in a contingency table n_{ij} where each entry n_{ij} denotes the number of common objects between A_i and B_j, i.e., $n_{ij} = |A_i \cap B_j|$. Then the Adjusted rand index (ARI) (Hubert and Arabie, 1985) between A and B is defined as

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}$$

where $a_i = \sum_\ell n_{i\ell}$, and $b_j = \sum_\ell n_{\ell j}$. ARI is the corrected-for-chance version of the Rand index, which is computed based on the number of pairs of data points on which two clusterings agree or disagree. ARI ranges between -1 and 1 and it yields a value 0 if the clusterings A and B are completely independent and 1 if they are identical.

Normalized mutual information (NMI) (Studholme et al., 1999) is another measure of the similarity between two clusterings, which quantifies the statistical information shared between two distributions. The NMI between two clusterings A and B is defined as

$$NMI = \frac{\sum_{i=1}^{r} \sum_{j=1}^{s} n_{ij} \log \frac{n_{ij} n}{|A_i||B_j|}}{\sqrt{\left[\sum_{i=1}^{r} |A_i| \log \frac{|A_i|}{n}\right] \left[\sum_{j=1}^{s} |B_j| \log \frac{|B_j|}{n}\right]}}.$$

The range of NMI is from 0 to 1 and the larger the NMI the more agreement between the two clusterings. Similar to ARI, NMI = 0 indicates that the two clustering are independent and NMI = 1 if they are identical.

The ARI and NMI require a ground truth or they are used to compare two clusterings, the Silhouette score on the other hand relies only on the clustering itself and the data. The Silhouette score (Rousseeuw, 1987) ranges between -1 and 1 and it measures how much overlapping clusters (score 0) and how well separated (score 1) they are. Given a clustering $C = \{C_1, C_2, \ldots, C_k\}$, for each data point $x \in C_i$, let

$$a(x) = \frac{1}{|C_i| - 1} \sum_{y \in C_i, y \neq x} d(x, y)$$

be the average distance between x and all data points in the same cluster as x, where $d(x, y)$ is the distance between x and y. Here $a(x)$ represents how well x is assigned to its own cluster. Small $a(x)$ indicates that x lies in middle of its cluster. Next we define the dissimilarity of x to other clusters as follows

$$b(x) = \min_{j \neq i} \frac{1}{|C_j|} \sum_{y \in C_j} d(x, y).$$

Here $b(x)$ is the smallest average distance of x to all points in any other clusters which do not contain x. Then the Silhouette of x is given by

$$s(x) = \frac{b(x) - a(x)}{\max\{(a(x), b(x)\}}, \text{if } |C_i| > 1,$$

and $s(x) = 0$, if $|C_i| = 1$. The Silhouette score is the average Silhouette over all data points, that is, $\sum_{x \in X} s(x)/n$.

2.2 Methods

In this section we describe our clustering algorithm Specter. We begin with an overview of the method.

2.2.1 Overview of Specter

We adopt the idea of landmark-based spectral clustering described in the previous section, where a sample of cells are selected to create a sparse representation of the *full* data from which a spectral embedding can then be computed in $\mathcal{O}(n)$. Since the LSC algorithm is sensitive to the choice of parameters (the number of landmarks p and the Gaussian bandwidth σ), we run different choices of parameters and reconcile the resulting clustering components into a single (consensus) clustering, this is also known in literature as cluster ensembles (Strehl and Ghosh, 2003). In addition to combining clusterings from different runs of the algorithm on the same data, consensus clustering can also be used to reconcile clusterings of cells based on different modalities, for example, gene expression and surface protein levels produced by CITE-seq (Stoeckius et al., 2017). Specter's consensus clustering paradigm can resolve subpopulations of cells that cannot accurately be distinguished based on transcriptomic differences alone. We combine consensus clustering with a novel *selective sampling*

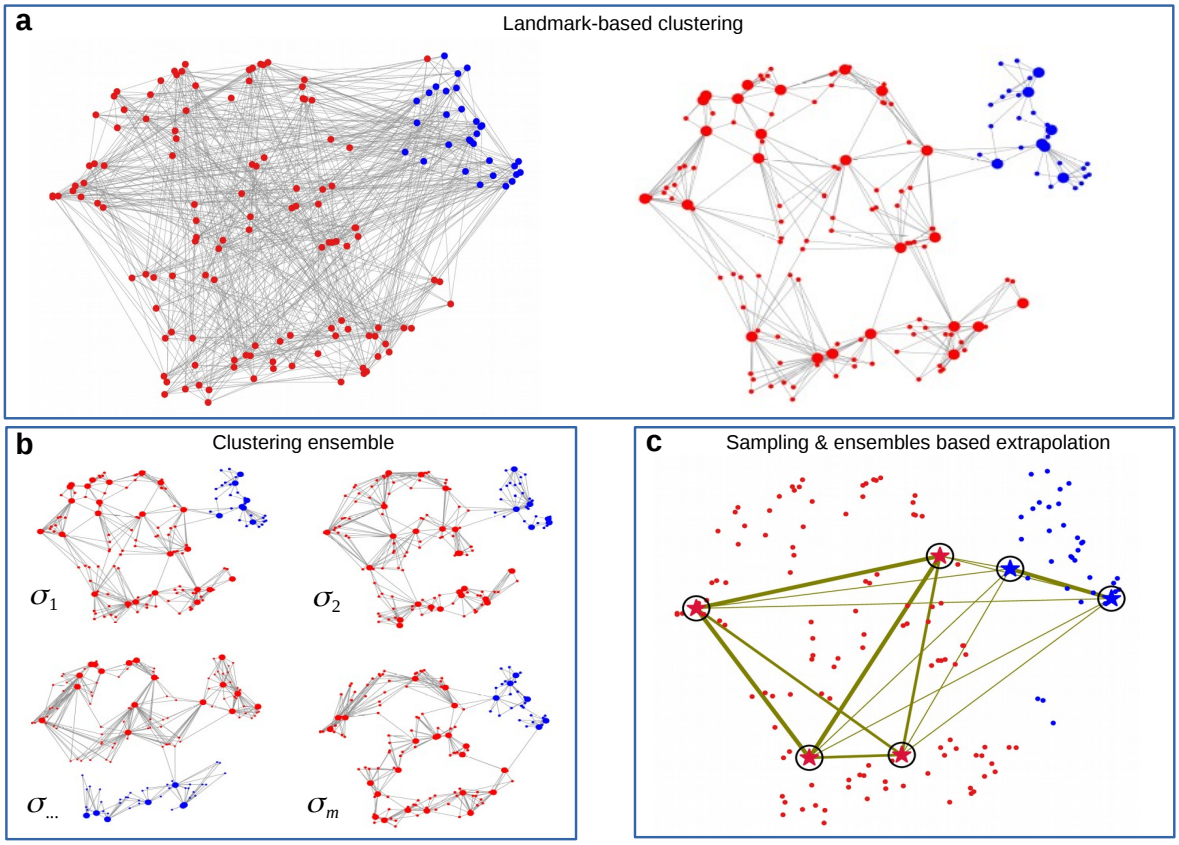

Figure 2.2: Overview of Specter. Illustrations are based on t-SNE visualizations of a random subsample of scRNA-seq data by Grün et al. (2016). (**a**) Standard spectral clustering constructs an affinity matrix that captures (transcriptional) similarities between all pairs of cells (left) which renders its eigen decomposition prohibitively expensive for large data sets. In contrast (right), describing each cell (small circles) with respect to its nearby *landmarks* (big circles) that were initially selected as the means computed by k-means clustering, creates a sparse representation of the full data that dramatically speeds up the computation of a spectral embedding. Cell are colored to distinguish sorted hematopoetic stem cells (blue) from other mouse bone marrow cells (red) assayed by Grün et al. (2016). (**b**) Specter does not rely on a single set of parameters, but performs multiple runs of landmark-based clustering using different sets of landmarks of different size and different measures of similarities between cells (parameterized by σ). Three clusterings closely resemble the true labeling shown in (a), while one differs substantially. (**c**) Specter reconciles all individual clusterings into a consensus clustering. It clusters a carefully selected subset of cells (marked by circled stars) based on their co-association across all individual clusterings in (b), indicated by the width of the corresponding edge. The thicker an edge, the more often its two endpoints were placed in the same cluster. Here, the 4 red stars and the 2 blue stars correctly form 2 groups of cell, whose labels are finally propagated to the remaining cells using 1-nearest neighbor classification. The final clustering shown in (c) closely resembles the true clustering in (a).

strategy that makes use of clustering information obtained from the *full* data set to achieve overall linear time complexity. Finally, we transfer cluster labels to the remaining cells using k-nearest neighbors classification. We provide an overview of the approach in Figure 2.2.

2.2.2 Landmark-based spectral clustering of single cells

In the following description of our algorithm we assume a given number of clusters k. In Specter we determine the number of clusters based on the Silhouette index (Rousseeuw, 1987), which performed particularly well in recent benchmark studies (Arbelaitz et al., 2013; Chouikhi et al., 2015).

We tailor the idea of landmark-based spectral clustering described in the previous section to the characteristics and scale of modern scRNA-seq data sets. In particular, the choice of bandwidth σ used in the (Gaussian) kernel to smooth the measure of similarity between pairs of data points heavily depends on the type of data and can have a strong impact on the final clustering. In the original approach, parameter σ is set to the average Euclidean distance between data points and their K-nearest landmarks, i.e. to the average value of all elements in matrix Z. We empirically find that replacing the average by the maximum value, i.e. by setting $\sigma = \gamma \times \text{mean}(\max(Z))$, where $\max(Z)$ denotes a vector of maximum values for each row in Z and γ a randomly chosen parameter between 0 and 1, is able to better capture the transcriptional similarity between single cells and yields more accurate clusterings of cells. We set the parameter r in LSC algorithm to be equal to k in Specter (and in all experiments).

Furthermore, we pair the theoretical reduction in time complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$ with a practical speed-up of the LSC algorithm by applying a hybrid strategy when selecting the landmarks. The choice of representative data points, here single cells, plays a crucial role in the quality of the final clustering. Random selection or k-means clustering were originally proposed as procedures for picking landmarks (Cai and Chen, 2011). Random selection of representative cells is very efficient but often yields random sets of cells that do not represent the full data well and thus lead to poor clustering results. k-means, on the other hand, better takes into account the structure of the data when selecting landmark cells but its higher computational cost makes it impractical for large scRNA-seq data sets where it accounts for around 90% of the overall running time in our experiments. Our hybrid strategy seeks to balance the efficiency of random sampling and the accuracy of k-means based landmark selection. It first picks a set of p' candidate landmarks uniformly at random with $p' \ll n$ (by default, $p' = 10p$), from which it subsequently selects $p < p'$ final landmark cells using the k-means algorithm. Note that despite the initial random sampling, the *full* data are represented by the final set of landmarks.

Finally, for data sets that contain a small number of clusters, we adjust the spectral embedding based on which the original data is clustered using k-means in the last step of spectral clustering. For a small number of clusters (e.g. $k \leq 4$), the top k eigenvectors used in the original approach typically do not contain enough information to represent the full data well. In this case, we therefore use the top $k + 2$ eigenvectors to compute the spectral embedding.

2.2.3 Clustering ensembles across parameters and modalities

Different data types require a different choice of parameter values and there is no general rule how to select the best one. To address this issue, we employ consensus clustering, also known in literature as cluster ensembles (Strehl and Ghosh, 2003), in the same way as ensemble learning is used in supervised learning. In particular, we generate a series of component clusterings by varying the number of selected landmarks p and the kernel bandwidth. We

randomly select parameter γ which controls the bandwidth of the Gaussian kernel from interval [0.1, 0.2] and choose p from interval

$$[\min(8k\log(k), \lceil n/3 \rceil), \min(10k\log(k), \lceil n/2 \rceil)]\,.$$

This choice of p is motivated by a result by Tremblay et al. (2016) who used sampling theory of bandlimited graph-signal developed in Puy et al. (2016) to prove that clustering a random subset of size $O(k\log(k))$ is sufficient to accurately infer the cluster labels of all elements. To avoid sampling too many landmarks for small data sets (i.e. small number of cells n), we additionally set upper bounds $\lceil n/3 \rceil$ and $\lceil n/2 \rceil$ for the left and right boundaries of the interval, respectively. All clusterings produced by the different runs of our tailored LSC algorithm are then summarized in a co-association matrix H (Fred and Jain, 2005) in which entry (i, j) counts the number of runs that placed cells i and j in the same cluster. We compute the final clustering through a hierarchical clustering of matrix H. Our LSC-based consensus clustering approach is summarized in Algorithm 2.

Different parameter choices (e.g. kernel bandwidths) provide different interpretations of the same data. In the same way as clustering ensembles can help unifying these different views on a single modality, they can help reconcile the measurements of multiple modalities, such as transcriptome and proteome, of the same cell. More specifically, Specter produces an identical number of clusterings for each modality in step 2 of Algorithm 2 which it then combines through the same co-association approach (steps 3 and 4).

Algorithm 2: LSC ensemble

1 **Input:** Cells $x_1, \ldots, x_n$; number of clusters k
2 Run the tailored LSC algorithm for different kernel bandwidths and varying numbers of landmarks.
3 Summarize all clusterings in a co-association matrix H.
4 Apply the single linkage hierarchical clustering algorithm to H to obtain the final k clusters.

Time complexity The time complexity of the tailored LSC algorithm is $O(n)$, and single linkage hierarchical clustering requires $O(n^2)$ time, yielding an overall complexity of $O(n^2)$ for Algorithm 2, assuming k is small enough to be considered a constant.

2.2.4 Selective sampling-based clustering ensemble

With a running time that scales quadratically with the number of cells, the application of Algorithm 2 to large-scale scRNA-seq data sets becomes infeasible. We therefore apply step 3 of our clustering ensemble approach (Algorithm 2) to a carefully selected sketch of the data. Note, however, that the co-association matrix H built in step 3 of the algorithm is based on cluster labels that were learned from the *full* data in step 2 using our tailored LSC algorithm. In addition, we propose a simple sampling technique that uses all clusterings computed in step 2 to guide the selection of cells.

Selective Sampling Sampling cells uniformly at random is naturally fast, since the decision to include a given cell into a sketch does not depend on any other cell. At the same time,

these independent decisions ignore the global structure of the data such as the abundance of different cell types and may thus lead to a loss of rare cell types (Hie et al., 2019b). We therefore propose a sampling approach that utilizes the clusterings of the data computed in step 2 of Algorithm 2 to inform the (fast) selection of cells. More specifically, Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$, where $\pi_i = (\pi_{i1}, \pi_{i2}, \ldots, \pi_{ik})$ is the ith clustering returned in step 2 of Algorithm 2, $i = 1, 2, \ldots, m$. We select a sketch S of size $\min\{[10k\log(k)], [k\sqrt{n}]\}$ that contains roughly the same number of cells in each cluster π_{ij}, for all i and j. This selective sampling procedure iterates through all clusters contained in all clusterings from which it randomly picks a cell not already contained in the sketch, until the size of the sketch reaches $\min\{[10k\log(k)], [k\sqrt{n}]\}$ (see Algorithm 3).

Algorithm 3: Selective sampling

Input: Component clusterings $\Pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$, number of clusters k.
Initialization: $S = \emptyset$.
while $|S| < \min\{[10k\log(k)], [k\sqrt{n}]\}$ **do**
  for $i = 1$ *to* m **do**
    for $j = 1$ *to* k **do**
      Randomly select a cell s from $\pi_{ij} \backslash S$
      $S = S \cup \{s\}$
    end
  end
end

Inference Given a selectively sampled sketch S, we apply steps 3 and 4 in Algorithm 2 to cells in S, using labels obtained from the full data in step 2. That is, we construct a co-association matrix whose entries count the number of times the two corresponding cells in S were placed in the same cluster by a run of the LSC algorithm in step 2. From this matrix, we compute a consensus clustering of S using hierarchical clustering and finally transfer cluster labels to the remaining cells using supervised k-nearest neighbors classification. That is, we assign each cell not in S to the cluster that the majority of its k nearest neighbors were placed in by the preceding consensus clustering of S. Our selective sampling-based cluster ensemble approach is summarized in Algorithm 4.

Algorithm 4: Selective sampling-based clustering ensemble

Input: Cells $x_1, \ldots, x_n$; number of clusters k
Run the tailored LSC algorithm for a varying number of landmarks and different kernel bandwidths. Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$ be the set of m clusterings.
Run selective sampling (Algorithm 3) on Π to obtain a sketch S of size $|S| = \min\{10k\log(k), [k\sqrt{n}]\}$.
Summarize all clusterings of cells in S computed in step 2 in a co-association matrix H^S.
Apply single linkage hierarchical clustering to H^S to obtain k clusters for S.
Transfer labels to full data using k-nearest neighbors classification

Time complexity Landmark-based spectral clustering performed in step 2 of Algorithm 4 takes $O(n)$, see above. Since we selectively sample a sketch of size $|S| = O(\sqrt{n})$ in step 3, the complexity of steps 4 and 5 now reduces to $O(n)$. Together with the k-NN classification that runs in $O(n)$ in step 6, our selective sampling based cluster ensemble scheme scales linearly with the number of cells n.

2.3 Results

We implemented Algorithm 2− 4 in software tool Specter. We show results for Specter when using 20 ensemble members (*Specter20E*) and 50 ensemble members (*Specter50E*), which we motivate below through experiments addressing the dependence of Specter's accuracy on the number of ensemble members. The results for these two variants are nearly identical and we therefore simply refer to them as Specter unless we explicitly distinguish these two settings. Due to our clustering ensemble scheme, no additional tuning of parameters is required to apply Specter to the data sets.

2.3.1 Specter is more accurate than competing methods

We compared the performance of Specter to representative scRNA-seq clustering methods SC3 (v1.10.1) (Kiselev et al., 2017), Seurat (v2.3.4) (Satija et al., 2015), dropClust (v2.1.0) (Sinha et al., 2018), RCA (v2.0) (Li et al., 2017), TSCAN (v1.24.0) (Ji and Ji, 2016), RaceID3 (v0.2.1) (Herman et al., 2018), CIDR (v0.1.5) (Lin et al., 2017), RtsneKmeans (Duò et al., 2018) as well as to a geometric sketching based clustering approach (Hie et al., 2019b). SC3 and Seurat consistently demonstrated superior performance over competing methods in several clustering benchmarks (Duò et al., 2018; Tian et al., 2019) and are routinely used in scRNA-seq based cell type analyses. The graph-based Louvain clustering approach used by Seurat has an additional speed advantage over SC3, which applies a consensus clustering scheme to obtain particularly accurate clusterings. dropClust was recently proposed for the analysis of ultra-large scRNA-seq data sets and follows a strategy outlined above. It first reduces the size of the data to a maximum of 20,000 cells using random sampling. After a second sampling step based on Louvain clusters, it applies average-linkage hierarchical clustering on the sampled cells. Cluster labels are then transferred to the remaining cells using a Locality Sensitive Hashing forest (Bawa et al., 2005) for approximate nearest neighbor searches. In contrast, the geometric sketching algorithm proposed in Hie et al. (2019b) samples cells evenly across the transcriptional space rather than proportional to the abundance of cell types as uniform sampling schemes do. Experiments in Hie et al. (2019b) demonstrated that clustering a geometric sketch using the graph-based Louvain algorithm followed by propagating labels to the remaining cells via k-nearest-neighbor classification accelerates clustering analysis and yields more accurate results than uniform sampling strategies. We include the same geometric sketching based clustering method in our benchmark and refer to it simply as geometric sketching throughout the text. We further included methods RCA, TSCAN, RaceID3, and CIDR to cover a diverse set of algorithms commonly used to cluster scRNA-seq data (see recent benchmarks (Duò et al., 2018; Tian et al., 2019; Freytag et al., 2018)), from nearest neighbor based graph clustering to hierarchical clustering to k-medoids to model-based clustering. Finally, we included general-purpose K-means clustering (RtsneKmeans) as a baseline that performed surprisingly well in Duò et al. (2018) compared to

methods specifically developed for clustering scRNA-seq data.

Data sets and evaluation

We evaluated Specter and competing methods on 21 public scRNA-seq data sets and 24 simulated data sets (Table 2.1, Supplemental Table S1). The former includes 16 data sets for which cell type labels were inferred in the original publication from clusterings of scRNA-seq measurements which typically underwent manual refinement and annotation as well as all but one real data sets that were used in Duò et al. (2018) to benchmark clustering methods based on cell phenotypes defined independently of scRNA-seq. Identically to Duò et al. (2018), we used "true" cell types annotated by FACS sorting in the *Koh* data set, and partitioned cells by genetic perturbation and growth medium in the *Kumar* data set. In data sets *Zhengmix4eq* and *Zhengmix4uneq* the authors of Duò et al. (2018) randomly mixed equal and unequal proportions, respectively, of pre-sorted B-cells, CD14 monocytes, naive cytotoxic T cells and regulatory T cells. Data set *Zhengmix8eq* additionally contained roughly equal proportions of CD56 NK cells, memory T cells, CD4 T helper cells, and naive T cells. Again, annotated cell types were used as reference partitioning of cells in the evaluation. We excluded a single data set from Duò et al. (2018) in which ground truth labels correspond to collection time points which all methods tested in Duò et al. (2018) failed to reconstruct. Data sets vary in size and number of cell populations and are described in Table 2.1. We used Splatter (Zappia et al., 2017) to simulate 24 data sets that varied in the relative abundance of cell types that were either all equal (G*eq*), unequal (G*neq*), or based on cell type abundances among peripheral blood mononuclear cells (PBMCs) in healthy individuals (G*pbmc*), in number of cells (N*1k*, N*2k*, N*5k*), and in the probability of a gene being differentially expressed in a group, which was either 0.01 (DE*1*), 0.02 (DE*2*), 0.05 (DE*5*), or differed between groups (DE*neq*). Following Zappia et al. (2020), we set the number of genes to $1,000$ or $10,000$ (D*10k*). Supplemental Table S1 lists the characteristics of all simulated data sets.

We apply standard and uniform preprocessing (Duò et al., 2018) on all real and simulated data sets, including natural log-transformation of gene counts after adding a pseudo-count of 1, selection of top 2,000 most variable genes (omitted for simulated data sets with less than 2,000 genes), followed by dimensionality reduction to 100 principle components (Vijayan, 2020). The geometric sketching based Louvain clustering is provided with the same pre-processed data as Specter, all other methods are run with their built-in data preprocessing. Consistent with the original publication (Hie et al., 2019b), geometric sketches ranging from 2% to 10% of the original number of cells were computed and clustered as described above. All methods were provided the correct number of clusters or corresponding parameters were tuned accordingly. All experiments were run on a Intel Xeon CPU @2.30GHz with 320 GB memory. Methods SC3, RCA, RaceID3, and CIDR failed to run on the three largest data sets that included more than $450,000$ cells (Table 2.1) due to insufficient memory. In fact, with a running time that grows cubic with the number of cells, SC3 is not designed for large data sets. On data set *chen*, for example, it takes SC3 five hours to cluster 14,000 cells. Similarly, on the three largest data sets we replaced the R implementation of the Louvain clustering algorithm called in the Seurat clustering pipeline by a more efficient python implementation of the same algorithm in the SCANPY package (v1.4.6) (Wolf et al., 2018). SCANPY was specifically designed for the analysis of large-scale gene expression data sets and was used originally (Cao et al., 2019) to identify cell types in data set *trapnell* comprising more than

Table 2.1: Overview of the real data sets used in this study. Names listed in the left-most column are used throughout the text. A line separates data sets in which cell type labels were inferred from scRNA-seq measurements from data set were labels are based on cell phenotypes defined independently of scRNA-seq. k: number of populations.

Data set	# Cells	k	Description	Reference
grun	1502	2	mouse stem cells	Grün et al. (2016)
xin	1600	8	human islet cells	Xin et al. (2016)
baron	1886	13	human and mouse pancreas	Baron et al. (2016)
biase	56	4	mouse embryo devel	Biase et al. (2014)
deng-1	268	6	mouse embryo devel (RPKMs)	Deng et al. (2014)
deng-2	268	6	mouse embryo devel (Reads)	Deng et al. (2014)
goolam	114	5	mouse embryo	Goolam et al. (2016)
muraro	2126	10	human pancreas	Muraro et al. (2016)
patel	430	5	human glioblastoma	Patel et al. (2014)
pollen	301	11	human developing cortex	Pollen et al. (2014)
klein	2717	4	mouse embryo stem cells	Klein et al. (2015)
zeisel	3005	9	cortex and hippocampus	Zeisel et al. (2015)
chen	14,437	45	mouse brain	Chen et al. (2017)
CNS	465,281	7	mouse central nervous system	Zeisel et al. (2018)
saunders	665,858	11	adult mouse brain	Saunders et al. (2018)
trapnell	2,058,652	38	mouse organogenesis cell atlas	Cao et al. (2019)
Koh	531	9	human embryonic stem cells	Koh et al. (2016)
Kumar	246	3	mouse embryonic stem cells	Kumar et al. (2014)
Zhengmix4eq	3,994	4	mixture of purified PBMCs	Zheng et al. (2017)
Zhengmix4uneq	6,498	4	mixture of purified PBMCs	Zheng et al. (2017)
Zhengmix8eq	3,994	8	mixture of purified PBMCs	Zheng et al. (2017)

2 million cells.

Consistent with other benchmarks (see, e.g., (Duò et al., 2018; Sinha et al., 2018; Freytag et al., 2018)), we used the Adjusted Rand index (ARI) (Hubert and Arabie, 1985) to measure the similarity between the inferred clusterings and the ground truth clustering that is based on the biological cell types annotated or pre-sorted in the original study or was provided by the simulator. We additionally applied routinely used (Freytag et al., 2018) clustering metrics Normalized Mutual Information (NMI) (Studholme et al., 1999) and a homogeneity score (Rosenberg and Hirschberg, 2007) to provide a more detailed analysis of clustering performance.

Evaluation on real data

Consistent with previous benchmarks, SC3 and Seurat overall outperform existing methods, with RCA showing a competitive performance especially with respect to homogeneity scores (Figure 2.3 and Supplemental Figures S1, S2). Specter, however, improves mean clustering accuracy over both methods, in all three metrics. The biggest improvement can be observed with respect to ARI and homogeneity scores, whose mean values (excluding the three largest data sets where SC3 failed to run) achieved by Specter (*Specter50E*) are 0.88 and 0.89, respectively, compared to 0.69 and 0.76 for Seurat and 0.78 and 0.84 for SC3. Overall, most methods achieved higher scores in NMI than in the other two metrics. On 17 out of 21 real data sets, Specter obtained more accurate clusterings in all three metrics than

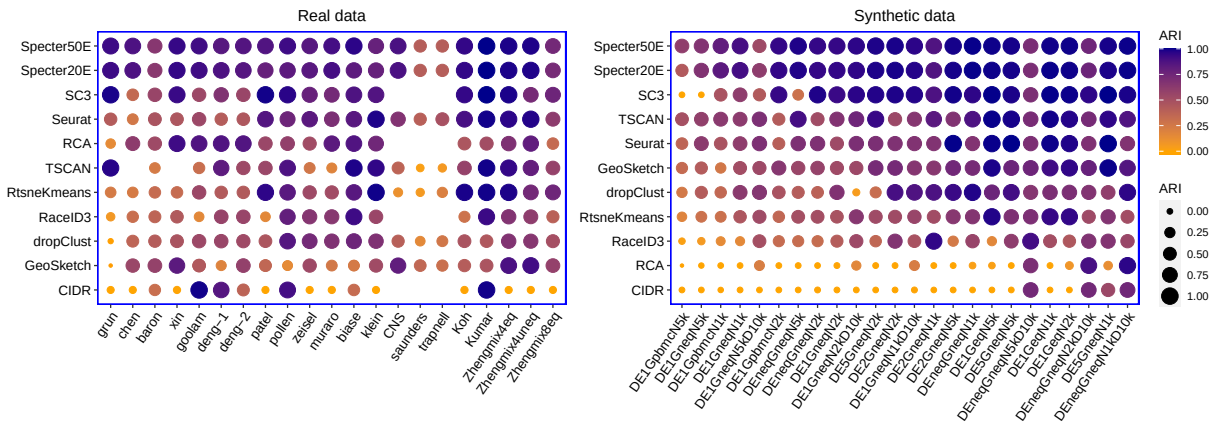

Figure 2.3: Clustering performance measured in ARI of Specter and competing methods on real and synthetic scRNA-seq data sets. Methods are ordered by mean ARI score across data sets decreasing from top to bottom. In the calculation of mean scores we excluded for each method the data sets where the method did not run successfully. For the rightmost 5 real data sets ground truth labels are based on cell phenotypes defined independently of scRNA-seq (see Table 2.1). Synthetic data sets are ordered from left to right by increasing mean ARI over all methods. SC3, RCA, RaceID3, and CIDR failed to run on the three largest data sets *CNS*, *saunders*, and *trapnell* due to insufficient memory. TSCAN failed to run on data sets *chen* and *skin* for unknown reasons. Geometric sketching refers to the Louvain clustering of 10% of the cells sampled using geometric sketching. Results for different sketch sizes are shown in Supplemental Figures S3.

Seurat and without exception achieved higher ARI scores than sampling based methods dropClust and geometric sketching, even when sampling as many as 10% of cells in the latter approach. Results for smaller sketch sizes are shown in Supplemental Figure S3. A similar preeminence can be observed when applying metrics NMI and homogeneity score. On many instances, the improvement was substantial. In fact, on average methods dropClust and geometric sketching achieved slightly lower scores with respect to all three metrics than baseline algorithm RtsneKmeans that simply applies standard k-means clustering on t-SNE projected cells. Note that the ground truth labeling of cell types in data sets *trapnell*, *CNS*, and *saunders* was obtained in the original publication using Seurat or its underlying Louvain clustering algorithm. Despite the additional manual refinement applied in (Zeisel et al., 2018; Cao et al., 2019; Saunders et al., 2018), this might positively impact the evaluation results of Seurat and the geometric sketching based Louvain clustering. On several instances, Specter achieved considerably higher ARI scores than SC3, while on others their performance was similar (within less than 10% difference in ARI). Note, however, that SC3 is not designed to cluster large data sets and had to be excluded from the comparison on the three largest data sets for computational reasons.

Evaluation on simulated data

As expected, simulated data sets G*pbmc* that reflect the unbalanced cell type composition among PBMCs pose the biggest challenge to clustering algorithms, while uniform cell type abundances (G*eq*) or a larger number of marker genes (DE*neq**D*10k* or DE*5*) facilitate the detection of transcriptionally distinct groups of cells (Figure 2.3, Supplemental Figures S1

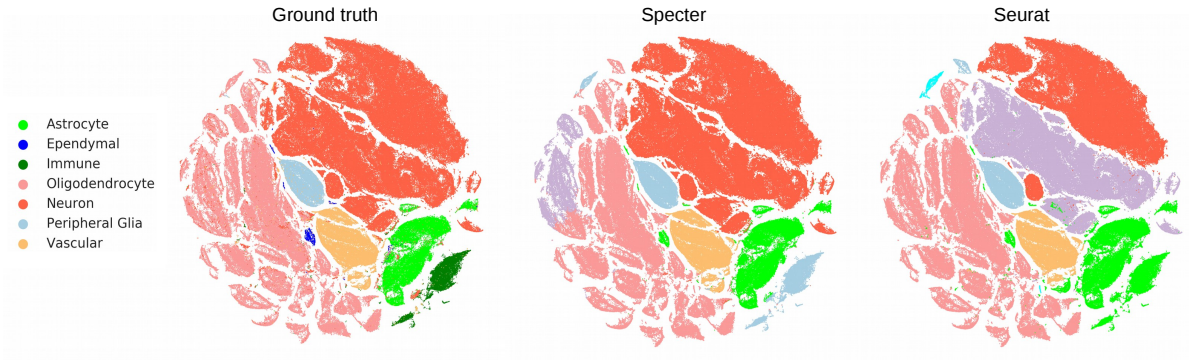

Figure 2.4: t-SNE visualization of single cells of the CNS data set. Cells in the ground truth representation (left) are colored by cell type specified by the legend. The visualization of Specter (middle) and Seurat (right) clusterings use the same 2D embedding as the ground truth, but cells are colored according to clusters inferred by the two methods; colors do not directly reflect cell types specified by the legend. As expected by the higher ARI (0.89 vs 0.67) (and higher homogeneity scores of 0.81 vs 0.71 and NMI of 0.84 vs 0.78), Specter makes fewer mistakes. In contrast to Specter, Seurat wrongly splits neurons into 2 populations, is not able to distinguish astrocytes from immune cells, and is similarly not able to distinguish a subpopulation of vascular cells from astrocytes.

and S2). Consistent with results on real data sets, Specter achieved highest accuracy in terms of mean ARI, NMI, and homogeneity score across 24 simulated data sets, with scores in NMI being generally higher for most methods than in the other two metrics. Again, SC3 performs best among remaining methods in terms of mean ARI and mean homogeneity score which may be attributed to a consensus clustering scheme that it applies similarly to Specter. With respect to NMI, Seurat and TSCAN achieved slightly higher mean scores than SC3, mainly due to the two presumably most difficult instances where SC3 returned clusterings with a score of 0 (in all 3 metrics) and is thus no better than a random partition of cells. Seurat performed well on data sets with equal cell type proportions (Geq) and on data sets where groups are identified by a large number of marker genes (DE5) whereas a substantial drop in ARI and homogeneity score can be observed on the remaining data sets. Seurat's NMI scores exhibit a similar but less pronounced pattern. Geometric sketching, which uses the same Louvain clustering algorithm as Seurat, behaves similarly. TSCAN performed better on synthetic than on real data sets (in all 3 metrics), while the opposite is true for RCA. The baseline algorithm RtsneKmeans yields remarkably accurate clusterings, especially on data sets with balanced cell type composition. On more difficult data sets, however, its accuracy drops significantly compared to several methods tailored to scRNA-seq analysis, especially in terms of ARI and homogeneity score. dropClust, on the other hand, achieved mean accuracy scores on synthetic data sets which are close to the baseline algorithm's ones (ARI 0.63 vs 0.57, homogeneity score 0.65 vs 0.63, NMI 0.67 vs 0.71).

Finally, we illustrate in Figure 2.4 how higher performance scores translate into a more meaningful representation of cell types.

2.3.2 Specter facilitates robust landmark-based clustering of single cells

In addition, we compared Specter to the original implementation of the landmark-based spectral clustering (LSC) algorithm and dissect the relative contribution of our hybrid landmark selection strategy, the clustering ensemble approach and the novel selective sampling scheme (see the "Methods" section) to the overall improvement in performance by Specter (using 50 ensemble members). We show results for three variants of Specter in which we either replace the k-means based landmark selection or the selective sampling approach by standard random sampling, or in which we omit the clustering ensemble step altogether. Figures 2.5 and 2.6 demonstrate the effectiveness of our adoptions and extensions of the original algorithm to the analysis of scRNA-seq data. Across all 24 simulated data sets, Specter achieved a higher ARI (mean ARI 0.89) than LSC (mean ARI 0.59) (Figure 2.5). In fact, even without the benefit of a clustering ensemble, further algorithmic adjustments implemented in Specter such as a modified bandwidth of the Gaussian kernel yielded an improvement over LSC on 19 out of 24 data sets. When disabling the clustering ensemble approach in Specter, however, its performance decreased consistently, on several data sets the decrease in ARI was substantial. Similarly, on 21 out of 24 data sets the selective sampling in Specter was more effective in terms of ARI than random sampling. On two instances with unbalanced cell type compositions (*pbmc*), the score more than doubled. Remarkably, coupled with random sampling (instead of selective sampling), the consensus clustering obtained from a clustering ensemble was often even less accurate than a single clustering.

The hybrid k-means based landmark selection led to an improvement in ARI on all but one data sets (Figure 2.6). In many cases this improvement was substantial, especially on difficult instances with unbalanced cell type compositions (*pbmc*, G*neq*).

In Supplemental Figure S4 we further addressed the dependence of Specter's accuracy on the number of ensemble members from which Specter computes a consensus clustering. Consistent with our observation in Figure 2.5, the clustering ensemble approach yielded on average more accurate results on the 24 simulated data sets than relying on a single clustering for each data set. Even a small number of ensemble members (e.g. 10) improved clustering accuracy substantially, while only minor improvements were achieved when increasing their number further to more than 20 ensemble members. Nevertheless, a clustering ensemble of size 200 yielded highest mean ARI with lowest score variance.

Finally, we demonstrate robustness of Specter to the choice of parameter γ that controls the bandwidth of the Gaussian kernel that is set differently in Specter compared to LSC (see the "Methods" section). Even though this parameter is randomly selected from interval $[0.1, 0.2]$ consistently across all 45 data sets in this benchmark, Supplemental Figure S5 shows that with very few exceptions choosing γ from different intervals would yield nearly identical results.

2.3.3 Specter is sensitive to rare cell populations

In this section, we evaluate Specter's sensitivity to rare cell populations. We devised three simulation experiments with increasing degree of difficulty. First, we repeated the experiment performed by Sinha et al. (2018) and randomly sampled a rare population of cells that comprise between 1% and 10% of total cells. More specifically, starting from two (equal size) groups of 2000 cells each that were simulated using Splatter (data set RareCellExp1 in Supplemental Table S1), we randomly downsample one group to comprise $1-10\%$ of the total

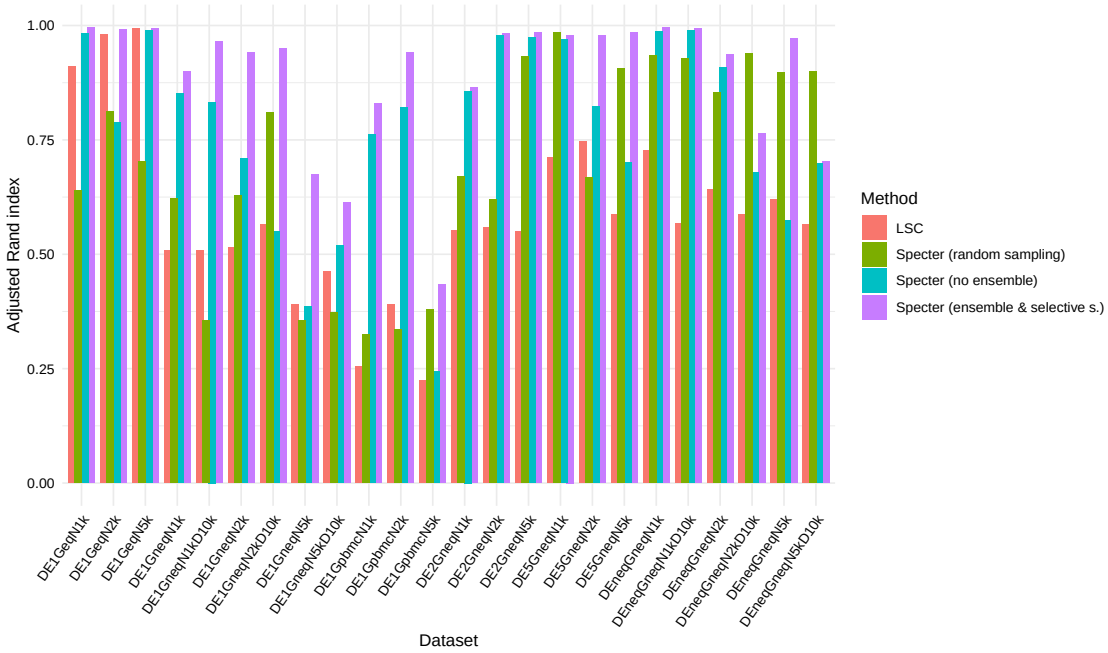

Figure 2.5: Improvements in Specter over LSC. The clustering accuracy of Specter using 50 ensemble members (ensemble & selective s.) is compared to the accuracy of the original implementation of the landmark-based spectral clustering algorithm (LSC) and two variants of Specter in which we either disable consensus clustering in Specter (no ensemble) or in which we replace the novel selective sampling in Specter (Algorithm 2) by random sampling. When no clustering ensemble is used (no ensemble), we set parameters to the median values of intervals probed by the ensemble scheme ($\gamma = 0.15, p = 9k\log(k)$).

number of cells. We repeat the experiment five times for each group and similar to Sinha et al. (2018) report the average F_1 score over the 10 runs in Figure 2.7 (top). The F_1 score denotes the harmonic mean of the recall and precision, which we define identically to Sinha et al. (2018) with respect to the predicted cluster with the largest number of rare cells. While several methods performed well on a sample of 10% of cells (SC3 being a notable exception), only Specter and Seurat are able to accurately detect a cell population that is composed of only 1% of cells. Additionally, we performed an experiment in which we randomly sampled cells from a group that is initially smaller (1,000 cells) than the second group (9,000 cells) (data set RareCellExp2 in Supplemental Table S1). Compared to the previous experiment, the rare population of cells will then occupy a smaller transcriptional space relative to the larger group, which may represent a more realistic, but also a more challenging scenario for clustering methods. Note that the smaller group initially consists of 10% of total cells and was therefore downsampled to comprise $1 - 5\%$ of cells. Again, each sampling experiment was repeated 10 times and average F_1 scores are shown in Figure 2.7 (bottom). Here, several methods obtained an F_1 score of close to 0 even when sampling 5% of cells, underlining the added difficulty of clustering unbalanced cell types. After further reducing the abundance of the rare cell type to 1%, only Specter achieved an almost perfect F_1 score (0.96), followed again by Seurat with an F_1 score of 0.78. In the most challenging scenario, we randomly downsampled naive cytotoxic or regulatory T cells that partly overlap in the Zhengmix4eq

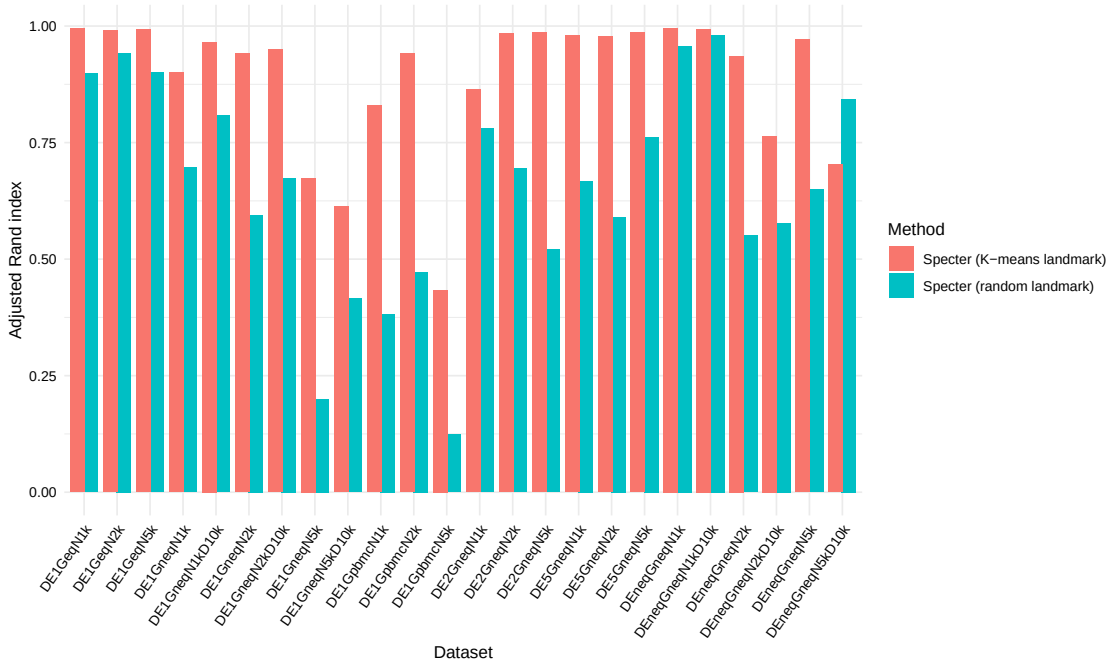

Figure 2.6: Comparison of landmark selection strategies. The clustering accuracy of Specter using our hybrid k-means based landmark selection strategy (K-means landmark) is compared to a variant of Specter in which we select landmarks uniformly at random.

data set (see Supplemental Figure S6) to comprise 1%-10% of the total number of cells and repeated this experiment five times for each group. Average F_1 scores are shown over the 10 runs in Supplemental Figure S7. Even though Specter consistently demonstrates highest accuracy among all methods, its F_1 score monotonically decreases from close to 1 for 10%, to 0.26 for just 1% of cells, highlighting the intrinsic difficulty of detecting rare cell types that are transcriptionally similar to more abundant cell populations.

Finally, we confirmed Specter's sensitivity to rare cell types on a rare population of inflammatory macrophages that was reported and experimentally validated by Hie et al. (2019b). In Hie et al. (2019b), the authors applied Louvain clustering to a geometric sketch of 20,000 cells sampled from a data set of 254,941 umbilical cord blood cells. In their experiments the authors observed that this rare subtype is invisible to Louvain clustering, the algorithm used by Seurat, unless cells are initially sampled evenly across transcriptional space to better balance the abundance of common and rare cell types. In contrast, Specter reveals a similar population of inflammatory macrophages characterized by the same set of marker genes *CD74*, *HLA-DRA*, *B2M* and *JUNB* (AUROC > 0.9) without any prior preprocessing (Figure 2.8).

2.3.4 Specter utilizes multi-modal data to resolve subtle transcriptomic differences

In this Section, we demonstrate the ability of Specter to utilize complementary information provided by multi-modal data to refine the clustering of single cells. More specifically, we re-analyzed two public data sets of 4,292 healthy human peripheral blood mononuclear cells (PBMC) (Mimitou et al., 2019) and 8,617 cord blood mononuclear cells (CBMC) (Stoeckius

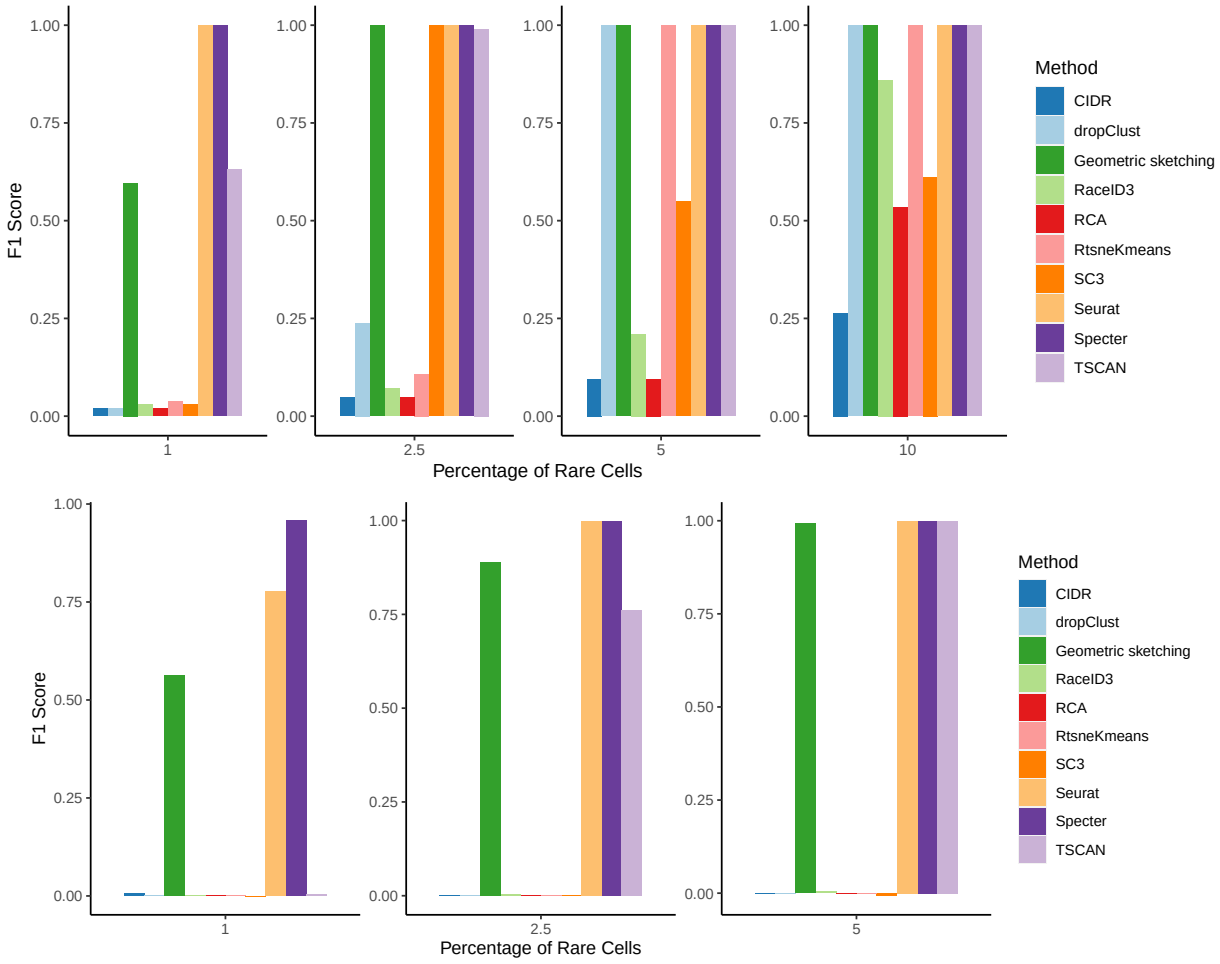

Figure 2.7: *top*: Sensitivity to rare cell types with equal starting abundances. 4000 cells from two equal size groups (2000 cells each) were simulated using Splatter. We randomly downsampled one group to comprise 1%, 2.5%, 5%, and 10% of the total number of cells. We repeated this experiment five times for each group and show the average F_1 score over the 10 runs. For geometric sketching, the average F_1 score was taken over 10 random trials with a sketch size of 10% of the full data. *bottom*: Sensitivity to rare cell types in initially smaller group. Cells were randomly sampled from the smaller of two simulated groups (1,000 and 9,000 cells) to comprise 1%, 2.5%, and 5% of the total number of cells. We show the average F_1 score over 10 runs of this experiment. For geometric sketching, the average F_1 score was taken over 10 random trials with a sketch size of 10% of the full data.

et al., 2017), for which both mRNA and protein marker expressions (ADT, antibody-derived tags) were measured simultaneously using CITE-seq (Stoeckius et al., 2017). In these experiments, the authors used 49 and 13 antibodies, respectively, that recognize cell-surface proteins used to classify different types of immune cells.

Consistent with previous analyses of CITE-seq data (Satija, 2019; Kim et al., 2020), we used the Seurat R package (Butler et al., 2018) to preprocess RNA and ADT counts. We normalized ADT expression using centered log-ratio (CLR) transformation and log-transformed RNA counts after adding a pseudocount of 1. After selecting the top 2,000

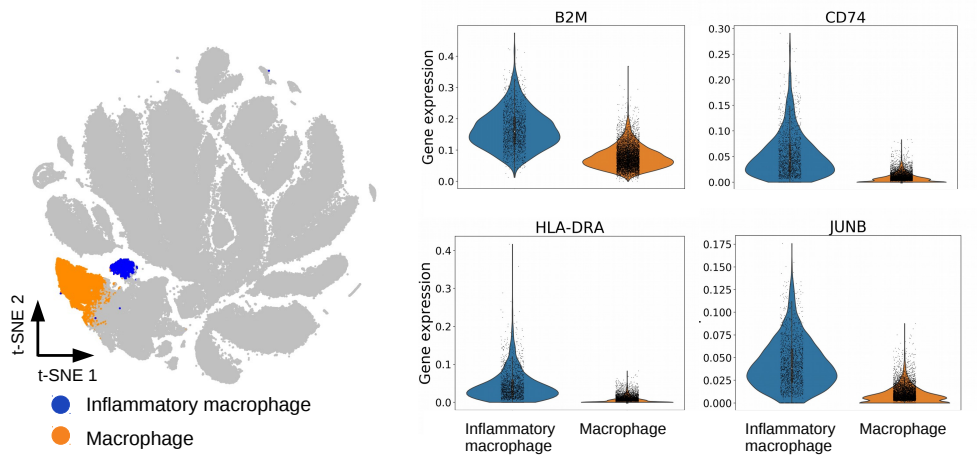

Figure 2.8: Clustering of 254,941 umbilical cord blood cell by Specter. Among macrophages defined by CD14 and CD68 marker gene expression, Specter detects a rare subpopulation of inflammatory macrophages that was recently discovered (Hie et al., 2019b) *(left)*. This rare subtype can be distinguished in Specter's clustering by the expression of the same set of inflammatory marker gene expression (*B2M*, *CD74*, *HLA-DRA*, and *JUNB*) used for its identification in Hie et al. (2019b).

most variable genes, the expression of each gene was scaled to have mean expression 0 and variance 1, followed by dimensionality reduction to 20 principal components.

Doublets in the PBMC data set were removed using the same cell hashing-based approach with identical parameters as in Kim et al. (2020). Similar to the analysis in Stoeckius et al. (2017), a putative cluster of doublets coexpressing different RNA and protein lineage marker were removed from further analysis. On the CBMC data we relied on the doublet removal of Seurat performed in a prior analysis (Satija, 2019) of this data set.

We annotate clusters based on differential expression of marker genes (Wilcoxon rank-sum test) for immune cell types listed in Table 2.2. The analysis of both data sets is documented at `https://github.com/canzarlab/Specter`.

On both data sets, both Seurat and Specter fail to accurately distinguish naive CD4 T cells and CD8 T cells based on transcriptomic data alone (Figures 2.9; Supplemental Figure S8). Many CD4-/CD8+ T cells identified by protein measurements (ADT) in the CBMC data set are wrongly grouped together with CD4 T cells by Seurat and Specter. Similarly, CD4 and CD8 T cells are mixed in the PBMC data set by both methods.

On the other hand, dendritic cells and megakaryocytes cannot be identified in the CBMC data set based on protein marker expression, see analysis using Seurat (Satija, 2019). Similarly, Figure 2.10 shows that ADT-based clustering by Specter is not able to separate CD14+ from FCGR3A+ Monocytes nor megakaryocytes from other cell types in the PBMC data set. This can be analogously observed in the clustering by Seurat (Supplemental Figure S8).

We therefore aimed to correct and improve the individual clusterings of RNA and surface marker protein measurements by combining the two distinct species through our clustering ensemble approach. In particular, Specter first produces an identical number of clusterings (here 200) for each modality. It then combines the transcriptome-based clusterings and the protein-based clusterings through a co-association approach (see the "Methods" section).

The joint clustering of RNA and protein expression by Specter profits from both modalities, yet differs from both unimodal analyses: On the PBMC data set, an ARI score of 0.78

Table 2.2: Markers used in the annotation of clusters in the CBMC and PBMC data sets. P-values indicate significance of differential expression according to a Wilcoxon rank-sum test between clusters inferred by Specter from the joint analysis of mRNA and surface protein expression.

Cell-type	Data set	Markers
CD8+CD27-	PBMC	*CD8A* (p = 3.1e-15), *CD8B* (p = 4.3e-6), low *CD27* ADT
CD8+CD27+	PBMC	*CD8B* (p = 3.2e-4), high *CD27* ADT
Naive CD4+ T	PBMC	*SELL* (Haining et al., 2008) (p = 2.6e-9)
CD4+CD27+	PBMC	*IL7R* (Colpitts et al., 2009) (p = 9.4e-11), high *CD27* ADT
CD4+CD27-DR+	PBMC	*IL7R* (Colpitts et al., 2009) (p = 4.4e-7), *NKG7* (Fonseka et al. 2018)(p = 1.2e-3), *GZMA* (Fonseka et al., 2018) (p = 2.0e-4)
CD4+CD27-DR-	PBMC	*IL7R* (Colpitts et al., 2009) (p = 1.4e-6), low expression of *NKG7* and *GZMA*; low *CD27* ADT.
CD14+ Mono	PBMC	*LYZ* (p = 7.5e-34), *CST3* (p = 1.4e-32)
FCGR3A+ Mono	PBMC	*FCGR3A* (p = 1.0e-9)
Megakaryocytes	PBMC	*PF4* (Lambert et al., 2016) (p = 1.2e-3)
NK	PBMC	*GNLY* (Ogawa et al., 2003) (p = 7.7e-21), *NKG7* (Turman et al., 1993)(p = 1.1e-15)
Dendritic cells	CBMC	*CST3* (Hruz et al., 2008) (p = 4.7e-29), *CD1C* ((Collin et al., 2013; Merad et al., 2013)) (p = 1.1e-27), and *FCER1A* (Hruz et al., 2008) (p = 1.3e-27)
Megakaryocytes	CBMC	*PF4* (Lambert et al., 2014) (p = 1.6e-25), *PPBP* (Sakurai et al., 2016) (p = 5.8e-24)

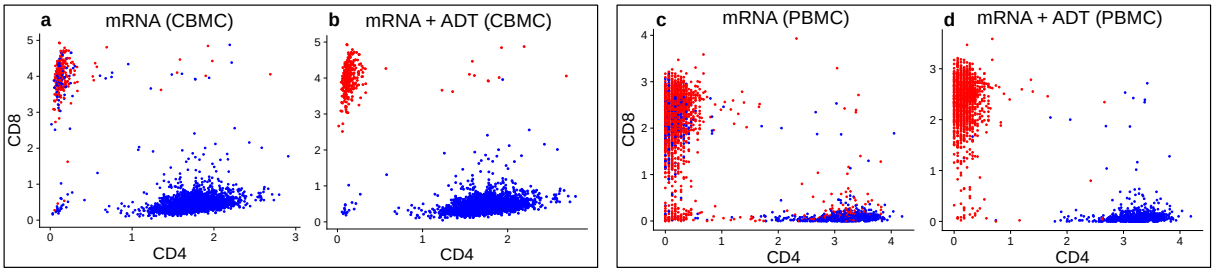

Figure 2.9: Comparison of unimodal and joint clustering by Specter. CBMCs (left box) and PBMCs (right box) with coordinates of protein expression (ADT) along CD4 and CD8 axis. Cells are clustered by Specter into CD4 T cells (blue) and CD8 T cells (red) either based on mRNA expression alone (**a, c**) or jointly from mRNA and surface protein expression (**b, d**). The mixing of CD4 T cells and CD8 T cells in the mRNA based clustering is corrected through the co-association of both modalities by Specter.

comparing multimodal and RNA-based clustering and a score of 0.72 between multimodal and ADT-based clustering indicate complementary aspects of cellular identity utilized in their joint clustering. On the CBMC data set, higher ARI scores of 0.87 and 0.91 between the multimodal clustering and RNA and ADT-based clusterings, respectively, reflect a higher agreement between the two modalities.

More specifically, the joint clustering of RNA and protein expression of CBM and PBM cells allows Specter to more accurately separate CD4 T cells and CD8 T cells compared to a simple transcriptome-based clustering (Figure 2.9). In contrast to ADT expression based clustering of PBM cells, the joint clustering of RNA and surface protein expression by

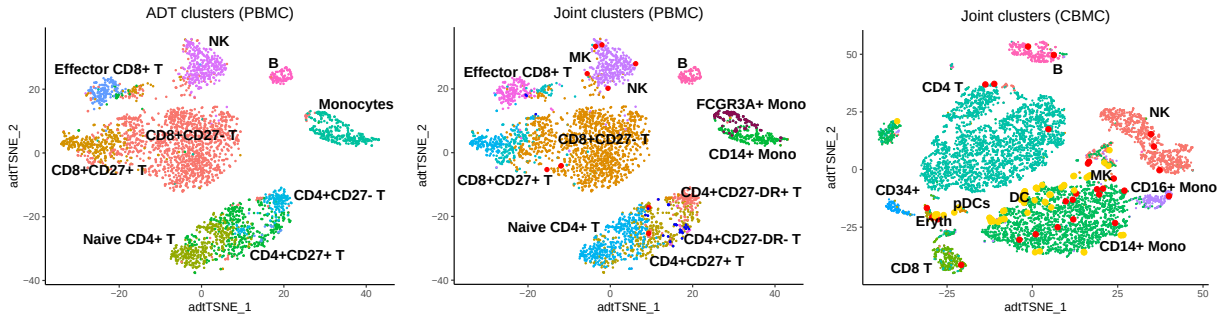

Figure 2.10: t-SNE visualization of clusters identified by Specter. Clusters of PBM cells were inferred from protein expression (ADT) alone (left) or from combined mRNA and protein expression (middle). In contrast to the joint clustering of both modalities, ADT-based clustering cannot discriminate CD14+ and FCGR3A+ Monocytes, does not detect megakaryocytes (red) and does not allow to discriminate between CD27-DR+ and CD27-DR- subpopulations of CD4+ T cells. The simultaneous clustering of RNA and protein expresssion in CBM cells (right) additionally reveals a rare population of megakaryocytes (red).

Specter correctly identifies megakaryocytes, CD14+, and FCGR3A+ Monocytes (Table 2.2 and Figure 2.10). In addition, only the combined clustering of ADT and RNA allows Specter to discriminate between CD27- DR+ and CD27- DR- subpopulations of CD4+ memory T cells. In contrast to the clustering of protein data of CBM cells, Specter also correctly detects dendritic cells and megakaryocytes based on the markers listed in Table 2.2 (see Figure 2.10).

We compare the joint clustering by Specter to the results of CiteFuse (v0.99.10) (Kim et al., 2020), a method that was recently proposed specifically for the computational analysis of single cell multimodal profiling data. As proposed initially for the combination of (bulk) genome-wide measurements across, e.g., patients (Wang et al., 2014), CiteFuse applies the similarity network fusion algorithm to combine RNA and ADT expression of single cells and then clusters the fused similarity matrix using spectral clustering. We ran CiteFuse as originally described in Kim et al. (2020) including the removal of doublets and the (internal) selection of highly variable genes.

Overall, the clusters of CBM and PBM cells as computed by Specter and CiteFuse are highly similar, as indicated by a high ARI score of 0.94 and 0.86 for the two data sets (Supplemental Figures S9 and S10). In both data sets, however, only Specter is able to identify a rare population of megakaryocytes (Table 2.2). Furthermore, in contrast to the analysis performed in Kim et al. (2020), CiteFuse was not able to discriminate between CD27- DR+ and CD27- DR- subpopulations of CD4+ memory T cells in the PBMC data set, neither when using identical parameters as in (Kim et al., 2020) nor when applying more conservative parameters in the doublet removal (parameters taken from CiteFuse tutorial (Lin and Kim, 2020)) (Supplemental Figure S11). The authors of Kim et al. (2020) attribute this discrepancy to a different selection of highly variable genes applied in an earlier version of the software used to produce the results in Kim et al. (2020).

The major advantage of Specter over CiteFuse is its speed and scalability. CiteFuse requires 15 minutes and nearly 2 hours to jointly cluster the 3,880 PBM cells and 7,895 CBM cells (after doublet removal), respectively, and is thus not expected to scale well on larger data sets due to the computational expensive fusion of networks. In contrast, Specter returns a high resolution clustering of the two data sets in just 20 and 50 seconds, respectively.

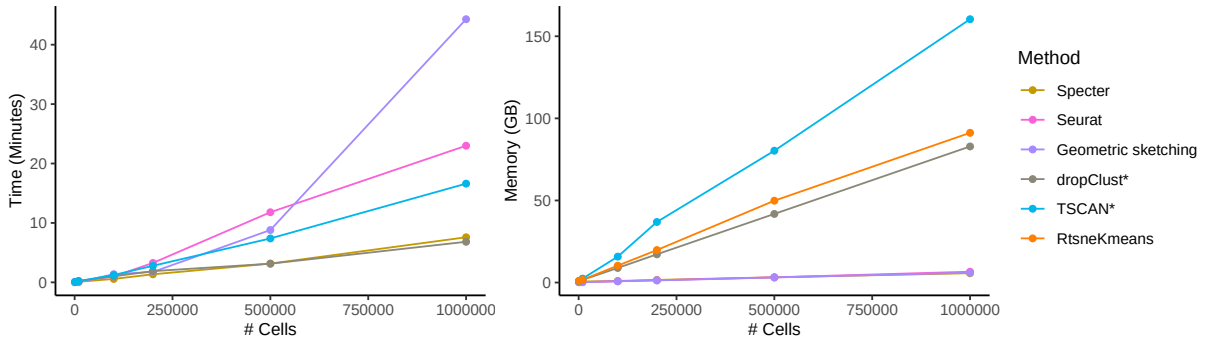

Figure 2.11: Runtime and peak memory usage as a function of sample size. Seurat was run with a call to the more efficient SCANPY implementation of the Louvain clustering algorithm. Running times exclude preprocessing for all methods except TSCAN and dropClust, whose implementation did not allow to isolate the core algorithm. Memory usage of Specter, Seurat, and geometric sketching are nearly identical and cannot be distinguished in this plot. For ease of visualization we show runtime results of method RtsneKmeans in Supplemental Figure S16.

2.3.5 Scalability

Here, we demonstrate the scalability of Specter to large single-cell data sets. To experimentally confirm the theoretical linear-time complexity of our algorithm, we devised different size simulated data set containing between 1,000 and 1 million cells (with characteristics DE1Geq, see Supplemental Table S1.). As expected (Cai and Chen, 2011), the landmark-based sparse representation of the data allows to compute a spectral embedding in linear time (see Supplemental Figure S12). Furthermore, the experiment confirms that our novel selective sampling strategy reduces the quadratic complexity of the hierarchical clustering step that reconciles multiple ensemble members (see the "Methods" section) to an overall linear dependence on the number of cells. As expected, the rate of increase in running time, i.e. the slope of the lines shown in Supplemental Figure S12, is larger when Specter includes multiple clusterings (here 20) in the ensemble scheme. More precisely, we observed a linear increase in running time with the size of the clustering ensemble, that is, with the number of independent runs of the core algorithm (Supplemental Figure S13). However, as shown in our experiments assessing the importance of individual algorithmic components in Specter, a relatively small number of runs is sufficient to improve accuracy of the resulting consensus clustering substantially. Even more, the independent computation of individual clusterings in an ensemble lend themselves to parallel processing. In Supplemental Figure S14 we therefore explored how the use of multiple threads can speed-up the clustering ensemble approach and thus counterbalance the inclusion of an increasing number of ensemble members. With just 4 threads, the time required to compute a consensus clustering from 50 individual clusterings of $100,000$ cells reduced from around 92 seconds to just 34 seconds. Increasing the number of threads further has a decreasing effect on total running time, reaching 15 seconds total computation time using 20 threads. Again, we observed a roughly linear increase in running time with increasing sample size for fixed number of threads (Supplemental Figure S15), where 4 threads reduced the running time of 50 runs in the clustering ensemble to a time that is nearly identical to the time a single threads needs to compute a consensus clustering from 20 ensemble members.

In Figure 2.11 we compared Specter's running time to all methods that ran successfully on the three largest real data sets. For all methods except TSCAN and dropClust we measured the running time of the core algorithm and exclude preprocessing. The time Specter required to preprocess the data (using a single thread), including log-transformation, the selection of highly variable genes, and principle component analysis, is negligible (Supplemental Tables S2). Seurat was run with a call to the more efficient SCANPY implementation of the Louvain clustering algorithm. Even in single-threaded mode, Specter's running time that included 20 individual clusterings of 1 million cells is with 7.6 min considerably faster than Seurat which required 23 min for a single Louvain-based clustering of the same set of cells (Supplemental Table S2). Note that 20 ensemble members were used by Specter in Figure 2.3 (and Supplemental Figures S1, S2) to achieve overall more accurate clusterings than competing methods. With just 4 threads Specter's running time further drops to 3.2 min (Supplemental Figure S15), whereas Seurat's clustering algorithm cannot be run with multiple threads. dropClust required 6.8 minutes to preprocess and cluster 1 million cells, but is not able to make use of multiple threads. The running time of geometric sketching increases the fastest while RtsneKmeans is as expected the slowest method (Supplemental Figure S16).

Finally, Supplemental Table S3 gives the CPU times in minutes on the three largest real data sets used in this study. Again, we excluded preprocessing for all methods except TSCAN and dropClust. We additionally report the total running time of Specter including all prior preprocessing. In this analysis of real data sets, we exploited the full performance potential of Specter and used 20 threads to compute consensus clusterings from 50 individual runs, which outperformed all other methods in terms of accuracy in Figure 2.3 and Supplemental Figures S1, S2. In this setting, Specter required around 15 min to cluster 2 million cells (23 min including single-threaded preprocessing) and was 5-10 times faster than Seurat that is not able to utilize multiple threads. On the largest data set, dropClust was with just 12 minutes of total computation time using just a single thread the fastest method. In contrast to Specter, however, dropClust considers only around 1% of the data (20,000 cells) and its simplified model comes at the cost of a substantial loss in accuracy (see Figure 2.3 and Supplemental Figures S1, S2). Again, RtsneKmeans is the slowest among methods that terminate successfully on these large data sets.

Furthermore, Figure 2.11 shows peak memory usage as a function of number of cells on the same simulated data sets used to evaluate runtime performance. Together with Seurat and geometric sketching, Specter required the least amount of memory (less than 7 GB for 1 million cells), while memory usage of methods TSCAN and dropClust increased rapidly for data sets containing more than 200,000 cells.

2.3.6 Publicly available data used in this study

The original publication of data sets used in this study to assess the accuracy of Specter in comparison to existing methods are listed in Table 2.1. The real data sets in Duò et al. (2018) were downloaded from `https://github.com/markrobinsonuzh/scRNAseq_clustering_comparison`. All other real data sets smaller than 15,000 cells were downloaded from `https://hemberg-lab.github.io/scRNA.seq.datasets`, the 3 largest data sets from `http://mousebrain.org` (*CNS*), `http://dropviz.org` (*saunders*), and `https://oncoscape.v3.sttrcancer.org/atlas.gs.washington.edu.mouse.rna/downloads` (*trapnell*). The umbilical cord blood cell data (Hie et al., 2019b) were downloaded from `http://cb.csail.mit.edu/cb/geosketch`.

The Specter software is available at `https://github.com/canzarlab/Specter` and as Supplemental Code under the open source MIT license. The Specter repository also includes all code necessary to reproduce the results of this manuscript as well as a step-by-step documentation of the analysis of the PBMC and CBMC CITE-seq data sets (Stoeckius et al., 2017; Mimitou et al., 2019) as described in this study.

2.4 Conclusions

We have introduced Specter, a novel method that identifies transcriptionally distinct sets of cells with substantially higher accuracy than existing methods. We adopt and extend algorithmic innovations from spectral clustering, to make this powerful methodology accessible to the analysis of modern single-cell RNA-seq data sets.

We have demonstrated the superior performance of Specter across a comprehensive set of public and simulated scRNA-seq data sets and illustrated that an overall higher accuracy also implicates an increased sensitivity towards rare cell types. At the same time, its linear time complexity and practical efficiency makes Specter particularly well-suited for the analysis of large scRNA-seq data sets. Besides technological advances, the integration of cells from multiple experiments spanning different tissues or diseases may yield data sets with massive numbers of cells. Coupled with data integration methods such as Scanorama (Hie et al., 2019a) or Harmony (Korsunsky et al., 2019) that can remove, e.g., tissue-specific differences, Specter can help to leverage such reference data sets to reveal hidden cell types or states. When combining different samples from the same experiment, simpler linear methods such as ComBat (Johnson et al., 2006) might be preferable (Luecken and Theis, 2019) to correct for batch effects between samples prior to identifying groups of cell with distinct gene expression profiles using Specter.

Furthermore, we have illustrated how the flexibility of its underlying optimization model allows Specter to harness multimodal omics measurements of single cells to resolve subtle transcriptomic differences between subpopulations of cells. The application of our cluster ensemble scheme to the joint analysis of multimodal CITE-seq data sets yielded a slightly more fine-grained distinction of cell (sub-)populations compared to the recently proposed multimodal clustering method CiteFuse. More importantly, in contrast to CiteFuse whose running time increased ≈ 8 fold after doubling the number of cells, Specter will scale well to much larger data sets produced by droplet-based approaches that can measure multiple modalities of up to millions of cells together. While the consensus clustering approach applied by Specter can in principle integrate the ensemble of clusterings generated from various molecular features, this work has focused on the combination of mRNA and protein marker expression as measured by CITE-seq or REAP-seq (Peterson et al., 2017). The practical suitability and potential limitations as well as necessary refinements of this strategy when applied to other assays that simultaneously measure, for example, accessible chromatin and gene expression (Cao et al., 2018), or more than two modalities at the same time (Clark et al., 2018), will need to be addressed in future experiments. Taken together, we believe that Specter will be useful in transforming massive amounts of (multiple) measurements of molecular information in individual cells to a better understanding of cellular identity and function in health and disease.

Chapter 3

Spherical sketching of large single-cell datasets

In the previous chapter we proposed an algorithm to handle big data for clustering. In this chapter we provide a general scheme to deal with big data. This chapter is adapted with minimal modification from: Van Hoan Do, Khaled Elbassioni, and Stefan Canzar. *Sphetcher: Spherical thresholding improves sketching of single-cell transcriptomic heterogeneity*. iScience, 23(6):101126, 2020.

In practice, methods are often run on a smaller subset of the data to bridge the gap between the scalability of the algorithm and the volume of the data (Hie et al., 2019b). Recently, geometric sketching was introduced as an alternative to uniform subsampling. It selects a subset of cells (the sketch) that evenly cover the transcriptomic space occupied by the original dataset, to accelerate downstream analyses and highlight rare cell types. Here, we propose algorithm Sphetcher that makes use of the thresholding technique to efficiently pick representative cells within spheres (as opposed to the typically used equal-sized boxes) that cover the entire transcriptomic space. We show that the spherical sketch computed by Sphetcher constitutes a more accurate representation of the original transcriptomic landscape. Our optimization scheme allows to include fairness aspects that can encode prior biological or experimental knowledge. We show how a fair sampling can inform the inference of the trajectory of human skeletal muscle myoblast differentiation. Sphetcher requires only 16 minutes to compute a sketch for a mouse embryonic dataset comprising 2 million cells.

3.1 Methods

3.1.1 Overview of our spherical sketching algorithm

Given a large scRNA-seq dataset, we seek to select a subset of cells, a so-called *sketch* (Hie et al., 2019b), that evenly represents the geometry of the transcriptional space occupied by the original data. As originally proposed in Hie et al. (2019b), we intuitively aim at capturing the transcriptional heterogeneity of single cells by removing predominantly cells that show similar expression patterns to other cells while preserving rare cell states. A sketch of a given size represents the full data well if every original cell is close to a cell in the sketch, according to some measure of distance between two cells. In other words, spheres of a small radius centered at each cell in the sketch must contain, or *cover*, every cell in the full dataset. The smaller the radius, the better the sketch represents the original transcriptional space.

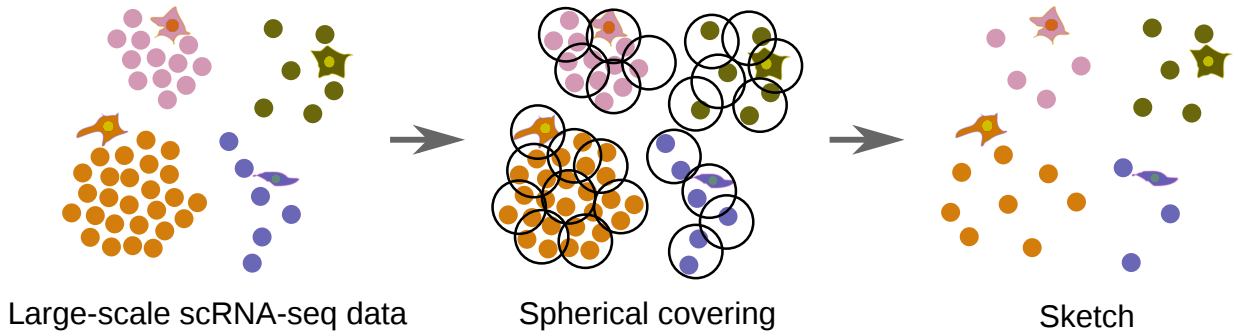

Figure 3.1: Overview of Sphetcher. For a (large) scRNA-seq dataset (left), Sphetcher uses a disk-friendly greedy algorithm to compute a smallest size set of spheres of a fixed radius that cover all cells (middle). It guesses the smallest possible radius such that a given number of spheres of that radius suffice to cover all cells. One representative cell (the center) from each sphere is selected into the final spherical sketch (right).

Our algorithm implemented in software tool Sphetcher guesses the smallest possible radius for which a sketch of a given size exists that covers all remaining cells with spheres of this radius (Figure 3.1). For each guess, it computes the smallest size sketch that covers all cells and tries a smaller or larger radius in the next iteration if the resulting sketch contains too few or too many cells, respectively. It computes the smallest sketch that covers all cells using a greedy set cover approach: In each iteration, it adds the cell to the sketch that contains the largest number of yet uncovered cells within the given distance. We employ the disk-friendly greedy (DFG) algorithm developed in Cormode et al. (2010) that scales to very large scRNA-seq datasets. For very large datasets, the spherical sketching approach is combined with a prior grid sampling that we show increases the radius of covering spheres by only a small factor.

In addition, our greedy algorithm can incorporate prior categorical information on, e.g., biological cell types or collection time point of cells. In a *fairness*-inspired model it selects at least a given number of representatives from each class into the sketch. We provided a theoretical analysis that shows that if we are willing to include slightly more cells in the sketch, our greedy algorithm is guaranteed to find the covering of cells with spheres with optimal, that is, with smallest possible radius. Furthermore, we gave theoretical justification for the practical performance of our greedy set cover approach and its robustness to noise present in scRNA-seq data.

3.1.2 Sketching scRNA-seq as k-center problem

Given a large scRNA-seq dataset, we seek to select a subset of cells, a so-called *sketch* (Hie et al., 2019b), that evenly represents the geometry of the transcriptional space occupied by the original data. As originally proposed in Hie et al. (2019b), we use the *Hausdorff distance* to measure how well the sketch captures the transcriptional heterogeneity in the data. Given n data points $X = \{x_1, x_2, \ldots, x_n\}$ representing the m-dimensional gene expression measurements $x_i \in \mathbb{R}^m$ of n individual cells, and a metric d that measures the dissimilarity between pairs of cells, the Hausdorff distances between a sketch $X_S \subseteq X$ and the full dataset

is given by:

$$d_H(X_S, X) = \max_{x \in X} \left\{ \min_{y \in X_S} d(x, y) \right\} \tag{3.1}$$

A sketch achieves a small Hausdorff distance if it includes for every cell in the original dataset a cell that is close to it in gene expression space. Finding a best sketch of size k, i.e. a sketch that minimizes the Hausdorff distance is known as the metric k-center problem in the combinatorial optimization literature. It is known to be NP-hard but a solution with Hausdorff distance at most 2 times the optimal distance can be found by a simple greedy strategy: In each iteration, pick the point farthest away from the current set of centers and add it as a new center. Although this greedy approach has time complexity $O(nk)$, it does not scale efficiently to large scRNA-seq datasets that require a larger number of cells k to be accurately represented.

3.1.3 A thresholding algorithm

To find a sketch of size k with small Hausdorff distance (3.1) to a single-cell dataset, we employ the *thresholding* technique that was originally proposed for the design of approximation algorithms for bottleneck problems (Hochbaum and Shmoys, 1986). In essence, we are guessing the optimal distance in (3.1) and for every guess L try to find a feasible solution, that is, a subset of cells of cardinality at most k such that spheres of radius L centered at cells in the subset cover all remaining cells. Then the smallest L^* for which such a feasible sketch exists denotes the optimal solution. We model the problem of finding the smallest set of cells such that the maximal distances from any other cell to the subset is at most a given threshold L as a set cover problem, $\textsc{SetCover}_X(L)$: Given a universe $\mathcal{U} = X$ of n data points, we build a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ of n subsets of $\mathcal{U}$ by including in each set S_i all points that lie within distance of L from x_i, i.e. $S_i = \{x_j \mid d(x_i, x_j) \leq L\}$. Then the minimum number of sets in $\mathcal{S}$ that cover every element of the universe corresponds to a smallest subset of points covering all remaining points with spheres of radius L.

A widely used algorithm for the set cover problem is based on a greedy strategy (Johnson, 1974): Starting from an empty set, in each iteration pick the set in $\mathcal{S}$ that covers the largest number of elements yet uncovered and add it to the solution. The greedy algorithm is guaranteed to find a cover which is within a logarithmic factor of the optimal solution (Johnson, 1974). Moreover, it has been observed across a wide range of instances that the greedy algorithm produces solutions close to the optimum. A direct implementation of the greedy algorithm, however, scales poorly to large scRNA-seq datasets. We therefore employ the disk-friendly greedy (DFG) algorithm developed in Cormode et al. (2010) for very large datasets. It achieves a dramatic performance improvement over the standard greedy algorithm by applying a geometric scale bucketing approximation. Furthermore, the DFG algorithm runs in linear time with respect to the total size of candidate sets, i.e in $O(\sum_i |S_i|)$, while guaranteeing to output a set cover which is within a logarithmic factor of the optimum. More precisely, the algorithm allows to choose a parameter p that represents a trade-off between the running time (which is $O((1 + \frac{1}{p-1}) \sum_i |S_i|)$) and the approximation ratio (which is $1 + p \ln n$). The complete algorithm is summarized in Algorithm 5. Let us denote by $\textsc{Greedy}(\mathrm{L})$ the set cover returned by the greedy algorithm when applied to sets $S_i = \{x_j \mid d(x_i, x_j) \leq L\}$, and let $\widetilde{L}(k) := \min\{L \mid \textsc{Greedy}(L) \text{ has size at most } k\}$ which can be found by a logarithmic number of calls to the greedy algorithm via binary search: If

GREEDY(L) is at most k, we decrease the threshold, otherwise we increase it (halving the length of the search interval in both cases), until the radius L lies in an interval of size at most ε.

Algorithm 5: Sphetcher

Input: Dataset $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^m$, size of the sketch k, and precision ε.
Initialization: $L_{\min} = 0$, $L_{\max} = \max_{i,j} d(x_i, x_j)$.
while $L_{max} - L_{min} > \varepsilon$ **do**
  $L \leftarrow (L_{\min} + L_{\max})/2$
  Solve SETCOVER$_X(L)$ using the DFG algorithm.
  if $|\text{GREEDY}(L)| \leq k$ **then**
    $L_{\max} \leftarrow L$
  else
    $L_{\min} \leftarrow L$
  end
end
Output: $X_S = \{x_i | S_i \in \text{GREEDY}(L)\}$.

If we are willing to increase the size of X_S by a logarithmic factor, Algorithm 5 is guaranteed to return a sketch with optimal Hausdorff distance.

Theorem 1. *Let L^* be the optimal distance in (3.1) for $|X_S| = k$. If we run the thresholding approach for $|X_S| = k\ln(n)$, then the solution we obtain has Hausdorff distance at most L^*. In other words, $\widetilde{L}(k\ln(n)) \leq L^*$.*

Proof. By definition of L^*, SETCOVER$_X(L^*)$ has size at most k. Thus, by the known approximation factor of the greedy algorithm, GREEDY(L^*) has size at most $k\ln(n)$, which implies by the definition of $\widetilde{L}(k\ln(n))$ that $\widetilde{L}(k\ln(n)) \leq L^*$. □

3.1.4 Grid sampling with guarantees

For datasets much larger than 100,000 cells, we apply a hybrid strategy to reduce the computational cost of determining the neighborhood of each point in Algorithm 5. To this end, we divide the space into equal-sized boxes from which we pick one point at random. In contrast to geometric sketching, we do not attempt to optimally define boxes in each dimension, but leave it to the subsequent thresholding algorithm to properly cover the space by spheres. In fact, we show that if we carefully choose the applied threshold taking into account the size of the grid, our hybrid sampling strategy increases the Hausdorff distance by at most a factor of $(1+\varepsilon)$, where $\varepsilon > 0$ controls the size of the grid.

Let SETCOVER$_X(L, Z)$ denote an optimal set covering all the points in X with spheres of radius L whose centers are chosen from $Z \subseteq X$. Let GREEDY(L, Z) denote the set obtained by the greedy algorithm described above covering all the points in X with spheres of radius L whose centers are chosen from $Z \subseteq X$. We know that $|\text{GREEDY}(L, Z)| \leq |\text{SETCOVER}_X(L, Z)|\ln(n)$, where $n = |X|$. Let $L_{\min}$ be the minimum distance between two points in X and $L_{\max}$ be the maximum distance between two points in X. Let I be the smallest integer such that $(1+\varepsilon)^I L_{\min} \geq L_{\max}$. Our hybrid algorithm that carefully combines grid sampling with the thresholding approach is given in Algorithm 6 (Sphetcher-H).

Algorithm 6: Sphetcher-H

1 **Input**: Dataset $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^m$, size of the sketch k, and $\varepsilon > 0$.
2 **Initialization**: $L_{\min} = \min_{i,j} d(x_i, x_j)$, an integer I as defined before.
3 **for** $i = 0, \ldots, I$ **do**
4 $\quad L \leftarrow (1+\varepsilon)^i L_{\min}$
5 $\quad$ Partition the space into a uniform grid G(L) of size $\varepsilon L/\sqrt{m}$
6 $\quad$ Let $Z(L) \subseteq X$ be the set obtained by choosing one point in each non-empty cell
7 $\quad Y(L) \leftarrow \textsc{Greedy}\big(\big(1+(1+\varepsilon)\varepsilon\big)L, Z(L)\big)$
8 **end**
9 **Output**: $Y(\hat{L}(k))$, where $\hat{L}(k) = \min\{L : |Y(L)| \leq k\}$.

The following theorem limits the increase in Hausdorff distance through Sphetcher-H by at most a factor of $(1+\varepsilon)$.

Theorem 2. *Let L^* be the Hausdorff distance $d_H(X_S, X)$ between X and an optimal set X_S of size k, then $d_H(Y(\hat{L}(k\ln(n))), X) \leq (1+\varepsilon)L^*$.*

Proof. Let L be the distance set in the for loop (Algorithm 6: steps 3 to 7) such that $L^* \leq L < (1+\varepsilon)L^*$. By definition of L^*, we know that $|\textsc{SetCover}_X(L^*, X)| \leq k$. So, let us write $\textsc{SetCover}_X(L^*, X) = X_S := \{x_1, ..., x_k\}$. Let $X'_S = \{x'_1, ..., x'_k\} \subseteq Z(L)$ be chosen such that x'_i lies in the same cell of the grid $G(L)$ as x_i. Hence, $d_H(x_i, x'_i) \leq \varepsilon L$ implies that

$$d_H(X_S, X'_S) \leq \varepsilon L < (1+\varepsilon)\varepsilon L^*.$$

Thus for any point $x \in X$, we have

$$d_H(x, X'_S) \leq d_H(x, X_S) + d_H(X_S, X'_S) \leq (1+(1+\varepsilon)\varepsilon)L^* \leq (1+(1+\varepsilon)\varepsilon)L.$$

It follows that $|\textsc{SetCover}_X((1+(1+\varepsilon)\varepsilon)L, Z(L))| \leq k$ and hence,

$$|\textsc{Greedy}((1+(1+\varepsilon)\varepsilon)L, Z(L))| \leq k\ln(n),$$

that is, $|Y(L)| \leq k\ln(n)$. By definition of $\hat{L}$, we have $\hat{L} \leq L < (1+\varepsilon)L^*$. □

3.1.5 Fair sampling

One of the advantages of our model is its flexibility to incorporate fairness aspects. For example, assume we have prior knowledge of (some) of the cell types present in the sample. Cells might have been pre-sorted, and some cell types such as T cell subtypes are well characterized and can be identified based on known markers, without relying on an unsupervised clustering of the data. Furthermore, when reusing scRNA-seq datasets shared through repositories or data archives, the annotation of cell types, i.e. their labels, are typically provided as part of the original study. Similarly, in time series studies of gene expression, cells are collected at different time points which can supervise the sketching algorithm to preferentially select cells for which collection time point and transcriptomic state agree.

Our goal is to use prior categorical information on, e.g., biological cell types or collection time point to guide the selection of cells into a representative sketch, without fully relying on the correctness of cell type labels nor their synchronous progression through biological

processes. We incorporate prior categorical information as *covering constraints* into our model: We seek to select a subset of cells that represent the geometric space of the original data according to (3.1) but at the same time contain at least a given number of representatives from each class. More formally, let $X_1, X_2, \ldots, X_m \subseteq X$ denote known clusters that do not necessarily partition the whole dataset X, we want to sample k cells that contain at least $l_i \in \mathbb{N}^+$ cells from each X_i, for all $i = 1, 2, \ldots, m$, while minimizing the Hausdorff distance of the sketch to the original dataset. This generalization of the k-center problem is similar to the *colorful k-center* problem, which does not require to include class members into the sketch but instead a certain number of elements from each class need to be covered by spheres around selected centers. For the colorful k-center problem a constant approximation in the Euclidean plane was recently introduced (Bandyapadhyay et al., 2019). In Anegg et al. (2020), the authors study a variant of this problem in which classes are allowed to overlap. Neither of the proposed algorithms is directly applicable to scRNA-seq data, due to low-dimensionality assumptions or the use of the ellipsoid method, respectively.

If $l_i = 1$, for all $i = 1, \ldots, m$, we have hitting set constraints $X_S \cap X_i \neq \emptyset$, $i = 1, \ldots, m$, which can be modeled as m additional elements in the universe of our set cover formulation of the problem. Given a threshold L, the corresponding set cover problem $(\mathcal{U}, \mathcal{S})$ is $\mathcal{U} = \{x_1, \ldots, x_n, X_1, \ldots, X_m\}$ and $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ with $S_i = [x_i] \cup \{X_j \mid x_i \in X_j\}$. Here $[x_i]$ contains x_i and its neighbors within distance L. Picking a set S_i into our set cover solution now does not only cover all cells within distance L of x_i, but x_i also hits all clusters $\{X_j \mid x_i \in X_j\}$. Having cast the constrained sampling problem as an instance of our thresholding framework, we solve it by the same algorithm (Algorithm 5). For general $l_i \in \mathbb{N}^+$, we simply partition X_i into l_i parts and apply the above approach, which however is no longer guaranteed to obtain the optimal Hausdorff distance.

3.1.6 Set cover under perturbation

This section provides the theoretical insight for the practical performance of the greedy set cover approach and its robustness to noise present in, e.g., scRNA-seq data. In step 5 of Algorithm 5 we need to construct the neighborhood for every point x_i that contains all points within a given distance threshold. Due to noise, the true distances will be slightly perturbed and yield imprecise estimates of neighborhoods. Since an instance to our set cover formulation contains a set for the neighborhood of each point, error-prone neighborhoods will affect our (greedy) search for the set with the largest number of uncovered elements. Here, we show that as long as we are able to pick a set with large enough number of uncovered elements, we can essentially preserve the approximation guarantee. More precisely, denote by C_t the set of elements covered *after* t iterations of greedy ($C_0 = \emptyset$). Assume that in each iteration t, errors in the distances prevent us from finding the set S_t^* with the maximum value of $|S_i \backslash C_{t-1}|$, but instead we select a set S_t such that $E(|S_t \backslash C_{t-1}|) \geq c \max_i |S_i \backslash C_{t-1}|$ for some constant c, where $E(X)$ denotes the expected value of random variable X. We show that with high probability, we will find a set cover within $2\ln(n)/c$ the size of an optimal solution, which differs only by a constant factor from the approximation guarantee of the (precise) greedy algorithm. Note that inapproximability results (Slavík, 1997) show that the greedy algorithm is essentially the best-possible polynomial time approximation algorithm for set cover up to lower order terms. Let $\mathcal{U}$ be the whole set of elements of size n. We have the following theorem.

Theorem 3. *If an iterative algorithm always chooses a set S_t to add to the current solution with*

$$E(|S_t \backslash C_{t-1}| \ \big| C_{t-1}) \geq c \max_i |S_i \backslash C_{t-1}|,$$

for $c \leq 1$, then with (high) probability $1 - \frac{1}{n}$ it returns a set cover that is larger than the optimum set cover by a factor of at most $2\ln(n)/c$.

Proof. Let the number of sets in the optimal solution be σ. We know that at each iteration there is some set that covers at least $|\mathcal{U} \backslash C_t|/\sigma$ new elements. It follows that

$$E(|\mathcal{U} \backslash C_{t+1}| \ \big| C_t) = |\mathcal{U} \backslash C_t| - E(|S_{t+1} \backslash C_t| \ \big| C_t) \leq |\mathcal{U} \backslash C_t| - c \max_i |S_i \backslash C_t| \leq \Big(1 - \frac{c}{\sigma}\Big) |\mathcal{U} \backslash C_t|.$$

Now taking the expectation over all possibilities for C_t we get

$$E(|\mathcal{U} \backslash C_{t+1}|) \leq \Big(1 - \frac{c}{\sigma}\Big) E(|\mathcal{U} \backslash C_t|),$$

and iterating we end up with

$$E(|\mathcal{U} \backslash C_t|) \leq |\mathcal{U}| \Big(1 - \frac{c}{\delta}\Big)^t \leq n e^{-tc/\sigma}.$$

Setting $t = 2\sigma \ln(n)/c$ implies that $E(|\mathcal{U} \backslash C_t|) \leq \frac{1}{n}$, and hence by Markov's Inequality:

$$\Pr(|\mathcal{U} \backslash C_t| \geq 1) \leq E(|\mathcal{U} \backslash C_t|) \leq \frac{1}{n}.$$

Thus, with probability at least $1 - \frac{1}{n}$, the sets we selected form a set cover. □

3.2 Results

We have implemented Algorithms 5 and 6 along with a fair sampling option in software tool Sphetcher in C++. Only on datasets zeiselCNS and saunders we apply our hybrid strategy Sphetcher-H (Algorithm 6) but refer to it simply as Sphetcher throughout the main text. Unless stated otherwise, Sphetcher uses Pearson correlation as distance metric d, and we set the precision $\varepsilon = 10^{-4}$ in Algorithm 5. Note that throughout this work, the size of our spherical sketch denotes the actual number of cells rather than their logarithmic approximation in Theorem 1.

Data and evaluation

All data were uniformly preprocessed by natural log-transformation of gene counts (after adding a pseudo-count of 1) followed by projection to 100 principle components.

We measure how well a sketch represents the original transcriptomic space by the robust Hausdorff distance. Compared to the classical definition of the Hausdorff distance, the robust variant of the distance between a sketch $X_S \subseteq X$ and the full dataset is less sensitive to outliers (Huttenlocher et al., 1993):

$$d_{HK}(X_S, X) = K^{th}_{x \in X} \Big\{ \min_{y \in X_S} d(x, y) \Big\}, \tag{3.2}$$

where $K^{th}_{x \in X}$ denotes the Kth largest distance to an element in X.

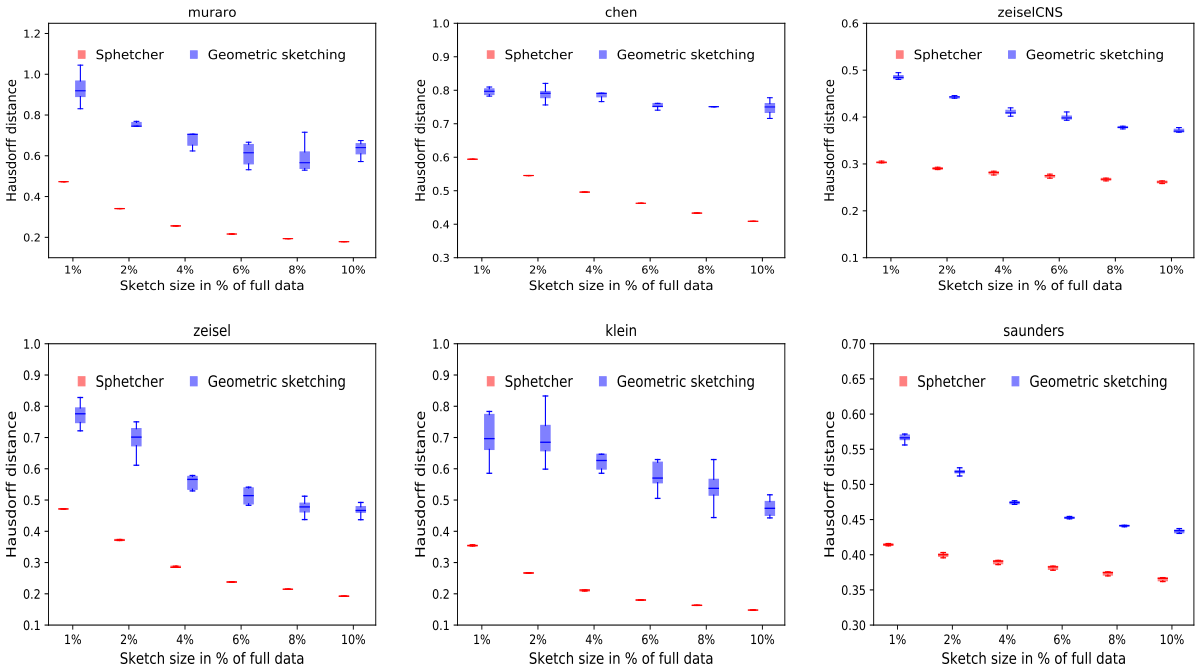

Figure 3.2: Comparison of Hausdorff distances. The spherical sketch computed by Sphetcher exhibits consistently smaller Hausdorff distances to the full dataset than geometric sketching, across datasets and sketch sizes. For each sketch size, the results of 10 random trials are shown. Supplemental Figure S18 shows Hausdorff distances achieved by our naïve grid sampling strategy on datasets zeiselCNS and saunders.

3.2.1 Sphetcher more accurately sketches the transcriptomic space

To evaluate how well the *spherical sketch* computed by our method Sphetcher represents the original transcriptomic space, we use the same robust Hausdorff distance measure as Hie et al. (2019b). Intuitively, a small Hausdorff distance between a sketch and a full dataset indicates an accurate representation that contains for every cell in the original data a close cell in the sketch. We compare our sketch to the *geometric sketch* computed by Hie et al. (2019b), which the authors demonstrated to consistently achieve smaller Hausdorff distances than uniform sampling and data-dependent sampling methods SRS and k-means++. The geometric sketch computed in Hie et al. (2019b) seeks to minimize the same objective function but simplifies the approximation of the geometric space by equal-sized boxes rather than spheres. We benchmark Sphetcher on 6 public single-cell datasets from mouse and human that vary in size and number of cell populations: human pancreas (*muraro*) (Muraro et al., 2016) with 2126 cells, 10 populations; mouse embryonic stem cells (*klein*) (Klein et al., 2015) with 2717 cells, 4 populations; mouse cortex and hippocampus (*zeisel*) (Zeisel et al., 2015) with 3005 cells, 9 populations; mouse hypothalamus (*chen*) (Chen et al., 2017) with 14,437 cells, 47 populations; mouse nervous system (*zeiselCNS*) (Zeisel et al., 2018) with 465,281 cells, 7 populations; and adult mouse brain (*saunders*) (Saunders et al., 2018) with 665,858 cells and 11 populations. Figure 3.2 shows the Hausdorff distances of 10 random trials on sketch sizes ranging from 1% to 10% of the full dataset. Values reported here can deviate slightly from the original publication (Hie et al., 2019b) due to different preprocessing. Our

sampling approach based on spheres results in sketches that consistently lead to smaller Hausdorff distances, across datasets and sketch sizes. As expected, larger sketches yield smaller Hausdorff distances, but across all datasets the geometric sketch based on 10% of the data does not represent the full data as well as our spherical sketch with just 1% of the data. In addition, sketches computed by Sphetcher exhibit a considerably smaller variability over the random trials (Supplemental Figure S17). While the geometric sketch randomly picks a cell in each box, Sphetcher's only random decision is in breaking ties between equal-sized sets during the greedy set cover computation. Remarkably, our naïve grid sampling strategy alone, which is part of our hybrid alternative for very large datasets, achieves competitive Hausdorff distances on datasets zeiselCNS and saunders, especially for small sketch sizes (Supplemental Figure S18).

3.2.2 Clustering of spherical sketches facilitates cell type identification

A common goal in scRNA-seq data analysis is to discover and characterise cell types, typically through clustering methods. The quality of the clustering therefore plays a critical role in biological discovery. The compact size of a geometric or spherical sketch that accurately summarizes the transcriptional heterogeneity in the full data facilitates such downstream analyses. Furthermore, Hie et al. (2019b) observed that a more balanced composition of abundant and rare cell types in a geometric sketch allows to better distinguish between cell types compared to a uniform sampling approach. Here, we apply a similar strategy as in Hie et al. (2019b) to evaluate the capability of a standard clustering algorithm to distinguish cell types based on our spherical sketch as compared to the geometric sketch. We first cluster the sketches using the graph-based Louvain algorithm (Blondel et al., 2008) and then propagate the labels to the remaining cells by k-nearest neighbor classification. We use the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) to measure the similarity between the inferred clusterings and the ground truth clustering which is based on the biological cell types taken from the original study. Hie et al. (2019b) demonstrated that unsupervised clustering of geometric sketches consistently outperform clusterings of uniformly sampled cells, while data-dependent methods k-means++ and SRS provide competitive results on only a few instances. In Figure 3.3 we show that the more even sampling of the transcriptional landscape by our spherical sketch facilitates the detection of biological cell types. Across datasets and sampling sizes, the clustering of our spherical sketches achieves better or comparable separation of cell types than the clustering of the corresponding geometric sketch. In only 3 out of 36 instances, geometric sketching yielded slightly better median ARI scores. Remarkably, in several cases the clustering of sketches better agrees with the true biological cell types than the clustering based on the full data. This observation is consistent with the assumption of a more balanced composition of cell types in a sketch, but an artifact of the clustering algorithm cannot be excluded, especially in light of the *impossibility theorem for clustering* (Kleinberg, 2003). Note that despite a small variability in Hausdorff distance, the non-deterministic behavior of the Louvain algorithm contributes to the different ARI scores observed in the repeated clustering of spherical sketches.

3.2.3 Impact of distance metrics

Downstream analysis of scRNA-seq such as clustering and trajectory inference relies on a metric that measures the distance between cells in gene expression space. Distance metrics

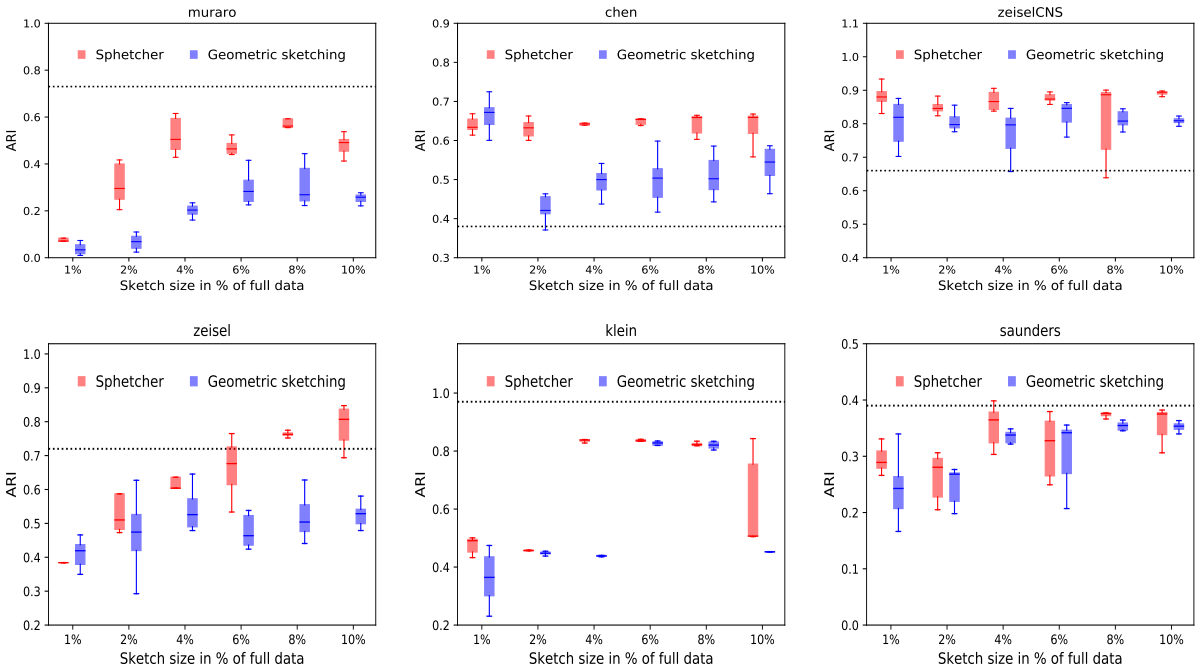

Figure 3.3: Comparison of sketch based clustering accuracy. Louvain clustering of spherical sketches computed by Sphetcher yields more accurate cell clusterings as measured by Adjusted Rand Index (ARI) than geometric sketching based clustering. In both cases, labels assigned to cells in the sketch are propagated to the remaining cells using k-nearest neighbor classification. The dotted line indicates the ARI score achieved by clustering the full data using the same Louvain algorithm.

such as Euclidean distance, correlation-based distance, and cosine similarity (adapted as distance) have been proposed as adequate measures of dissimilarity, and its specific choice might depend on assumptions made by computational analysis methods, properties of datasets, and the specific task at hand (Kim et al., 2018; Jaskowiak et al., 2014). While the Hausdorff distance is defined based on a given metric, geometric sketching ignores the metric space and considers absolute differences in each dimension independently.

Here, we illustrate the flexibility of Sphetcher in optimizing the Hausdorff distance under different distance metrics and demonstrate that the choice of metric can impact downstream clustering analysis of scRNA-seq data. To this end, we sample a subset of cells from a medium size dataset with complex population structure (*chen*) using Sphetcher with four different metrics: Euclidean, Manhattan, cosine, and Pearson correlation distance. We cluster the four resulting sketches using the same approach as in Section 3.2.2, and compare the quality of the clusterings to the one obtained from a geometric sketch. Note that the geometric sketching approach proposed in Hie et al. (2019b) cannot distinguish different distance metrics. Figure 3.4 shows that spherical sketches computed by Sphetcher using Euclidean distance as metric in the objective function yield most accurate clusterings of this dataset. While cosine and Pearson distances have a slightly negative effect on the quality of the clustering, Manhattan distance and geometric sketching yield substantially less accurate clusterings, especially for small sketch sizes.

On dataset *muraro*, geometric sketching again achieves overall lower ARI scores than Sphetcher using different metrics (Figure 3.4). In contrast to dataset *chen*, however, Eu-

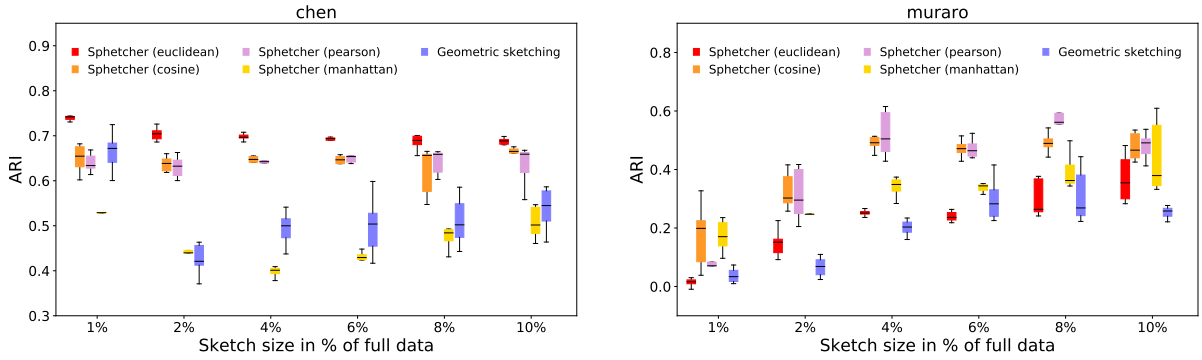

Figure 3.4: Impact of distance metrics on clustering performance. While clustering based on spherical sketches computed by Sphetcher using Euclidean distance yields most accurate results on dataset chen, alternative metrics used by Sphetcher lead to higher ARI scores on dataset muraro, illustrating the importance of Sphetcher's flexible optimization scheme. In contrast, geometric sketching does not distinguish different distance metrics and yields overall less accurate clusterings.

clidean distance based sampling does not show any improvement over alternative metrics, illustrating the benefit of Sphetcher's unique ability to take into account different metrics suitable for different tasks.

3.2.4 Sphetcher detects rare population of inflammatory macrophages

Hie et al. (2019b) report and experimentally validated the discovery of a rare population of inflammatory macrophages by clustering a geometric sketch of 20,000 cells sampled from a dataset of 254,941 umbilical cord blood cells. In contrast, clustering the full dataset or a uniform subsample did not reveal this rare population of cells, presumably due to their limited visibility among the more abundant inactive macrophages. We repeated the experiment by clustering our spherical sketch of same size (20,000 cells) obtained after prior grid sampling (Sphetcher-H) using the Louvain community detection algorithm. As expected, we were also able to discover a similar cluster of inflammatory macrophages based on the same set of marker genes CD74, HLA-DRA, B2M, and JUNB (AUROC$>$ 0.88).

3.2.5 Fairness incorporates time points in trajectory reconstruction

In time series studies of gene expression, single cells are typically collected at different (known) time points. In this section, we illustrate how fairness aspects can be used to incorporate this additional information into the construction of a spherical sketch. To compare the gene expression dynamics of human skeletal muscle myoblast (HSMM) differentiation to the reprogramming of fibroblasts to myotubes, in Cacchiarelli et al. (2018), single cells were sampled every 24 hours post induction of myoblast differentiation, between 0 and 72 hours. Consistent with the original publication, we reconstruct the single-cell trajectory of HSMM differentiation using Monocle 2 (Qiu et al., 2017), ignoring the information on the collection time point of cells. Figure 3.5 (left) shows the resulting trajectory, in which cells are initially in a cycling state and either fully progress to contractile myotubes or fail to differentiate. Cells are colored by the 4 different time points. For marked cells (black circle) the inferred pseudotime, i.e. their level of progression through differentiation, and the actual

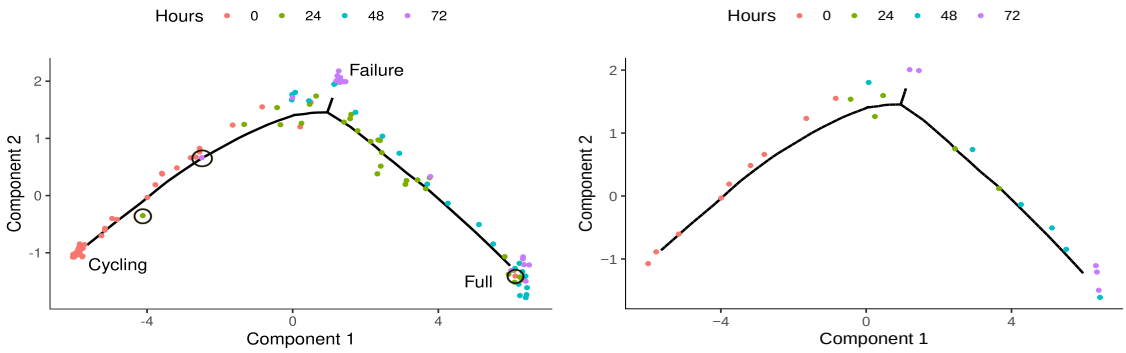

Figure 3.5: Single cell trajectories of HSMM differentiation. Single-cell trajectories of HSMM differentiation as reconstructed by Monocle 2 from the full data (left) and from Sphetcher's spherical sketch with fairness constraints (right) consistently describe progression through differentiation. Cells for which inferred pseudotime and collection time point disagree are marked with a black circle and were automatically removed as 'outlier' cells by Sphetcher. See also Supplemental Figures S19-S21.

time they were collected, disagree. Even though cells do not always progress through the process of differentiation in a synchronous manner, the presence of fully differentiated cells at time point 0, for example, is most likely an artifact caused by noise in the single cell measurements.

We sought to automatically detect and remove cells for which the collection time point disagrees with their transcriptomic state through a constrained sketching approach. Instead of imposing a hard constraint that removes "outlier" cells, we let our sketching algorithm decide if cells at different time points are necessary to evenly represent the global transcriptional space. Since our fairness-inspired model imposes covering constraints that require a certain number of cells to be sampled from each time point, a fair sampling of cells will implicitly discourage the selection of outlier cells that lie close to cells in a similar state but which have been collected at different time points.

We compare the trajectories computed by Monocle 2 from the geometric sketch, our (unconstrained) spherical sketch, and our fairness-inspired spherical sketch that picks at least four cells from each time point. On all sketches, the overall structure of the inferred trajectory agrees with the trajectory computed from the full data (Figure 3.5 (right) and Supplemental Figures S19-S21). However, while outlier cells are included in both the geometric sketch (8 out of 8 trials, Supplemental Figure S20) and the unconstrained spherical sketch (2 out of 8 trials, Supplemental Figure S21), Sphetcher under fairness constraints decides to not use outlier cells to represent the transcriptional space. Fairness encourages Sphetcher, for example, to not include fully differentiated cells from time point 0 into the sketch (Figure 3.5 (right) and Supplemental Figure S19). Even more, while constrained Sphetcher includes at least one cell collected at time point 72 in the final state (Full) in Figure 3.5 and in all trials in Supplemental Figure S19, unconstrained sketches do not retain any such cell in any but a single trial (Supplemental Figures S20 and S21).

In addition, we construct gene expression kinetics plots using Monocle 2 for a set of genes assessed in Cacchiarelli et al. (2018). The expression dynamics inferred from our fair spherical sketch appear smoother than those obtained from the full data, and cells in our

Table 3.1: Comparison of CPU time (in seconds) of geometric sketching and Sphetcher-H. Running times are reported separately for the prior grid sampling, the calculation of pairwise distances, and the computation of a covering of all cells with spheres using a greedy set cover approach.

Dataset	# cells	Sphetcher-H			Geometric sketching
		Grid	Distances	Set Cover	
Cord blood	254,941	1.0	43.0	88.0	23.0
ZeiselCNS	464,713	3.0	153.0	116.0	120.0
Saunders	665,385	5.0	318.0	200.0	201.0
Cao	2,026,641	10.0	600.0	400.0	1869.0

sketch better fit the interpolated expression (Supplemental Figure S22).

3.2.6 Scalability

Here, we demonstrate scalability of our hybrid strategy Sphetcher-H that combines grid sampling with subsequent spherical sketching to large single-cell datasets. In Table 3.1 we compare the running time of Sphetcher-H to the construction of a geometric sketch (Hie et al., 2019b) on the zeiselCNS, saunders, and umbilical cord blood datasets used in previous benchmarks as well as on a dataset (*cao*) comprising 2 million cells (Cao et al., 2019). On the latter dataset, geometric sketching and Sphetcher-H require in total around 30 minutes and 16 minutes of computation, respectively. Remarkably, our naïve grid sampling strategy alone is orders of magnitude faster than geometric sketching but achieves competitive Hausdorff distances on the zeiselCNS and saunders datasets (Supplemental Figure S18).

3.2.7 Data and Software Availability

Sphetcher is available at `https://github.com/canzarlab/Sphetcher`, where we also make spherical sketches of public, large scRNA-seq dataset available for download.

3.3 Conclusions

We have introduced Sphetcher, a novel method that computes a small sketch of single-cell datasets that accurately summarizes its transcriptional heterogeneity. Sphetcher utilizes the thresholding technique to efficiently pick representative cells within spheres that better approximate the global geometry than boxes. Furthermore, we provide theoretical justification for its robust performance in practice. Sphetcher is able to accelerate scRNA-seq analyses such as the detection of cell types through clustering or the reconstruction of developmental trajectories. At the same time, it has the ability to shift the focus from a "more data, less algorithm" regime to a "less (but accurate) data, more algorithm" approach. In addition, Sphetcher is sensitive to rare cell types, is flexible in its use of different distance metrics, and allows to use prior categorical information on, e.g., biological cell types or collection time point to guide the selection of cells into a representative sketch.

Chapter 4

Dynamic pseudo-time warping of complex trajectories

Single cell RNA sequencing enables the reconstruction of cellular lineages underlying biological processes such as cell development and differentiation. scRNA-seq can take a snapshot of cells at a time and has enabled the ordering of single cells using a number of trajectory inference methods. The comparison of single cell trajectories between two processes can illuminate the differences and similarities between the two and thus be a powerful tool. Current methods for the comparison of trajectories rely on the concept of dynamic time warping (dtw), which was used for the comparison of two time series. Consequently, these methods are restricted to simple, linear trajectories. Here we adopt a concept of arboreal matchings (Böcker et al., 2013) and propose an algorithm to compare and align complex trajectories that more realistically contain branching points that divert cells into different fates. Moreover, we provided theoretical link between dtw and arboreal matchings via our lower bound and upper bound theorems. We implement a suite of exact and heuristic algorithms in our tool Trajan. Trajan workflow is given in Figure 4.1. Trajan automatically pairs similar biological processes between conditions and aligns them in a globally consistent manner. In an alignment of single cell trajectories describing human muscle differentiation and myogenic reprogramming, Trajan identifies and aligns the core paths without prior information. From Trajan's alignment, we are able to reproduce recently reported barriers to reprogramming. In a perturbation experiment, we demonstrate the benefits in terms of robustness and accuracy of our model which compares entire trajectories at once, as opposed to a pairwise application of dtw. This chapter is adapted from: Van Hoan Do, Mislav Blažević, Pablo Monteagudo, Luka Borozan, Khaled Elbassioni, Soeren Laue, Francisca Rojas Ringeling, Domagoj Matijevic and Stefan Canzar. *Dynamic pseudo-time warping of complex single-cell trajectories*. bioXiv, 2019. The results were presented at RECOMB 2019.

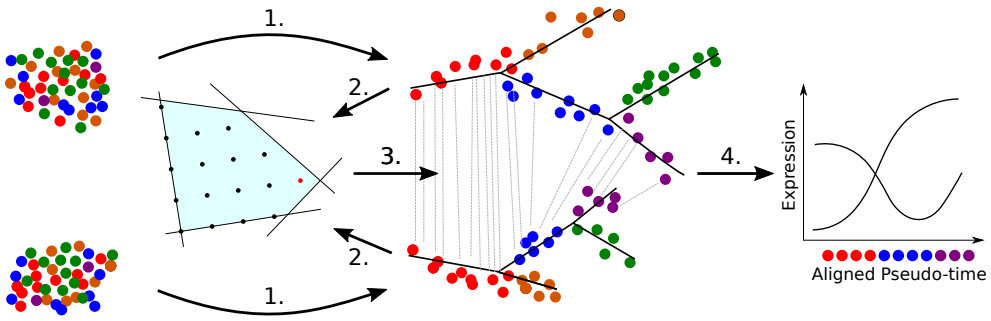

Figure 4.1: Trajan workflow. 1. Complex trajectories are reconstructed from single-cell RNA measurements using, e.g., Monocle 2. After smoothing and scaling (2.), Trajan aligns entire trajectories by computing an arboreal matching using a branch-and-cut approach (3.), which transforms (warps) the individual pseudo-time scales into a shared one along which expression kinetics can be compared (4.). For simplicity, only the alignment between one pair of paths is shown.

4.1 Methods

Dynamic time warping is the algorithmic workhorse underlying current methods that compare linear single-cell trajectories. In the next section we briefly review the concept of dynamic time warping and show that an attempt to generalize dtw to complex trajectories naturally leads to arboreal matchings between trees, which were introduced previously in the context of phylogenetic trees (Böcker et al., 2013).

4.1.1 DTW versus arboreal matching

As in classical sequence alignment, dtw matches similar elements in two sequences while preserving their order. To account for different speeds at which the two sequences advance, however, each element of one sequence can be mapped to multiple elements in the other sequence (Figure 4.2 left). More formally, given two time series $(x_i)_{i=1}^n, (y_j)_{j=1}^m$, and a distance or similarity measure $d(x_i, y_j) \geq 0$ between the time points x_i and y_j, a *warping* is a sequence $p = (p_1, \ldots, p_L)$ with $p_\ell = (n_\ell, m_\ell) \in [1:n] \times [1:m]$ for $\ell \in [1:L]$ that satisfies the following three conditions: (i) *Boundary*: $p_1 = (1,1)$ and $p_L = (n,m)$. (ii) *Monotonicity*: $n_1 \leq n_2 \leq \cdots \leq n_L$ and $m_1 \leq m_2 \leq \cdots \leq m_L$. (iii) *Step size*: $p_{\ell+1} - p_\ell \in \{(1,0),(0,1),(1,1)\}$ for $\ell \in [1:L-1]$. Note that the example warping in Figure 4.2 (left) contains no pair of crossing edges and thus preservers the order of the two sequences.

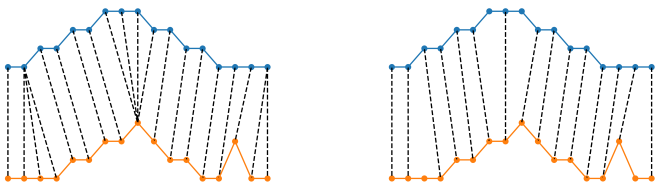

Figure 4.2: An example of a warping (left) and an arboreal matching (right) between two time series.

The classic dtw aims to find a warping p minimizing the total distance between mapped elements:

$$c_p(x, y) := \sum_{\ell=1}^{L} d(x_{n_\ell}, y_{m_\ell}).$$

The optimal warping can be computed by a dynamic program that solves:

$$D(i, j) = d(x_i, y_j) + \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\}. \tag{4.1}$$

There are various extensions of the classic dtw described above that can be mainly classified as 1) restricting the range of the mapping to a certain window; 2) assigning different weights to different types of steps; and 3) using different step patterns, e.g. $p_{\ell+1} - p_\ell \in \{(1,1),(1,2),(2,1)\}$ in (iii). In the following, we consider the widely used classic dtw which is also the default scheme for computing dtw (Zhao and Itti, 2018). Since state-of-the-art methods like Monocle 2 (Qiu et al., 2017) and DPT (Haghverdi et al., 2016) aim to construct smooth trajectories, the classic dtw provides the necessary flexibility for most single-cell alignment tasks.

Here, we propose a generalization of classic dtw from paths, i.e., linear trajectories, to trees, i.e., complex trajectories: We want to align each path in tree T_1 to at most one path in T_2 and vice versa and, similar to dtw, preserve the order of nodes along the paths, i.e., no crossing edges. In addition, we require all alignments to be consistent, that is, every node must be matched to the same node in all pairwise alignments it is part of. Böcker et al. (2013) introduced *arboreal matchings* that formalize such a consistent path-by-path alignment of trees: An arboreal matching is a matching M, i.e., one-to-one correspondence between nodes in trees T_1 and T_2 such that for any $(u_1, v_1), (u_2, v_2) \in M$, u_2 is a descendant of u_1 iff v_2 is a descendant of v_1.

In contrast to dtw, an arboreal matching M matches each node (cell) to at most one similar node (cell) in the other tree (trajectory). It is not required to cover all nodes between each pair of paths, but we can flexibly penalize nodes that remain unmatched by M in the objective function:

$$c(M) := \sum_{(u,v)\in M} d(u, v) + \sum_{\substack{u\in V_1 \\ u \text{ unmatched}}} d(u, -) + \sum_{\substack{v\in V_2 \\ v \text{ unmatched}}} d(-, v), \tag{4.2}$$

where the cost of leaving node u (v) unmatched is $d(u,-) > 0$ ($d(-,v) > 0$). In fact, the arboreal matching of minimum cost (4.2) between two paths $P = (x_1, \ldots, x_n)$ and $Q = (y_1, \ldots, y_m)$ can be solved by a very similar dynamic program as in dtw (4.1):

$$D(i, j) = \min\{D(i-1, j-1) + d(x_i, y_j), D(i-1, j) + d(i, -), D(i, j-1) + d(-, j)\}. \tag{4.3}$$

An example arboreal matching between two paths is shown in Figure 4.2 (right). Again, the non-crossing edges align the two time-series to reveal similarities and unmatched nodes indicate compressed or stretched sections. This makes arboreal matchings as flexible as dtw in the comparison of two trajectories. More specifically, we will show that by choosing an appropriate penalty for unmatched vertices, the optimal dtw and the optimal arboreal matching yield similar measures of similarity or distance of the compared trajectories. Denote by d_{dtw} and d_M the optimal value of the classic dtw and the arboreal matching between two paths P and Q, respectively. The following theorem provides an upper bound on d_{dtw}.

Theorem 4. *Let* $D = \max_{i,j} d(x_i, y_j)$. *If* $d(x,-) = \max_{y \in Q} d(x,y)$ *and* $d(-,y) = \max_{x \in P} d(x,y)$, *then*

$$d_{dtw} \leq d_M \leq d_{dtw} + kD,$$

where k *is the minimum number of edges that need to be removed to transform the optimal warping to an arboreal matching.*

Proof of Theorem 4. The first inequality is proven by induction on $i + j$.

Let p^* be the optimal warping and k the minimum number of edges that need to be removed to transform p^* to a feasible arboreal matching M. Since M has k unmatched vertices, we have

$$c(M) \leq c_{p^*}(x,y) + kD.$$

This implies that $d_M \leq d_{dtw} + kD$, which also completes the proof of the theorem. □

Next, we develop a lower bound theorem for the classic dtw. An edge (x,y) in the warping p is called *redundant* if both vertices x and y are covered by at least two edges in p.

Lemma 1. *There exists an optimal warping of the classic dtw without redundant edges.*

Proof of Lemma 1. Conversely, let p^* be an optimal warping such that $(x_i, y_j) \in p^*$ and both x_i and y_j are covered by at least two edges in p^*. From the coverage property of the warping we must have $(x_i, y_{j-1}) \in p^*$ or $(x_i, y_{j+1}) \in p^*$. If $(x_i, y_{j-1}) \in p^*$, we get $(x_{i+1}, y_j) \in p^*$ since y_j is covered by at least two edges in p^* and by the warping conditions. As a results, $D(i,j) = D(i,j-1) + d(x_i, y_j)$ and $D(i+1,j) = D(i,j) + d(x_{i+1}, y_j) = D(i,j-1) + d(x_i, y_j) + d(x_{i+1}, y_j)$. From (4.1), we obtain $D(i+1,j) \leq D(i,j-1) + d(x_{i+1}, y_j)$. This implies that $d(x_i, y_j) \leq 0$. Since $d(x_i, y_i) \geq 0$ we must have $d(x_i, y_j) = 0$. As a consequence, we can remove (x_i, y_j) from p^* without violating the warping conditions. The case $(x_i, y_{j+1}) \in p^*$ is proven in an analogous manner. □

Given an optimal warping p^*, we assign penalties to unmatched vertices such that $d_M \leq d_{dtw}$. Let p^* be an optimal warping without redundant edges, define

$$L_1(p^*) := \{x \in P | x \text{ is covered by at least two edges from } p^*\},$$
$$L_2(p^*) := \{y \in Q | y \text{ is covered by at least two edges from } p^*\}.$$

Then, we impose penalties

$$d(-,y) = \begin{cases} d(x,y) & \text{if } \exists x \in L_1(p^*) \text{ and } (x,y) \in p^*, \\ \max\limits_{x \in L_1(p^*)} d(x,y) & \text{otherwise.} \end{cases} \tag{4.4}$$

We define penalties $d(x,-), x \in P$, analogously. Since p^* has no redundant edges, $d(-,y)$ is uniquely defined. Conversely, if there exist $x_1, x_2 \in L_1(p^*)$ such that $(x_1, y) \in p^*, (x_2, y) \in p^*$, the non-redundancy of p^* is violated. We have the following lower bound theorem.

Theorem 5. *If* $d(x,-), d(-,y)$ *are defined as in* (4.4), *then* $d_M \leq d_{dtw}$.

Proof of Theorem 5. Let p^* be a non-redundant optimal warping. For every vertex $x \in L_1(p^*)$ and $y \in L_2(p^*)$ we delete all incident edges but one, which results in an arboreal matching of the same cost as dtw p^*. Hence, it implies that $d_M \leq d_{dtw}$. □

In Section 4.2.1 we illustrate how closely the optimal arboreal matchings based on lower and upper bound penalty scheme follow the optimal dtw path.

4.1.2 Limitations of the naïve ILP formulation

Finding the matching minimizing (4.2) can be phrased as a maximum matching problem that explicitly forbids the two possible types of ancestry violations: Two edges can be crossing, or two nodes on the same root-to-leaf path are matched to nodes on different root-to-leaf paths (Figure 4.3). The former constraint is equally imposed by dtw, the latter is a consequence of the simultaneous comparison of multiple paths and prevents arbitrary jumps between biological processes in the comparison. In the proof-of-concept study by Böcker et al.

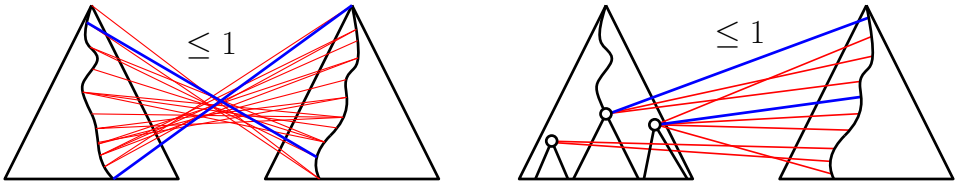

Figure 4.3: Pair of crossing edges (blue) extended to a clique of crossing edges (left) and pair of semi-independent edges (blue) extended to a clique of semi-independent edges (right).

(2013), they described feasible arboreal matchings between two rooted trees $T_1 = (V_1, E_1)$, $T_2 = (V_2, E_2)$, by the following simple ILP:

$$\max \sum_{i=1}^{|V_1|} \sum_{j=1}^{|V_2|} w(i,j) x_{i,j} \tag{P}$$

$$\text{s.t.} \sum_{j=1}^{|V_2|} x_{i,j} \leq 1 \qquad \forall i = 1 \ldots |V_1|, \tag{4.5}$$

$$\sum_{i=1}^{|V_1|} x_{i,j} \leq 1 \qquad \forall j = 1 \ldots |V_2|, \tag{4.6}$$

$$x_{i,j} + x_{k,l} \leq 1 \qquad \forall \{(i,j),(k,l)\} \in \mathcal{I}, \tag{4.7}$$

$$x_{i,j} \in \{0,1\}, \tag{4.8}$$

where indicator variables $x_{i,j}$ denote the presence or absence of an edge (i,j), weights $w(i,j) := d(i,-) + d(-,j) - d(i,j)$. Pairs of edges (i,j) and (k,l) are *compatible* if it holds that k is a descendant of i in T_1 iff l is a descendant of j in T_2. Set $\mathcal{I}$ contains pairs of edges $\{(i,j),(k,l)\}$ that are incompatible, i.e., they are either crossing or one-sided independent (Figure 4.3).

As our experiments in Section 4.2.3 show, this ILP formulation does not allow to practically align trajectories comprising as few as 100 single cells. In the following theorem, we identify its weak LP-relaxation as a theoretical explanation for this empirical performance, since the search space that needs to be explicitly explored by an ILP solver depends on the strength of the LP relaxation. Let OPT denote an optimal solution to the above ILP and let $w(\text{OPT})$ be its optimal score. Let $|V_1| = n, |V_2| = m$, and w.l.o.g we assume $n \leq m$.

Theorem 6. *The integrality gap of the linear programming relaxation of* (P) *is* $n - o(1)$.

Proof of Theorem 6. Let $K = \max_{i,j} w(i,j)$, hence K is bounded above by $w(\text{OPT})$. Moreover, for any feasible solution x to the relaxation, we have

$$\sum_{i=1}^{|V_1|} \sum_{j=1}^{|V_2|} w(i,j) x_{i,j} \leq \sum_{i=1}^{|V_1|} \sum_{j=1}^{|V_2|} K x_{i,j} \leq \sum_{i=1}^{|V_1|} K \Big(\sum_{j=1}^{|V_2|} x_{i,j}\Big) \leq \sum_{i=1}^{|V_1|} K = Kn.$$

Therefore, the optimal value of the LP relaxation is at most n times $w(\mathrm{OPT})$. Our bad instance consists of the two rooted trees shown in Figure 4.4, with $w(\text{red/blue edges}) = 1$ and $w(\cdot) = 0$ otherwise. Any pair of nonzero weight edges are incompatible, so the maximum cost matching is 1. Let $x(\text{red edges}) = \frac{1}{n-1}$, $x_{n,1} = 1 - \frac{1}{n-1}$, $x_{n,m} = \frac{1}{n-1}\mathbb{1}_{n\neq m}$, and $x(\cdot) = 0$ otherwise, where $\mathbb{1}_{n\neq m}$ is a binary number such that $\mathbb{1}_{n\neq m} = 1$ iff $n \neq m$. Hence, x is a feasible solution with cost of $(n-1)^2/(n-1) + 1 = n$ if $n \neq m$ and $n - \frac{1}{n-1}$ if $n = m$. Therefore, the optimal value of the LP relaxation at least $n - o(1)$. □

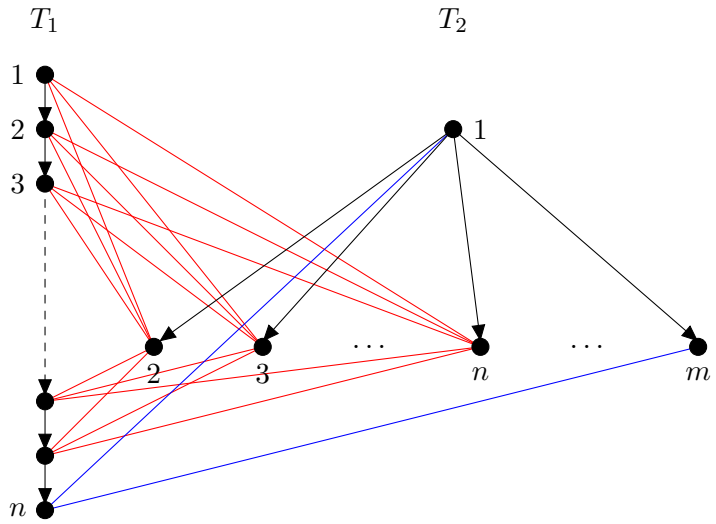

Figure 4.4: The integrality gap of the LP relaxation of (P) is n.

Do et al. (2019) provided an alternative formulation of the naïve ILP formulation and solved it using a branch and cut algorithm. Its main ingredients are (i) cuts that trim the LP relaxation closer to the convex hull of feasible arboreal matchings, (ii) polynomial-time algorithms that can find these cuts on demand, (iii) a branch-and- bound scheme that makes use of modern CPU architectures, and (iv) an in-house developed, non-commercial, non-linear solver that we use for all continuous optimization problems. For details we refer the reader to the paper (Do et al., 2019).

4.2 Results

We have implemented the branch-and-cut algorithm described above and have bundled it with our non-linear solver (Do et al., 2019) in our novel trajectory alignment tool Trajan. Trajan adopts a strategy similar to Cacchiarelli et al. (2018) to prepare the output of Monocle 2 (or similar trajectory reconstruction methods) for a meaningful alignment, including the smoothing and scaling of expression curves.

4.2.1 Lower and upper bounds on dtw

Here, we illustrate the practical relevance of the upper bound (UB, Theorem 4) and lower bound (LB, Theorem 5) that the optimal arboreal matching between two paths can provide on the optimal dtw. We align two simple trajectories constructed from scRNA-seq data on

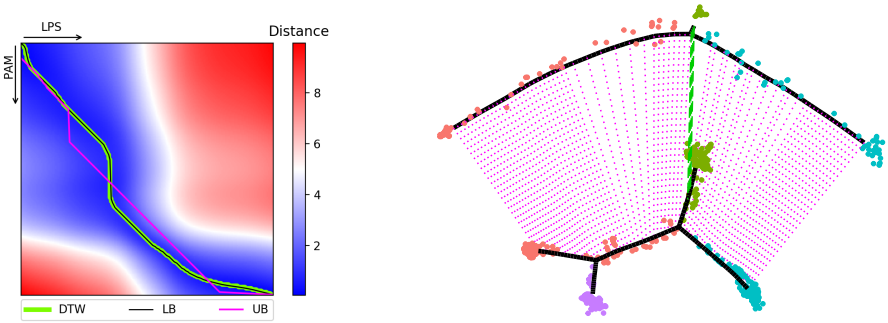

Figure 4.5: The optimal dtw path and two optimal paths computed by Trajan with lower bound (LB) and upper bound (UB) penalty scheme. The optimal dtw path and Trajan's LB path coincide (left). Alignment of myogenic reprogramming and differentiation dynamics (right). Trajan discovers the core branches of similar cell fates.

dendritic cells stimulated under two conditions (LPS and PAM) collected at 4 time points after stimulation (Shalek et al., 2014). The two linear trajectories and the dissimilarity matrix were obtained from a recent study that introduced cellAlign (Alpert et al., 2018), a method that aligns two simple trajectories based on dtw. The optimal solutions computed by cellAlign using dtw and by Trajan using the LB penalty scheme are equivalent (Figure 4.5). When using the UB penalty scheme, Trajan's optimal path through the dissimilarity matrix roughly follows the dtw path and represents a solution with almost 2 times larger score.

4.2.2 Trajan reproduces barriers in myogenic reprogramming

Here, we re-analyzed two public single-cell datasets: human skeletal muscle myoblast (HSMM) differentiation and human fibroblasts undergoing MYOD-mediated myogenic reprogramming (hFib-MyoD). These datasets were previously analyzed in Cacchiarelli et al. (2018), where the authors set out to compare these related processes in order to identify molecular barriers that hinder the efficient reprogramming of fibroblasts to myotubes. The authors used known myoblast differentiation markers (CDK1, ENO3, MYOG) to identify the core path within the complex trajectory constructed from hFib-MyoD, and they aligned this path to the core path in normal muscle development (HSMM) using dtw. The authors pointed out that the combined trajectory constructed from cells in both conditions did not intermix cells and thus did not allow to assess critical commonalities and differences in expression dynamics. We repeated the single-cell data analysis described in Cacchiarelli et al. (2018) to obtain the corresponding trajectories from Monocle 2 (Figure 4.6).

We then sought to align these complex trajectories using our algorithm. We show that Trajan is able to align the core paths of each complex trajectory, without any previous knowledge or path picking, using the same distance measure (correlation) as in the original publication. The global dynamics alignment of HSMM and hFib-MyoD are shown in Figure 4.5 (right). Interestingly, our approach not only aligns the core trajectories, but it also aligns the branches corresponding to failure of reprogramming, which are characterized in both processes by cells that exited the cell cycle, yet failed to proceed toward differentiation (Cacchiarelli et al., 2018; Qiu et al., 2017).

After performing the trajectory alignment with Trajan, we constructed gene expression

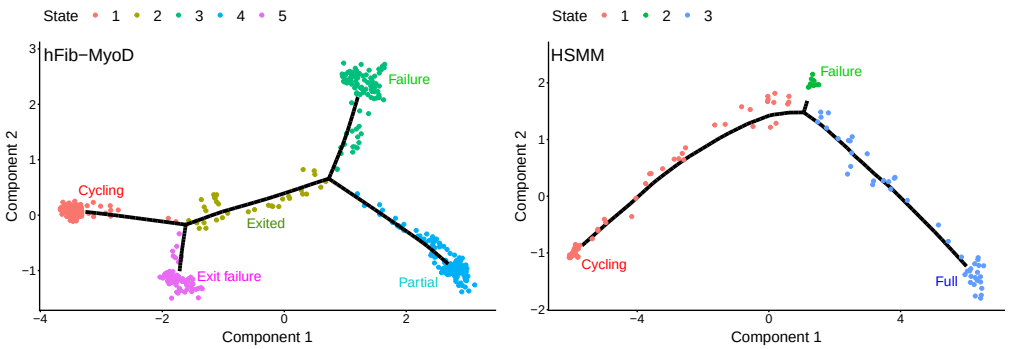

Figure 4.6: Trajectories of myogenic reprogramming (left) and differentiation (right). Cycling: undifferentiated, actively proliferating cells; Exited: cells lacking expression of cell cycle and muscle contraction genes; Exit failure: cells expressing genes of early myoblast differentiation yet still proliferating; Failure: cells lacking expression of cell cycle genes as well as of muscle contraction genes; Partial: cells expressing MYOG and multiple muscle contraction genes and lacking expression of cell cycle genes; Full: full progression to contractile myotubes.

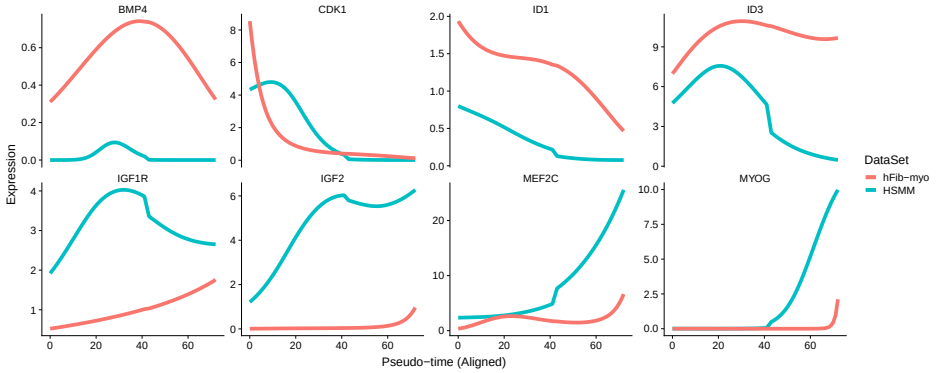

Figure 4.7: Gene expression dynamics after trajectory alignment with Trajan.

kinetics plots for a set of genes that were assessed in Cacchiarelli et al. (2018) to investigate whether our alignment was able to reproduce their reported findings regarding similarities and differences between these two processes. Indeed, we were able to reproduce their key findings: Proliferation marker CDK1 is downregulated both in HSMM and hFib-MyoD; Muscle transcriptional regulators (MEF2C, MYOG) are upregulated later and to a lesser extent in hFib-MyoD compared to HSMM; BMP4 is only expressed in hFib-MyoD and ID family proteins (ID1, ID3) which lie downstream of BMP signaling fail to be downregulated in hFib-MyoD; IGF pathway genes (IGF2, IGF1R) are expressed at higher levels in HSMM (Figure 4.7).

We evaluated Trajan using penalty schemes that assign the maximum and average weight of incident edges as well as the minimum cost implied by the lower bound Theorem 5 over all pairs of paths (*lb*). While the maximum scheme (*max*) is a direct generalization of the cost scheme applied by Theorem 4, the averaging scheme (*avg*) tries to capture the expected cost of leaving a vertex unmatched and is the default scheme applied by Trajan. All schemes correctly picked the correct core paths in the two trajectories and are robust under subsampling (Figure 4.8).

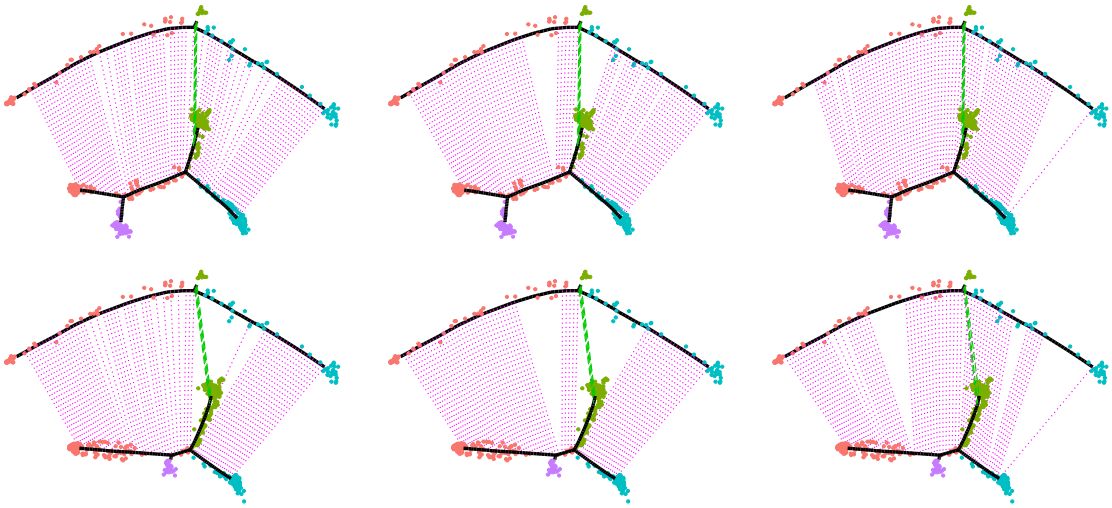

Figure 4.8: Alignment of myogenic reprogramming and differentiation dynamics using three different penalty schemes, from left to right: avg, max, lb. The bottom row shows results for random subsamples of input cells and ordering genes.

4.2.3 Accuracy of Trajan

Here, we compare the accuracy of Trajan in matching "correct" cells between complex trajectories to the path-wise alignment by dtw. To this end, we perturb the hFib-MyoD trajectory output by Monocle 2 by randomly subsampling 80% of the input cells and 80% of the genes used for ordering them along pseudo-time. We align isomorphic trajectories (trees) comprising a variable number of nodes (parameter *ncenter* in Monocle 2), measuring the difference between nodes by Euclidean distance. Since we know the true correspondence of nodes between different perturbed trees, we can count false positive and false negative alignments as a measure of accuracy. In Table 4.1 we report the number of false positive (FP) and false negative (FN) alignments of the classic dtw run on each true pair of paths, and Trajan using different penalty schemes (avg, max, lb). Trajan takes the entire trees as input, it is not given the correct path-to-path correspondence. Nevertheless, Trajan almost always finds the true correspondence between cells, compared to the path-wise dtw scheme, that introduced both FP and FN alignments.

Table 4.1: Average number of false positive ($^+$) and false negative($^-$) alignments of Trajan and path-wise dtw. The average is taken over a variable number of instances comprising a total # of nodes in the two input trees.

# of nodes	# of instances	Trajan						DTW	
		avg$^+$	avg$^-$	max$^+$	max$^-$	lb$^+$	lb$^-$	FP	FN
80	435	0.0	0.2	0.0	0.0	12.8	17.7	35.0	32.1
100	435	0.0	0.0	0.0	0.1	12.9	17.6	54.6	50.5
140	190	0.0	0.0	0.0	0.0	18.6	30.0	54.5	50.7
180	190	0.0	0.0	0.0	0.2	31.8	38.9	83.9	76.0
210	45	0.0	0.0	0.0	0.0	33.9	43.9	76.6	70.3

Table 4.2 reports the running times of the naïve ILP using the commercial solver IBM ILOG CPLEX 12.7 and Trajan coupled with our in-house non-linear solver on a random subset of the instances introduced above. On a 2.30GHz Linux system using up to 15 threads, Trajan is at least 13 times faster than the naïve ILP using CPLEX. CPLEX was not able to solve instances with more than 200 nodes since it exceeded the memory limit of 320 GB.

Table 4.2: Average runtime in seconds of Trajan vs CPLEX

# of nodes	# of instances	Trajan			CPLEX		
		avg	max	dtw	avg	max	lb
80	435	3.0	3.2	1.0	41.6	41.3	32.8
140	190	23.2	26.3	6.6	405.8	416.5	185.8
180	45	69.6	73.0	23.7	1381.8	1585.2	1041.0
210	45	120.9	147.4	47.6	-	-	-

4.3 Conclusions

We have introduced Trajan, a novel method that allows for the first time the alignment of complex (non-linear) single-cell trajectories. Originally introduced to compare phylogenetic trees, in Trajan we adopt arboreal matchings to perform an unbiased alignment enabling the meaningful comparison of gene expression dynamics along a common pseudo-time scale. Trajan does not make any assumptions concerning the algorithm used to reconstruct the trajectory and can in principle be coupled with any available reconstruction method. In a future algorithm, an arboreal matching between cells might prove useful in guiding a joint learning of trajectories for two biological processes.

Chapter 5

Visualization of single-cell multimodal omics

In this chapter we generalize t-SNE and UMAP to the joint visualization of multimodal single-cell measurements. While t-SNE and UMAP seek a low-dimensional embedding of cells that preserves similarities in the original (e.g. gene expression) space as well as possible, we propose j-SNE and j-UMAP that simultaneously preserve similarities across all modalities (Figure 5.1). Through Python package JVis they will allow to combine different views of the data into a unified embedding that can help to uncover previously hidden relationships among them. Our approach automatically learns the relative contribution of each modality to a concise representation of cellular identity that promotes discriminative features but suppresses noise. On eight real datasets, j-SNE and j-UMAP produce unified embeddings that better agree with known cell types and that harmonize RNA and protein velocity landscapes. This chapter is based on the following publication: Van Hoan Do and Stefan Canzar. *A generalization of t-SNE and UMAP to single-cell multimodal omics.* Genome Biology, 22(1):130, 2021.

5.1 Methods

5.1.1 Overview of method

In j-SNE we want to learn a joint embedding $\mathcal{E}$ of cells for each of which we have measured multiple modalities. Analog to t-SNE (van der Maaten and Hinton, 2008), we want to arrange cells in low-dimensional space such that similarities observed between points in high-dimensional space are preserved, but in all modalities at the same time. Generalizing the objective of t-SNE, we aim to minimize the convex combination of KL divergences of similarities in the original high-dimensional (distribution P) and similarities in the embedding low-dimensional space (distribution Q) for each modality k:

$$C(\mathcal{E}) = \sum_k \alpha_k KL(P^{(k)}||Q) + \lambda \sum_k \alpha_k \log \alpha_k, \tag{5.1}$$

where coefficients α of the convex combination represent the importance of individual modalities towards the final location of points in the embedding. We add a regularization term (with regularization parameter λ) that prevents the joint embedding from being biased to-

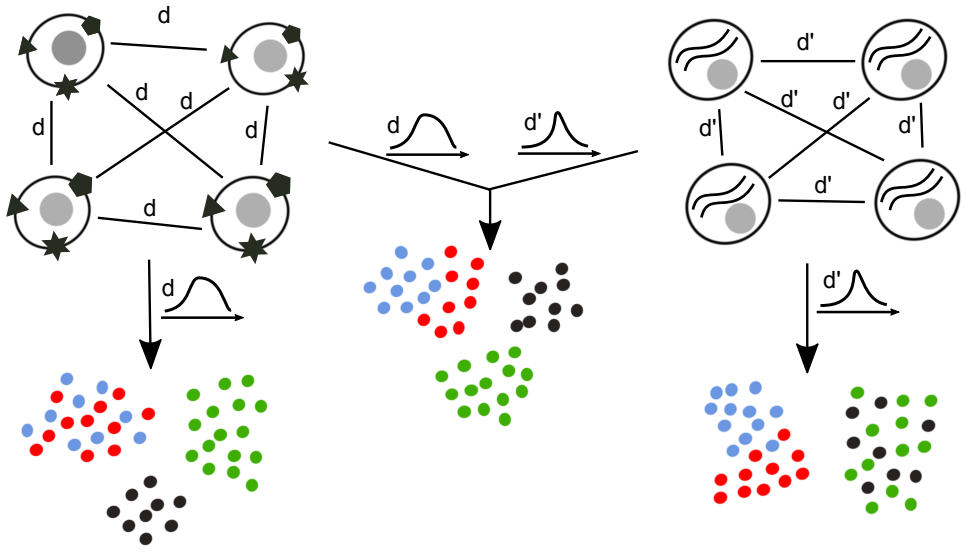

Figure 5.1: Overview of the joint embedding in JVis. Metrics d (left) and d' (right) measure the dissimilarity of different cellular phenotypes of individual cells, such as the expression of surface proteins (left) and mRNA (right). t-SNE and UMAP learn a low-dimensional embedding of cells that preserves the distribution of similarities that are quantified based on d or d' alone, which renders certain cell types indistinguishable to either modality. In this example, blue and red cells cannot be distinguished based on their measured surface proteins, and green and black cells overlap in transcriptomic space. In JVis we generalize t-SNE and UMAP to learn a joint embedding that preserves similarities in all modalities at the same time. We integrate d and d' in a convex combination of KL divergences (j-SNE) or cross entropies (j-UMAP) between corresponding similarities in low and high-dimensional space. An arrangement of cells that minimizes this convex combination with simultaneously learned weights takes into account similarities and differences in both mRNA and surface protein expression to more accurately represent cellular identity (middle).

wards individual modalities. In j-UMAP we generalize UMAP to multimodal data analogously, minimizing a convex combination of cross entropies instead of KL divergences. We jointly optimize the location of points in the embedding and the importance coefficients α of modalities through an alternating optimization scheme: We fix coefficients α and find the best point locations by gradient descent, and in turn find optimal coefficients α for fixed locations by solving a convex optimization problem.

5.1.2 Generalizing t-SNE to multimodal data

Given n data points $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, t-SNE is a nonlinear dimensionality reduction technique that aims to learn an embedding $\mathcal{E} = \{y_1, y_2, \ldots, y_n\}$ in a low-dimensional space that preserves the distribution of point similarities. Based on a given metric d that measures the dissimilarity between pairs of points, t-SNE first computes joint probabilities p_{ij} that

quantify the similarity between x_i and x_j:

$$p_{j|i} = \frac{\exp(-d(x_i, x_j)^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-d(x_i, x_k)^2/2\sigma_i^2)}, \qquad p_{i|i} = 0, \tag{5.2}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}, \tag{5.3}$$

where σ_i is the bandwidth of the Gaussion kernels centered at data point x_i. Given a predefined *perplexity* u, bandwidths σ_i are chosen such that the perplexity of conditional distributions P_i equal to u. The perplexity is often set to a value in the range between 5 and 50 (van der Maaten and Hinton, 2008). t-SNE uses the normalized Student-t kernel with a single degree of freedom to measure similarities q_{ij} in the embedding $\mathcal{E}$:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k\neq l}(1 + \|y_l - y_k\|^2)^{-1}}, \qquad q_{ii} = 0. \tag{5.4}$$

To determine the location of points in the embedding $\mathcal{E}$ that preserve the original similarities p_{ij} as well as possible, t-SNE seeks to minimize the Kullback-Leibler (KL) divergence of distribution $P = (p_{ij})$ from distribution $Q = (q_{ij})$.

$$C(\mathcal{E}) = KL(P||Q) = \sum_{i\neq j} p_{ij}\log\frac{p_{ij}}{q_{ij}}. \tag{5.5}$$

The minimization of equation (5.5) is performed using a gradient descent algorithm. The gradient is given by

$$\frac{\partial C}{\partial y_i} = 4\sum_{j\neq i}(p_{ij} - q_{ij})q_{ij}Z(y_i - y_j), \tag{5.6}$$

where $Z = \sum_{k\neq l}(1 + \|y_k - y_l\|^2)^{-1}$. While a naive implementation of t-SNE has runtime complexity $\mathcal{O}(n^2)$, the algorithm proposed in van der Maaten (2014) uses the Barnes-Hut algorithm (Barnes and Hut, 1986) and a sparse approximation of point similarities to reduce its complexity to $\mathcal{O}(n\log n)$ which allows it to scale to data sets comprising hundreds of thousands of cells.

We generalize t-SNE to the joint dimensionality reduction of multimodal data in j-SNE as follows. Given K-modal data points $X = \{x_1^{(k)}, x_2^{(k)}, \ldots, x_n^{(k)}\}, k = 1, 2, \ldots, K$, we want to learn a joint embedding $\mathcal{E} = \{y_1, y_2, \ldots, y_n\}$ that preserves similarities between points in X, analog to t-SNE, but in all modalities at the same time. Here, probability p_{ij}^k measures the similarity of points x_i^k and x_j^k with respect to modality k. Generalizing the objective of t-SNE, in j-SNE we aim to minimize the KL divergence of distributions $P^{(k)}$ for each modality k from distribution Q:

$$C(\mathcal{E}) = \sum_k \alpha_k KL(P^{(k)}||Q) + \lambda\sum_k \alpha_k\log\alpha_k \tag{5.7}$$

$$= \sum_k\sum_{i\neq j}\alpha_k p_{ij}^{(k)}\log\frac{p_{ij}^{(k)}}{q_{ij}} + \lambda\sum_k\alpha_k\log\alpha_k, \tag{5.8}$$

where $\alpha_k \geq 0$, $\sum_k \alpha_k = 1$, and $\lambda > 0$ is a regularization parameter. The first term of (5.7) denotes a convex combination of $\mathrm{KL}(P^{(k)}||Q), k = 1, 2, \ldots, K$, in which coefficients α_k represents the importance (contribution) of modality k to the joint embedding. That is, the larger α_k, the stronger the influence of modality k on the final location of points in the embedding. The second term of the objective function is a regularization term that prevents the joint embedding from being biased towards individual modalities.

Optimization algorithm

We use an alternating optimization scheme to minimize (5.8) jointly over the location of points y_i and the weighting of modalities α. In each iteration, we first fix α and use t-SNE to find points y in the joint embedding, after which we find the best (according to (5.8)) weighting α for fixed points y.

Initialization of α. We initially give uniform weights α to all modalities, i.e.,

$$\alpha_k = 1/K, \quad \text{for all} \quad k = 1, 2, \ldots, K. \tag{5.9}$$

Step 1: Fix α and optimize over y. Similar to conventional t-SNE, in j-SNE we employ the gradient descent algorithm to minimize (5.8). The gradient is given by:

$$\frac{\partial C}{\partial y_i} = 4 \sum_k \sum_{j \neq i} \alpha_k (p_{ij}^{(k)} - q_{ij}) q_{ij} Z (y_i - y_j) = 4 \sum_{j \neq i} \big(\sum_k \alpha_k p_{ij}^{(k)} - q_{ij} \big) q_{ij} Z (y_i - y_j). \tag{5.10}$$

Comparing gradient (5.10) to the original gradient (5.6), the former can be obtained by replacing distribution P used in (5.6) by the convex combination of distributions $P^{(k)}$, i.e., $P = \sum_k \alpha_k P^{(k)}$. In other words j-SNE, the joint t-SNE of multiple modalities, can be computed by applying conventional (unimodal) t-SNE to the convex combination of distributions $P^{(k)}$ for all modalities k.

Step 2: Fix y and optimize over α. Given a joint embedding of points y, problem (5.8) is a special case of the following problem:

$$\begin{aligned} \min_{w} \quad & a_i \alpha_i + \lambda \sum_k \alpha_k \log \alpha_k \\ \text{subject to} \quad & \sum_k \alpha_k = 1, \\ & \alpha_k \geq 0, \qquad k = 1, 2, \ldots, K. \end{aligned} \tag{5.11}$$

This problem has a convex objective and linear constraints. We derive a closed form solution using the Karush-Kuhn-Tucker (KKT) conditions. The Lagrangian function corresponding to the constrained optimization problem (5.11) is given by

$$\mathcal{L}(\alpha, u, v) = a_i \alpha_i + \lambda \sum_k \alpha_k \log \alpha_k + u \big(\sum_k \alpha_k - 1 \big) - \sum_k v_k \alpha_k.$$

The KKT conditions are given by

$$\begin{cases} \frac{\partial \mathcal{L}(\alpha,u,v)}{\partial \alpha_k} = a_k + \lambda(1+\log\alpha_k) + u - v_k = 0, \\ v_k\alpha_k = 0, \\ \sum_k \alpha_k = 1, \alpha_k \geq 0, \\ v_k \geq 0, \end{cases}$$

for all $k = 1, 2, \ldots, K$. Assuming non-zero contributions of each modality, i.e. strictly positive values for α, we have $v_k = 0$ for all k and $\alpha_k = \exp\{(-a_i - u)/\lambda - 1\}$. Together with the constraint $\sum_k \alpha_k = 1$, we obtain

$$\alpha_k = \frac{m_k}{\sum_k m_k},$$

where $m_k = \exp\{-a_i/\lambda - 1\}$.

We alternate iteratively between steps 1 and 2 until the improvement in the objective value falls below a predefined error threshold ϵ or until a maximum number *maxIter* of iterations has been reached. Here later iterations of j-SNE are faster than the first one because the distributions $P^{(k)}$ do not need to be recomputed.

Note that when $\lambda = 0$, the optimal value of problem (5.11) is $\min_i\{b_i\}$, and an optimal solution is $\alpha_{i^*} = 1, \alpha_j = 0$ for all $j \neq i^*$, where $b_{i^*} = \min_i\{b_i\}$. This solution is unique if $b_{i^*} = \min_i\{b_i\}$ is unique. In this case, the optimal joint embedding will converge to the optimal embedding of a single modality.

5.1.3 Generalizing UMAP to multimodal data

UMAP uses different definitions of high and low-dimensional similarities p_{ij} and q_{ij}, respectively. In particular, UMAP similarities $p_{j|i}$ are defined and symmetrized to give p_{ij} as follows:

$$\begin{aligned} p_{j|i} &= \exp[(-d(x_i, x_j) - \rho_i)/\sigma_i)], \qquad p_{i|i} = 0, \\ p_{ij} &= p_{j|i} + p_{i|j} - p_{j|i}p_{i|j}, \end{aligned}$$

where ρ_i is the distance to the nearest neighbor of x_i, and σ_i is the normalizing factor which is found through binary search using a criteria similar to the perplexity-based selection of bandwidth in t-SNE. The low-dimensional similarities are defined as:

$$q_{ij} = \left(1 + a\left\|y_i - y_j\right\|_2^{2b}\right)^{-1},$$

where a and b are user-defined parameters with default values $a \approx 1.929$ and $b \approx 0.7915$. Note that setting $a = b = 1$ gives the Student t-distribution used to define low-dimensional similarities in t-SNE (equation (5.4)). In contrast to t-SNE, however, p_{ij} and q_{ij} are not further normalized. UMAP determines the low dimensional embedding by minimizing the cross entropy:

$$C(\mathcal{E}) = \mathrm{CE}(P, Q) = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) + (1 - p_{ij}) \log\left(\frac{1 - p_{ij}}{1 - q_{ij}}\right), \tag{5.12}$$

which is equivalent to the following objective function (ignoring constant terms):

$$C(\mathcal{E}) = \sum_{i \neq j} -p_{ij} \log q_{ij} - (1 - p_{ij}) \log(1 - q_{ij}). \tag{5.13}$$

Objective (5.13) can be minimized by stochastic gradient descent. Note, however, that UMAP uses negative sampling when trying to optimize (5.13) which can have a large effect on the final embedding (Böhm et al., 2020).

The generalization of UMAP to multimodal data is analogous to that of t-SNE. In particular, we minimize the cross entropy between distributions $P^{(k)}$ for each modality k and distribution Q:

$$C(\mathcal{E}) = \sum_k \alpha_k \mathrm{CE}(P^{(k)}||Q) + \lambda \sum_k \alpha_k \log \alpha_k, \tag{5.14}$$

where, as in the joint t-SNE objective (5.7), the first term denotes a convex combination with coefficients α_i and the second regularization term is weighted with parameter $\lambda > 0$.

Similar to j-SNE, we use an alternating optimization approach to minimize objective (5.14). In contrast to t-SNE, UMAP does not normalize distributions P and Q (compare equations (5.2)-(5.4)). To be able to set λ to an identical value across experiments, we therefore normalize coefficients in the first term of (5.14) by their maximum value computed in the first iteration. Specifically, we optimize function $\frac{10}{c} \sum_k \alpha_k \mathrm{CE}(P^{(k)}||Q) + \lambda \sum_k \alpha_k \log \alpha_k$, where $c = \max_k\{\mathrm{CE}(P^{(k)}||Q)\}$ is a constant computed in the first iteration.

5.2 Results

In all experiments, we set the maximal number of iterations in our alternating optimization approach to 10 (*maxIter*=10). Guided by the results of our simulation study and by visual inspection of known cell types, we set the regularization parameter λ to 3 for j-SNE and to 1 for j-UMAP in all experiments.

Following best practice in Luecken and Theis (2019), we used standard preprocessing of the input data including log-transformation of the expression matrix followed by principal component analysis (PCA) and applied j-SNE and j-UMAP as well as their conventional counterparts to 20 or 50 principle components. In all protein velocity experiments, preprocessed data was taken from Gorin et al. (2020), no further preprocessing was performed. We computed protein acceleration using the protaccel Python package introduced in Gorin et al. (2020).

5.2.1 Proof of concept

As proof of concept, we first demonstrate the ability of JVis to integrate modalities with different signal strengths. scRNA-seq, for example, often allows a finer mapping of cell states than single-cell ATAC-seq (Stuart et al., 2019). We used JVis to compute a joint embedding of accessible chromatin and gene expression measured simultaneously by SNARE-seq (Chen et al., 2019) in 1,047 single cells from cultured human cell lines BJ, H1, K562, and GM12878. Similar to the conventional t-SNE and UMAP embeddings of transcriptomes or chromatin state alone, our joint j-SNE and j-UMAP embeddings clearly separate cells into four distinct clusters (Supplemental Figure S23). Even when randomly shuffling gene

Table 5.1: Overview of the simulated data sets used in this study. Data sets were simulated using Splatter and vary in number of cells (#Cells), number of genes (#Genes), number of antibodies (#Ab), number of clusters (k), and the relative abundance of cell types that were either equal, or based on cell type abundances among peripheral blood mononuclear cells (PBMCs) in healthy individuals.

Name	#Cells (N)	#Genes (D)	#Ab	k	Relative abundances (G)
GeqN1k	1,000	33,538	49	5	(0.2, 0.2, 0.2, 0.2, 0.2)
GeqN5k	5,000	33,538	49	5	
GeqN1kD1k	1,000	1,000	49	5	
GeqN5kD1k	5,000	1,000	49	5	
GpbmcN1k	1,000	33,538	49	5	PBMCs: DC: 0.02, NK: 0.2, B: 0.1 Mono: 0.08, T: 0.6
GpbmcN5k	5,000	33,538	49	5	
GpbmcN1kD1k	1,000	1,000	49	5	
GpbmcN5kD1k	5,000	1,000	49	5	

expression measurements between cell lines BJ and H1 in a toy experiment, JVis employs chromatin accessibility to disentangle mixed mRNA measurements and separate all four cell lines (Figures 5.2).

5.2.2 JVis is more accurate than conventional t-SNE and UMAP

We compared the performance of JVis to conventional t-SNE and UMAP applied to the concatenation of modalities that were normalized by dividing them by the Frobenius norm of the count matrix and to the embedding obtained when assigning (fixed) uniform weights to each modality ($\alpha_i = 1/3$ in (5.1)).

Data sets and evaluation scores

To examine the effectiveness of the joint optimization scheme underlying JVis, we devise a simulation study following a similar strategy as Wang et al. (2020). We used Splatter (Zappia et al., 2017) to simulate joint gene and ADT counts based on model parameters estimated from a real CITE-seq data set (Mimitou et al., 2019) in which mRNA and surface protein (ADT) expression were measured in human peripheral blood mononuclear cells (PBMC). In particular, the same number of genes and antibodies were used, and ADT counts were simulated based on estimated dropout rate, library size, expression outlier, and dispersion across features. We added a third modality by duplicating gene expression measurements and randomly permuting expression vectors between a variable size random subset of cell. The larger the subset of cells, the larger the artificially introduced level of noise in this third modality. We generated eight synthetic multimodal data sets that vary in the relative abundance of (five) cell types, number of cells, and in the number of genes (Table 5.1).

We measured the accuracy of an embedding using two different metrics. We introduce the k-nearest neighbor index (KNI), which denotes the fraction of k-nearest neighbors in the embedding that are of the same type. A high KNI value indicates homogeneous neighborhoods of cell types, while a random mixing of cells would cause low KNI values. We used $k = 10$ if not specified otherwise and computed the average across all points. In addition, we used the Silhouette score (Rousseeuw, 1987) that ranges between -1 and 1 to measure

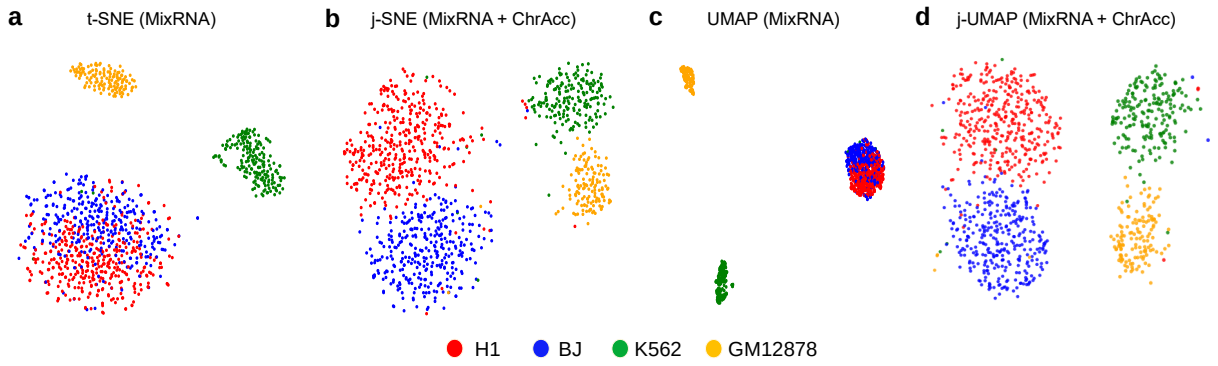

Figure 5.2: Visualization of perturbed SNARE-seq measurements. Accessible chromatin (ChrAcc) and gene expression was measured simultaneously in single cell from human cell lines BJ, H1, K562, and GM12878. Gene expression measurements were randomly shuffled between cell lines BJ and H1 (MixRNA). (**a**) Conventional t-SNE embedding of cells based on shuffled gene expression alone. (**b**) j-SNE visualization of shuffled gene expression and (unchanged) chromatin accessibility. (**c**) Conventional UMAP embedding of cells based on shuffled gene expression alone. (**d**) j-UMAP visualization of shuffled gene expression and (unchanged) chromatin accessibility.

how much cell types overlap (score 0) or how well separated (score 1) they are. We used the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) to measure the agreement between RNA and protein based clusterings.

Evaluation of JVis

In contrast to its conventional counterparts, j-SNE and j-UMAP learn weights for each modality from the data that reflect their relevance to the final embedding. Figures 5.3 and 5.4 show that these weights distinguish informative from noisy modalities. With an increasing amount of perturbation of the third modality, i.e. an increasing number of cells with shuffled gene expression, JVis assigns a lower weight to the corresponding modality. The rate of weight decrease (and the simultaneous increase mostly in ADT weight) is higher for data sets with a larger number of cells and, as expected, depends on the regularization coefficient λ. For λ close to 0, weights essentially include a single most informative modality (here ADT, see Table 5.2) (Supplemental Figure S24). Higher penalties associated with non-uniform weights result in a weaker adjustment of weights by the joint optimization scheme. The absolute adjustment of weights associated with cross entropy terms in j-UMAP is less pronounced than the adjustment of weights associated with KL divergences in j-SNE.

Figures 5.5, 5.6 demonstrate the benefit of borrowing information across modalities by the joint optimization scheme implemented in JVis. Compared to the normalized concatenation and the (uniform) averaging approach, the distinction between meaningful and noisy modalities in j-SNE and j-UMAP yields more accurate embeddings with respect to KNI score, across various noise levels. For data sets containing 5000 cells the separation of cell types in the embeddings obtained with (fixed) uniform weights continuously decreased with increasing noise levels. In contrast, the joint optimization scheme in j-SNE was able to retain a high accuracy on these data sets (Figure 5.5), especially for smaller penalties assigned to non-uniform weights, i.e. small values of λ. This is consistent with the sharper drop in the weight associated with the noise modality observed for data sets containing 5000 cells and

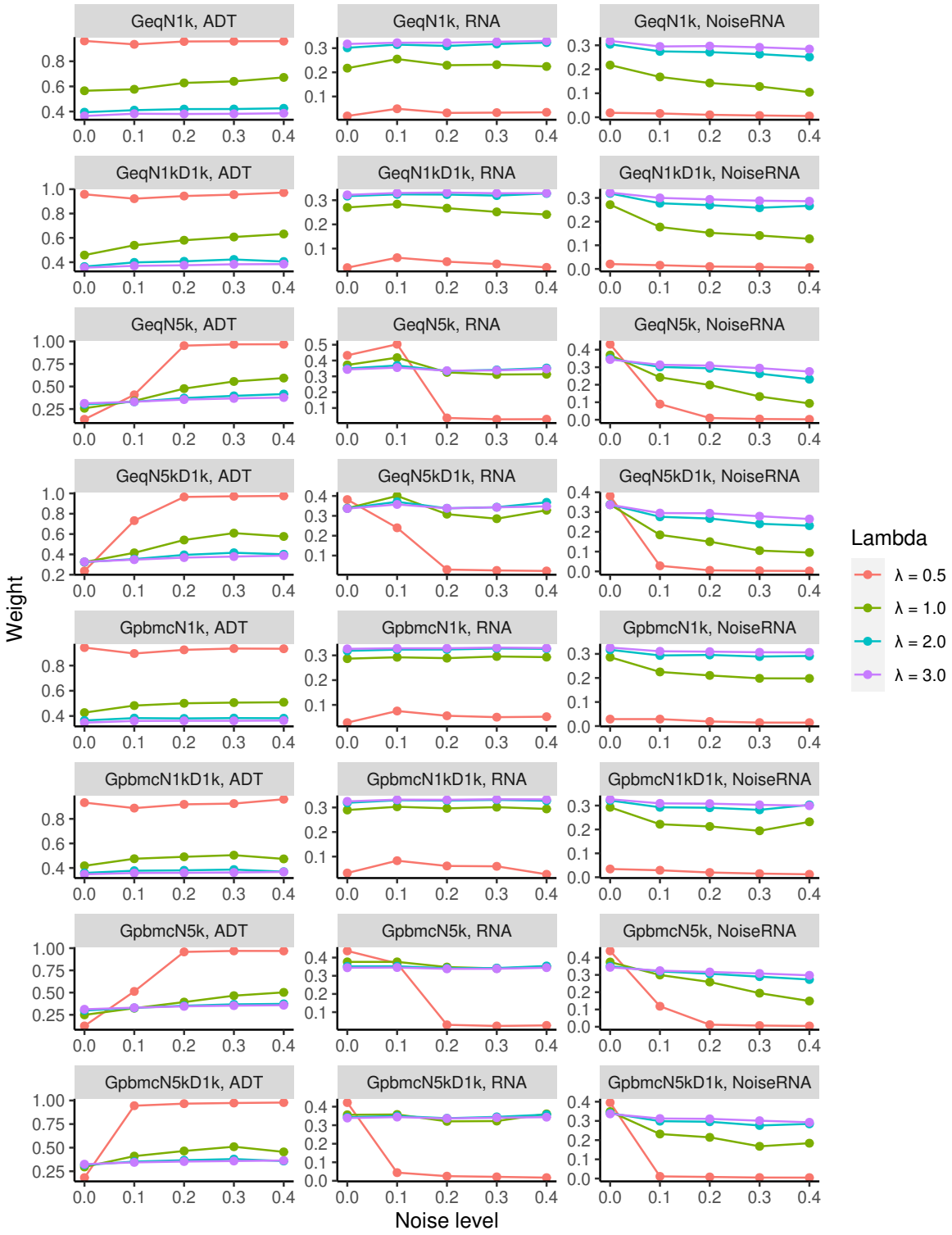

Figure 5.3: j-SNE modality weights for eight simulated data sets as function of noise. Weights (α) for modalities ADT (left), RNA (middle), and NoiseRNA (right) were computed using different regularization coefficients λ. Each data set is shown in one row.

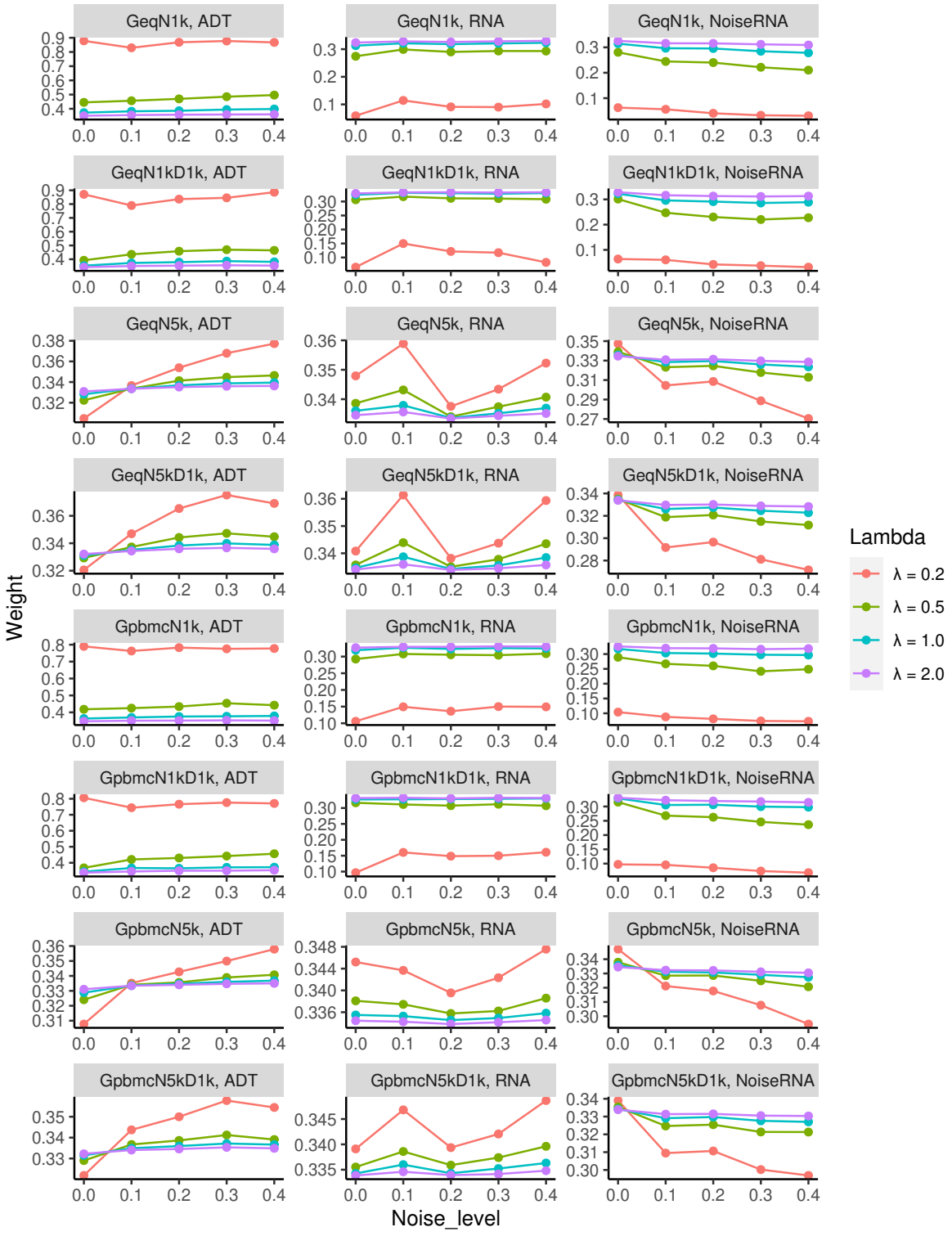

Figure 5.4: j-UMAP modality weights for eight simulated data sets as function of noise. Weights (α) for modalities ADT (left), RNA (middle), and NoiseRNA (right) were computed using different regularization coefficients λ. Each data set is shown in one row.

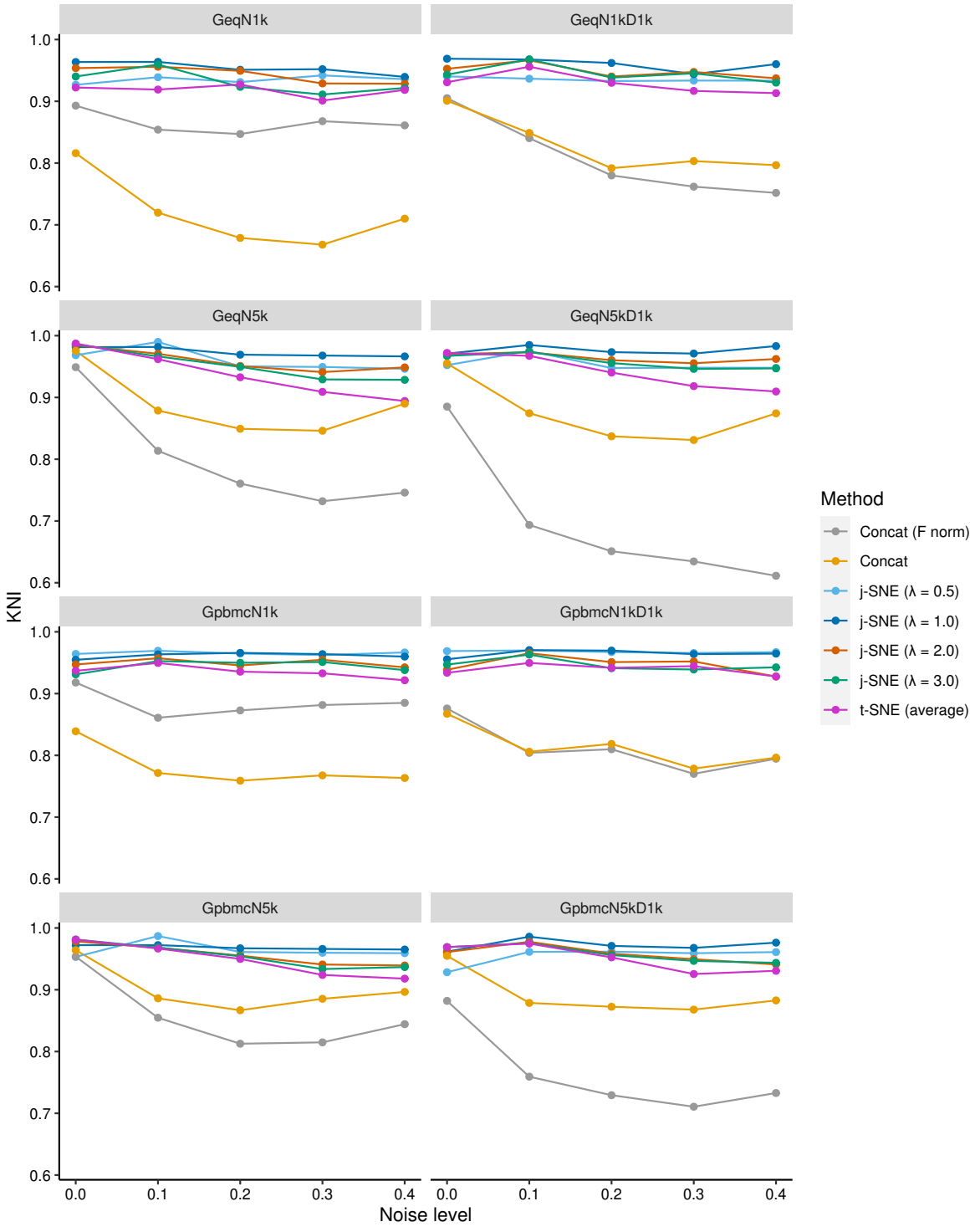

Figure 5.5: KNI values of embeddings computed by j-SNE and alternative methods on eight simulated data sets. Values are shown as a function of noise for different regularization coefficient λ used in j-SNE. Conventional t-SNE is run for uniform weights assigned to each modality ($\alpha_i = 1/3$) (t-SNE (average)), or on concatenated modalities (Concat) that are optionally normalized by the Frobenius norm (Concat (F norm)).

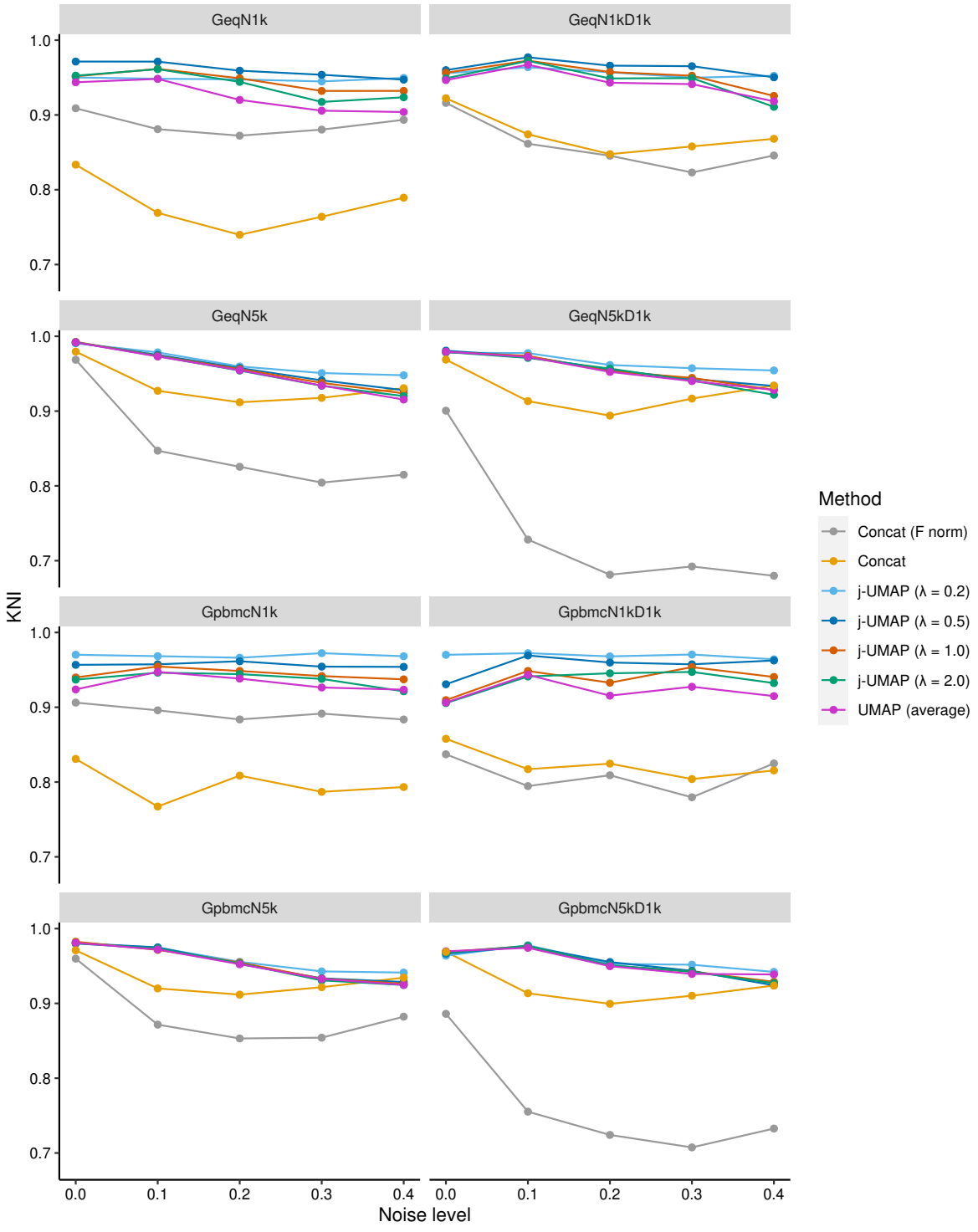

Figure 5.6: KNI values of embeddings computed by j-UMAP and alternative methods on eight simulated data sets. Values are shown as a function of noise for different regularization coefficient λ used in j-UMAP. Conventional UMAP is run for uniform weights assigned to each modality ($\alpha_i = 1/3$) (UMAP (average)), or on concatenated modalities (Concat) that are optionally normalized by the Frobenius norm (Concat (F norm)).

small values of λ (Figure 5.3). For j-UMAP, on the other hand, computed weights were close to uniform (for $\lambda \geq 0.5$) on data sets GpbmcN5k and GpbmcN5kD1k (Figure 5.4) and thus yielded embeddings with similar accuracy as the uniform weighting scheme on these data sets (Figure 5.6). The normalized concatenation approach works reasonably well on data sets with 1000 cells and large number of genes (N1k), but its performance varies substantially between different data sets and is even less accurate than its unnormalized version on data sets with 5000 cells. On most data sets, concatenation-based approaches show a sharp initial drop in accuracy for small levels of noise. Similarity, an evaluation of JVis using the Silhouette score provides consistent results as KNI (Supplemental Figures S25, S26). Together, they imply that JVis produces more accurate low-dimensional embedding of cells than the conventional methods.

5.2.3 JVis utilizes multi-modal data to resolve subtle transcriptomic difference

t-SNE and UMAP often produce embeddings that are in good agreement with known cell types or cell types computed by unsupervised clustering (Blondel et al., 2008; Kiselev et al., 2017) of high-dimensional molecular measurements such as mRNA expression. The simultaneous measurement of multiple types of molecules such as RNA and protein can refine cell types and JVis seeks to capture this refinement in their low-dimensional embedding. We compared unimodal and multimodal embeddings of mRNA and surface protein (ADT) expression measured in 4,292 healthy human PBMCs (Mimitou et al., 2019) and in 8,617 cord blood mononuclear cells (CBMC) (Stoeckius et al., 2017) using CITE-seq (Stoeckius et al., 2017). Cell type labels were inferred by methods Specter (Do et al., 2021) or CiteFuse (Kim et al., 2020), which have recently been introduced for the joint clustering of CITE-seq data.

Consistent with observations in Do et al. (2021); Kim et al. (2020), t-SNE and UMAP visualizations of transcriptomic data alone does not show a clear distinction of CD4+ T cells and CD8+ T cells in the CBMC data set, while the embedding of protein expression mixes dendritic cells with CD14+ cells (Figure 5.7, Supplemental Figure S27). In contrast, JVis makes use of both modalities to compute a joint embedding that accurately separates CD4+ and CD8+ T cells as well as dendritic and CD14+ cells. Again, we confirm the visual interpretation quantitatively using the same metrics as above (Table 5.2). The joint embedding of mRNA and ADT by JVis yields substantially larger Silhouette scores than the two unimodal t-SNE and UMAP embeddings.

Similarly, the joint embeddings of cells in the PBMC data set by JVis separate naïve and memory CD4+ T cell that are mixed in the ADT based t-SNE and UMAP embeddings as well as CD4+ and CD8+ T cells that are mixed in the mRNA based embeddings (Supplemental Figures S28, S29). Again, joint embeddings are more accurate in terms of Silhouette scores than unimodal embeddings (Table 5.2), even though overall the additional information provided by RNA measurements is limited relative to ADT counts on this data set.

5.2.4 JVis improves the visualization of joint velocity landscapes of protein and RNA

RNA velocity (La Manno et al., 2018) describes the rate of change of mRNA abundance estimated from the ratio of mature and pre-mRNA. While RNA velocity points to the future state of a cell, the recently introduced protein velocity (Gorin et al., 2020) extends this concept and utilizes the joint measurement of RNA and protein abundance to infer the

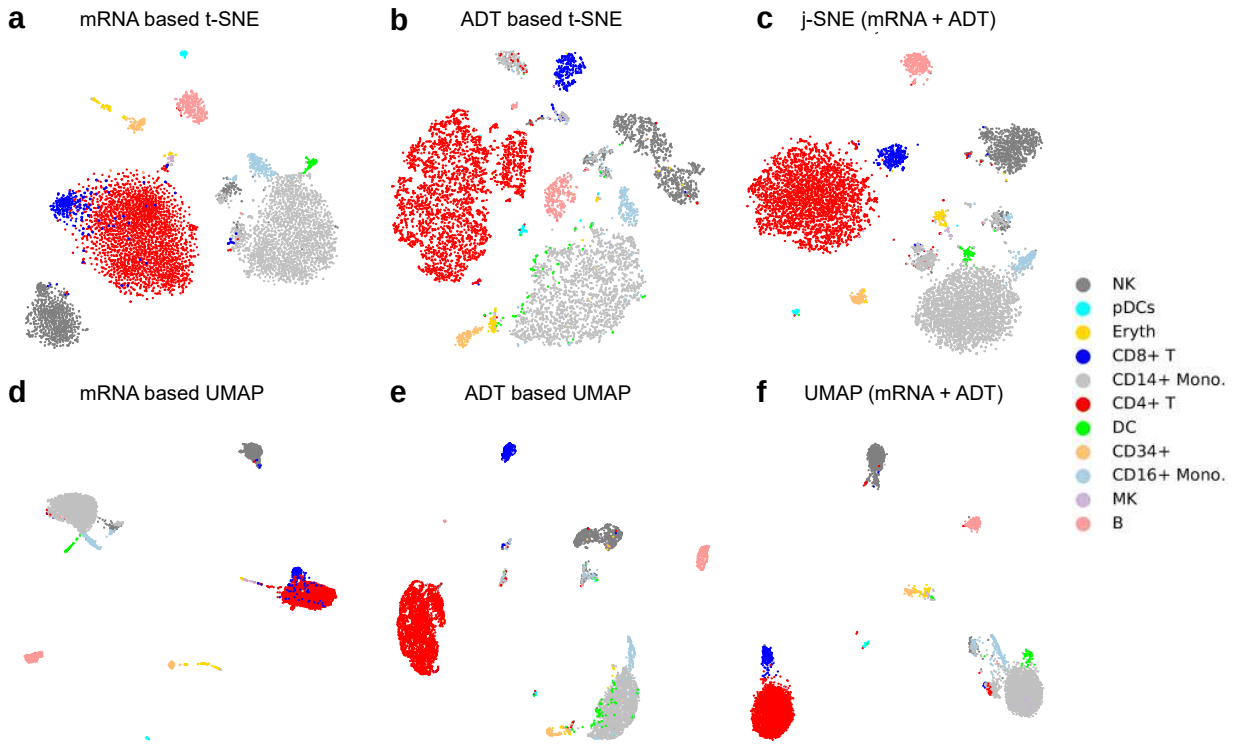

Figure 5.7: Visualizations of CBM cells. Cluster labels were identified by Specter. *First row:* t-SNE/j-SNE embeddings were computed from RNA measurements alone (**a**), protein expression (ADT) alone (**b**), or jointly from both (**c**). *Second row:* UMAP/j-UMAP embeddings were computed from RNA measurements alone (**d**), protein expression (ADT) alone (**e**), or jointly from both (**f**).

Table 5.2: Comparison of joint and unimodal embeddings on the PBMC and CBMC data sets. KNI denotes the fraction on k-nearest neighbors in the embedding that are of the same type, averaged over all cells. Larger Silhouette scores indicate a better separation of cell types. Only cells assigned identical labels based on joint clusterings by CiteFuse and Specter are considered in the evaluation.

	CBMC		PBMC	
Method	KNI	Silhouette	KNI	Silhouette
j-SNE	**0.998**	**0.432**	**0.985**	**0.383**
RNA based t-SNE	0.978	0.343	0.960	0.196
ADT based t-SNE	0.989	0.270	0.949	0.366
j-UMAP	**0.998**	**0.578**	**0.985**	**0.525**
RNA based UMAP	0.980	0.487	0.960	0.111
ADT based UMAP	0.982	0.468	0.948	0.511

past, present, and future state of a cell. In Gorin et al. (2020), the authors used PCA and t-SNE to visualize RNA and protein velocity as well as the resulting *protein acceleration* in six PBMC data sets that were generated using four different technologies: CITE-seq, REAP-seq (Peterson et al., 2017), ECCITE-seq (Mimitou et al., 2019) (data sets "CTCL", a

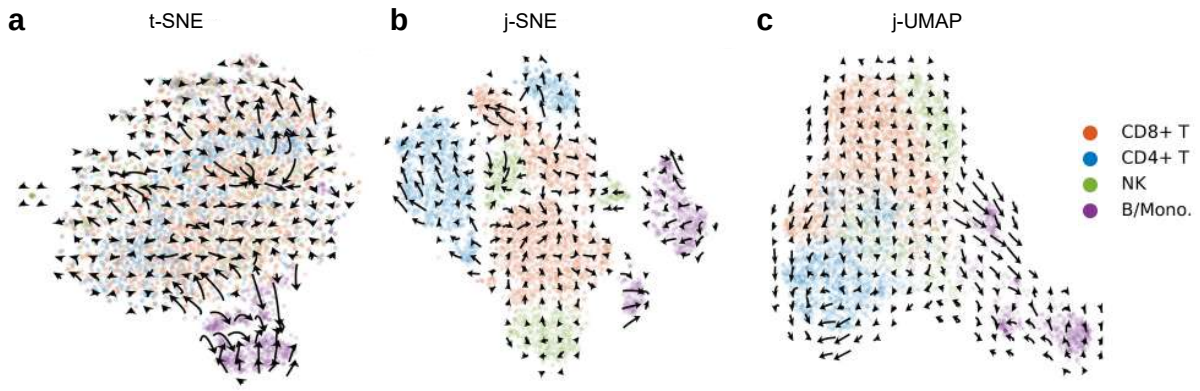

Figure 5.8: Protein acceleration in ECCITE-seq (ctrl) data set projected into transcriptome-based t-SNE (**a**), and joint mRNA and surface protein based embeddings j-SNE (**b**) and j-UMAP (**c**).

cutaneous T cell lymphoma patient, and "ctrl", a healthy control), and 10X Genomics (data sets 1k and 10k). The authors observed strong velocity signals offered by the CITE-seq and 10x Genomics technologies, while REAP-seq and ECCITE-seq yielded noisier acceleration landscapes. Both RNA and protein velocity, however, were projected into the same t-SNE embedding of transcriptomic measurements alone, rendering their interpretation difficult. We therefore repeated the analysis of the six different data sets but projected velocities into the joint embedding of both modalities computed by JVis. The noisy acceleration landscapes observed in Gorin et al. (2020) in the ECCITE-seq and REAP-seq data sets become aligned across cell types in their joint embeddings by JVis (Figures 5.8 and Supplemental Figures S30, S31). Consistent with the improved distinction of transcriptionally similar CD4 and CD8 T cells in the joint embeddings above, acceleration landscapes in all six data sets are projected onto an embedding that more clearly separates CD4 and CD8 T cells compared to the original ones proposed in Gorin et al. (2020) (Figure 5.8 and Supplemental Figures S30-S34).

The noisy acceleration landscapes reported in Gorin et al. (2020) for the REAP-seq and ECCITE-seq data sets might be a result of the larger number of measured surface proteins (44 and 49 antibodies versus 13 and 17 antibodies in CITE-seq and 10X, respectively) that provide a finer distinction of subpopulations of cells. In fact, we observed lower agreement between RNA and protein based clusterings for the ECCITE-seq data set shown in Figure 5.8 (ARI 0.21), compared to the clusterings obtained from the two modalities in the CITE-seq data set that agree well (ARI 0.82). Here clusterings of cells were computed using the Louvain algorithm (Blondel et al., 2008) where the resolution parameter is tuned to match the number of annotated cell types. Since protein acceleration is computed from both RNA and protein abundances, their joint embedding can help to reduce visualization artifacts that arise when protein velocities are projected into a purely transcriptome based t-SNE embedding as in Gorin et al. (2020).

5.2.5 Scalability

The complexity of Barnes-Hut based t-SNE is $\mathcal{O}(n \log n)$, where n is the number of input cells (van der Maaten, 2014). Although no theoretical complexity bounds have been established for UMAP, its empirical complexity is $\mathcal{O}(n^{1.14})$ (McInnes et al., 2018). Since

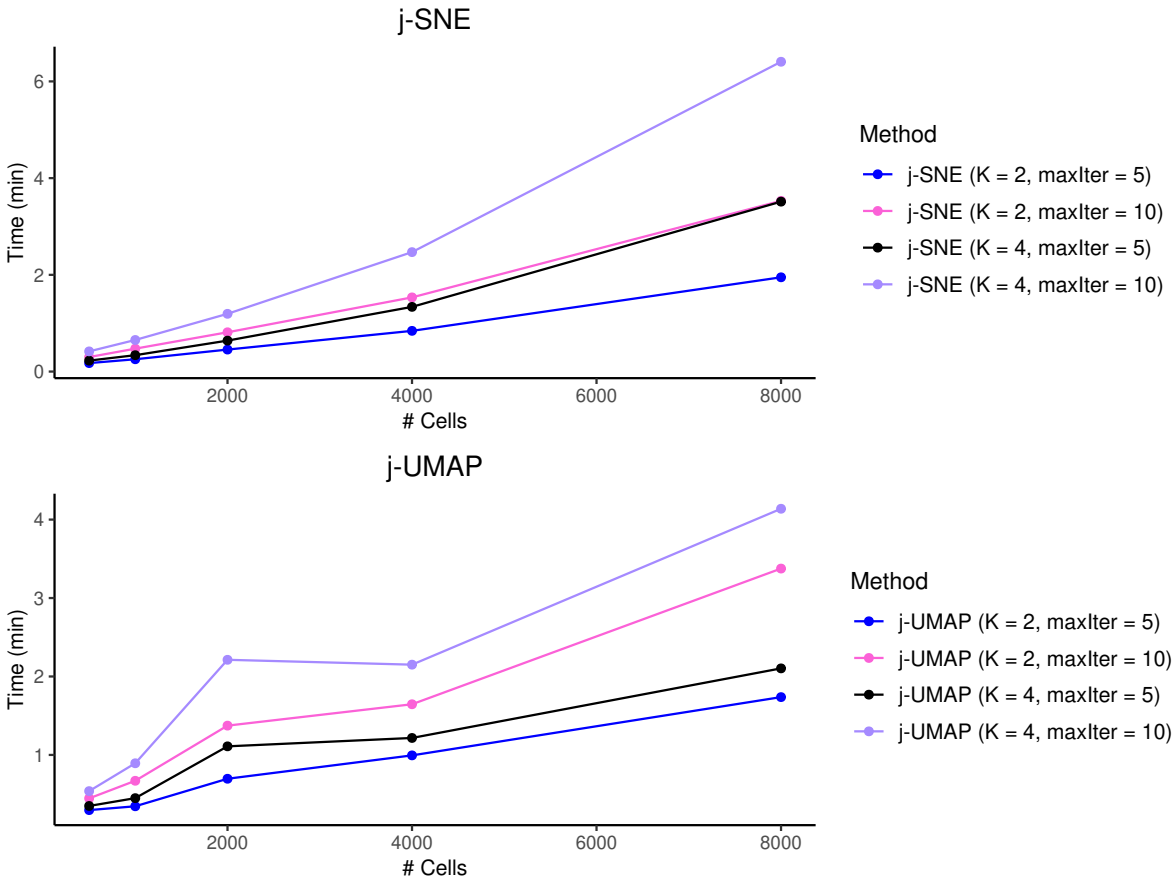

Figure 5.9: CPU times of j-SNE and j-UMAP as a function of the number of cells. Different size data sets were subsampled from the CBMC CITE-seq data set. Running time are shown using 5 or 10 iterations in both methods (controlled by parameter *maxIter*), on data sets containing K=2 and K=4 modalities. Two additional modalities were generated by duplicating and randomly shuffling the original two modalities. Note that the switching from exact to approximate nearest neighbor search in UMAP for data sets with more than $3,000$ cells slowed the increase in running time of UMAP, or even decreased it, from $2,000$ and $4,000$ cells.

in addition the alternating minimization in j-SNE and j-UMAP requires only a few iterations of (conventional) t-SNE and UMAP calculations to converge to its final estimation of modality weights (Supplemental Figure S35), JVis is expected to scale well to large data sets. For example, it took JVis less than 5 minutes to compute an embedding of the 10,000 cells contained in the largest data set used in this study (10x 10k). Here all experiments were performed on an Intel Xeon CPU at 2.30 GHz with 320 GB memory. Running time of j-SNE and j-UMAP shown in Figure 5.9 as a function of number of cells with 2 and 4 simulated modalities demonstrate practicability of both approaches in the analysis of larger and more complex multimodal data sets. The scalability of our approach to large data sets can be further improved by combining it with the recently proposed FFT-accelerated Interpolation-based t-SNE method (Linderman et al., 2019).

5.2.6 Availability of data and materials

The SNARE-seq and CBMC CITE-seq data sets were downloaded from Gene Expression Omnibus with accession codes GSE126074 and GSE126310, respectively. The six data sets used in the protein acceleration experiments were from Gorin et al. (2020). The implementations of j-SNE and j-UMAP are based on the *scikit-learn* v0.23.1 library (Pedregosa et al., 2011) and the UMAP v0.4.5 Python package (McInnes et al., 2018), respectively. The JVis Python package can be installed through PyPi and its open-source code is maintained at `https://github.com/canzarlab/JVis-learn`. Python scripts to reproduce all results in this paper are available at `https://github.com/canzarlab/JVis_paper`.

5.3 Conclusions

t-SNE and UMAP are routinely used to explore high-dimensional measurements of single cells in low-dimensional space. We have introduced method JVis that generalizes t-SNE and UMAP to the joint visualization of single-cell multimodal omics data. We have demonstrated that JVis combines multiple omics measurements of single cells into a unified embedding that exploits relationships among them that are not visible when applying conventional t-SNE or UMAP to each modality separately. Higher expected levels of noise in the measurements can be counteracted by smaller regularization coefficients λ that allow to downweight noisy modalities. Not surprisingly, projecting RNA and protein velocities into the joint embedding of both modalities yielded less noisy acceleration landscapes compared to embeddings of mRNA measurements alone. We therefore anticipate that JVis will aid in the meaningful visual interpretation of data generated by emerging multimodal omics technologies such as CITE-seq (Stoeckius et al., 2017) and SHARE-seq (Ma et al., 2020), the latter allowing to combine RNA velocity with *chromatin potential*.

Chapter 6

Conclusion and outlook

6.1 Conclusion

In this thesis, we developed diverse computational methods for scRNA-seq and multimodal omics data. In Chapter 2 we presented Specter for clustering ultra-large scRNA-seq and multimodal omics data. We showed the superior performance of Specter across comprehensive benchmarks on 45 real and simulated data sets. Specter facilitates the identification of rare cell types and resolves subtle transcriptomic differences in multimodal data. In Chapter 3 we proposed Sphetcher, a mathematical method that efficiently picks the representative cells and highlights the presence of rare cell types in the data. Sphetcher enables the shift from a "more data, less algorithm" paradigm to a "less (but accurate) data, more algorithm" regime. We then introduced Trajan in Chapter 4, a method allows for the first time the alignment of complex single-cell trajectories. Trajan automatically identifies and aligns core paths without prior information as used in the previous approach. Finally, in Chapter 5 we proposed j-SNE and j-UMAP as the natural generalizations to the joint visualization of multimodal omics data. On comprehensive benchmarks, we showed that j-SNE and j-UMAP produce unified embeddings that better agree with cell types and harmonize RNA velocity and protein acceleration landscapes than the conventional approaches.

6.2 Outlook

Extend Sphetcher for multimodal omics data

In Chapter 3 we introduced Sphetcher for downsampling large data sets. Given n data points $X = \{x_1, x_2, \ldots, x_n\}$ and a metric d that measures the dissimilarity between pairs of cells, Sphetcher finds a sketch $X_S \subseteq X$ that minimizes the Hausdorff distances between the sketch and the full dataset defined as:

$$d_H(X_S, X) = \max_{x \in X} \Big\{ \min_{y \in X_S} d(x, y) \Big\}.$$

Sphetcher is applicable to a broad range of data including scRNA-seq and scATAC-seq as long as we can define a distance between two points. However the use of Sphetcher for multimodal data is not obvious because it is not very clear how to compute the distance between two points in multimodal data. One way to do it is to concatenate the feature (with or without normalization) across all modalities to create a single feature vector for each cell.

But we already pointed out that the concatenation strategy might not work well because this approach might be biased to the modality of large dimension and scale or it is not clear which normalized concatenation strategies should be selected. Another approach is to use the convex combination of distances, that is, we define the distance between two points x and y as $d(x, y) = \sum_k \alpha_k d(x^{(k)}, y^{(k)})$, where $\alpha_k \geq 0, \sum_k \alpha_k = 1$, and $x^{(k)}$ and $y^{(k)}$ are k-th modality of x and y, respectively. It is easy to show that the convex combination of metrics is also a metric. However in this case it is not clear how to choose the parameters α_k. One way to solve this issue is to cast it as the convex combination problem as we did in the generalization of visualization methods to multimodal data. We then find the sketch and parameters jointly through an alternative optimization scheme. Finally one could also adopt the concept of metric learning from multi-view data, for example, Singh et al. (2021) proposed a version of metric learning adapted for multimodal omics data. We can see that there are numerous ways to generalize Sphetcher to multimodal data and it remains to effectively evaluate these approaches. This can be done similarly to the validation of Sphetcher with computational methods used for unimodal data being replaced by multimodal methods. We anticipate a method for downsampling multimodal omics data is worthwhile due to the growth of large-scale multimodal projects with a recent data set measuring more than 200k cells (Hao et al., 2021).

Outlook for Trajan

In Chapter 4 we introduced Trajan and showed a practical use of Trajan in an alignment of trajectories describing human muscle differentiation and myogenic reprogramming. In the future we explore the usefulness of Trajan. Note that Trajan can be used as a metric to measure the similarity/dissimilarity between two trajectories. From this we can align the trajectories computed by two methods and we can point out the similarities/differences between the trajectories computed by the two. In addition, we can compare the same method with different sets of parameters used in the method (e.g., parameter *ncenter* in Monocle 2) to assess the stability and parameter sensitivity of the model.

Based on the comprehensive benchmark on 45 TI methods (Saelens et al., 2019), there is no best method and it is not obvious which method should be used for a specific data set. Hence, another application of Trajan is to select a method for a given data. Here we assume that a reference trajectory is given and it can be taken from a public trajectory database or constructed from the Human Cell Atlas. Then we can compute trajectories by various TI methods and compare them to the reference trajectory, and then select the one which matches well with the reference. In addition, Trajan can be used for integration of time-resolved data based on the alignment between cells in two data sets. Data integration across different conditions and technologies is one of the most important and challenging problems in single-cell genomics data.

Develop more accessible and integrated bioinformatics toolkit

The Specter software is written in Matlab, which might pose some challenges for the broad community especially for biologists without much computation background or access to Matlab. Therefore it is important to implement the package in programming languages such as R or Python, which are widely used in the single cell genomics community. Moreover, it is

also easier to integrate Specter with popular single-cell libraries such as Seurat (in R) and Scanpy (in Python).

Currently j-SNE is implemented based on the scikit-learn Python package. We are planning to integrate j-SNE and j-UMAP with Scanpy and explore compatibility with faster implementations of t-SNE such as openTSNE and Fit-SNE. Finally we want to integrate all packages in a toolkit which allows users to perform various single-cell analysis tasks ranging from preprocessing (Sphetcher) to downstream tasks such as clustering (Specter), visualization (j-SNE, j-UMAP), and alignment of single-cell trajectories (Trajan).

Appendix A

Supplementary Figures

A.1 Supplemental Figures: Specter

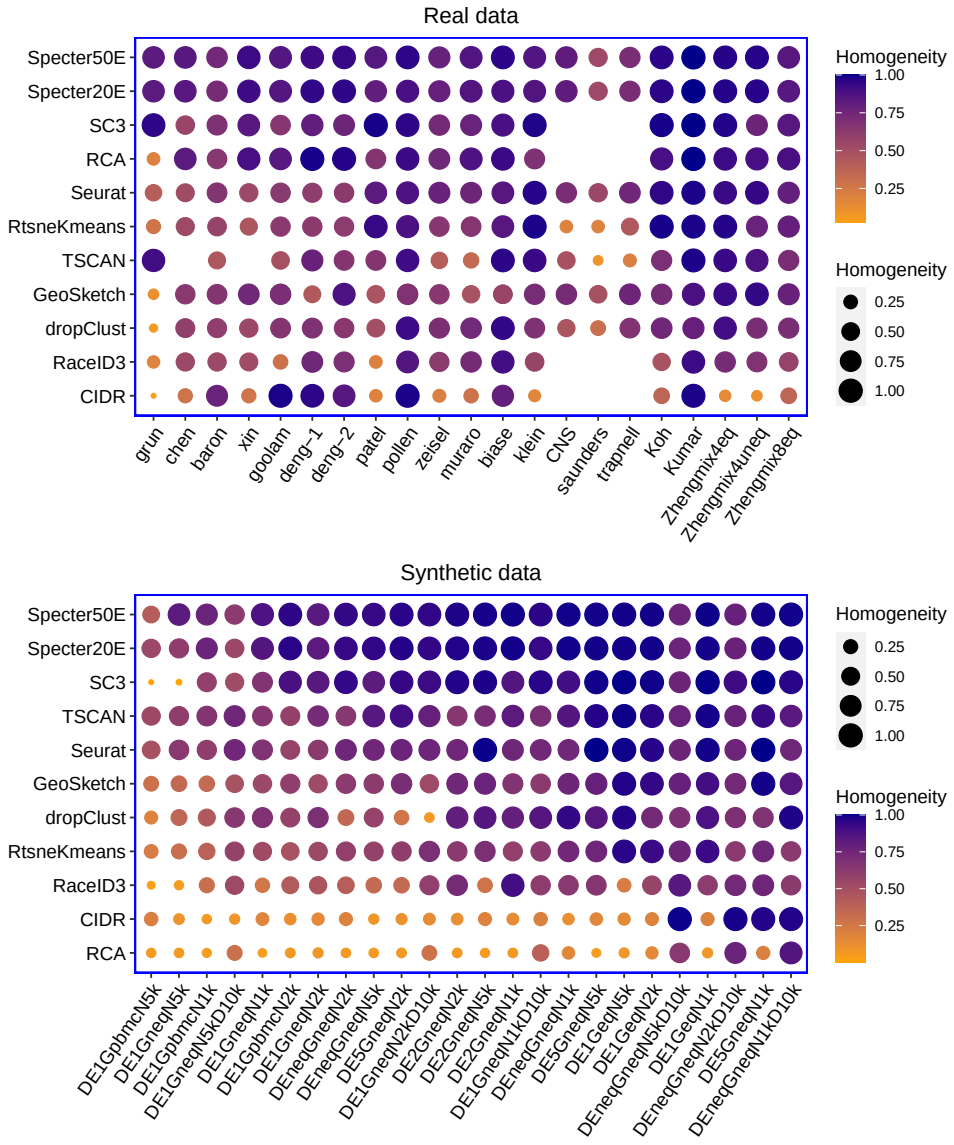

Figure S1: Clustering performance measured by homogeneity score of Specter and competing methods on real and synthetic data. Methods are ordered by mean homogeneity score across data sets decreasing from top to bottom. In the calculation of mean scores we excluded for each method the data sets where the method did not run successfully. Synthetic data sets are ordered from left to right by increasing mean homogeneity score over all methods. SC3 , RCA, RaceID3, and CIDR failed to run on the three largest data sets *CNS*, *saunders*, and *trapnell* due to insufficient memory. TSCAN failed to run on data sets *chen* and *skin* for unknown reasons. Geometric sketching refers to the Louvain clustering of 10% of the cells sampled using geometric sketching.

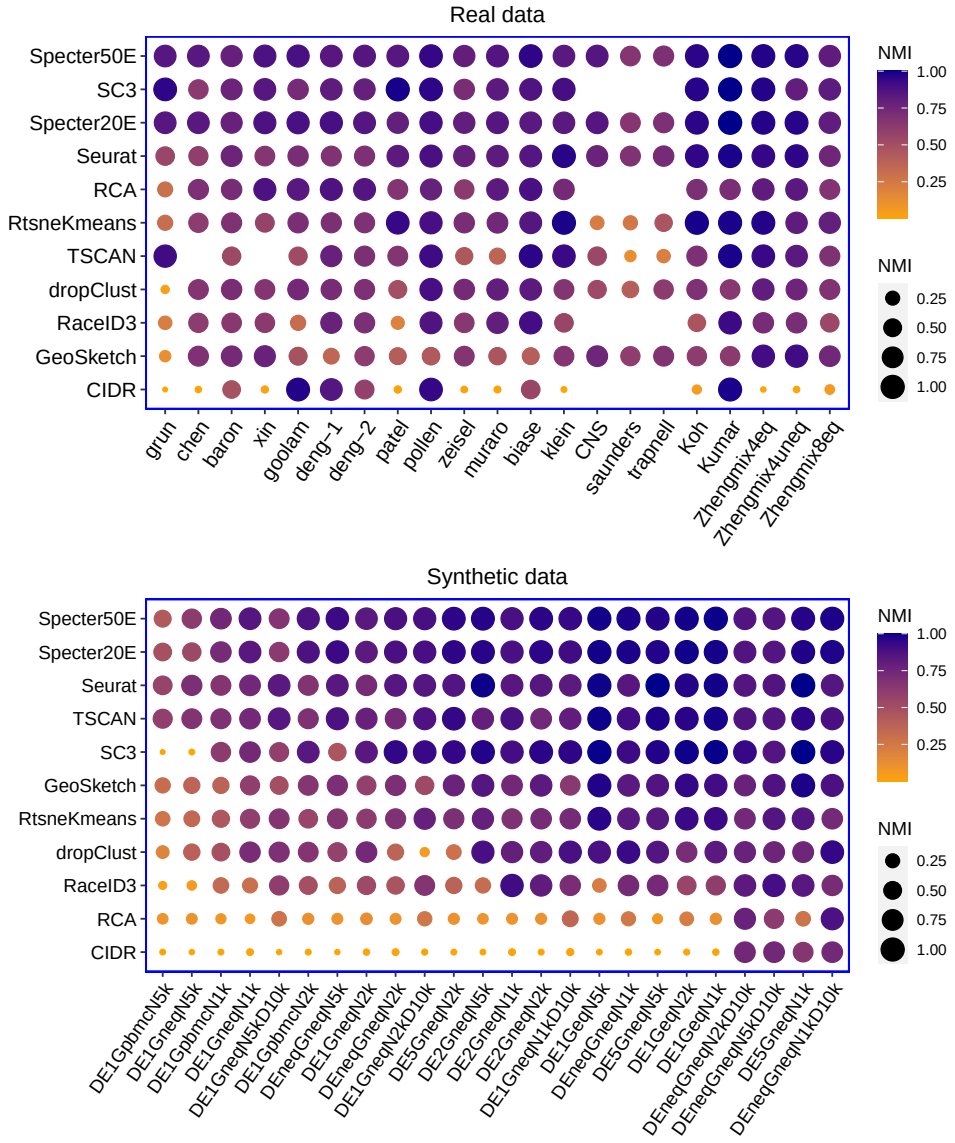

Figure S2: Clustering performance measured by NMI of Specter and competing methods on real and synthetic data. Methods are ordered by mean NMI across data sets decreasing from top to bottom. In the calculation of mean scores we excluded for each method the data sets where the method did not run successfully. Restricted to the same set of data sets as SC3, Specter20E was with a mean ARI of 0.87 marginally better than SC3 (mean ARI 0.85). Synthetic data sets are ordered from left to right by increasing mean NMI over all methods. SC3 , RCA, RaceID3, and CIDR failed to run on the three largest data sets *CNS*, *saunders*, and *trapnell* due to insufficient memory. TSCAN failed to run on data sets *chen* and *skin* for unknown reasons. Geometric sketching refers to the Louvain clustering of 10% of the cells sampled using geometric sketching.

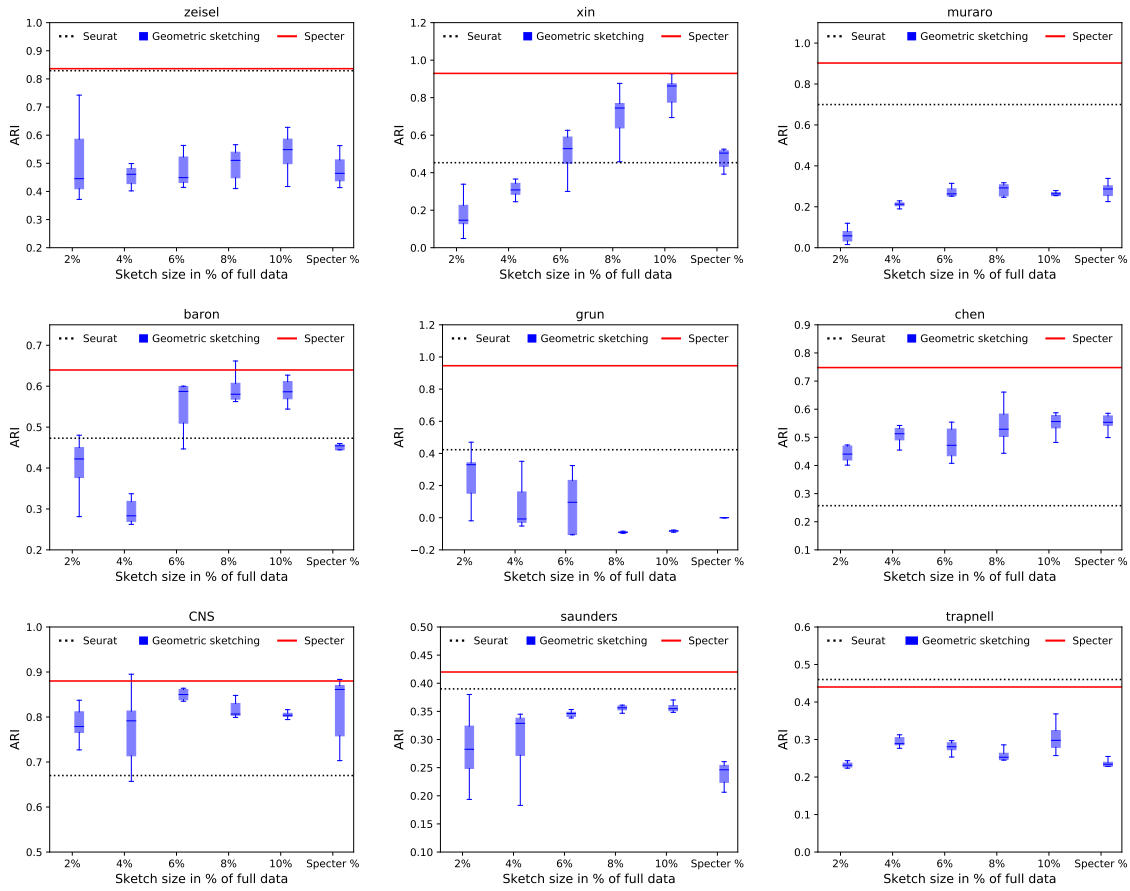

Figure S3: Accuracy (in ARI) of geometric sketching based Louvain clustering for varying sketch sizes. For each sketch size, the results of 10 random trials are shown. "Specter %" uses the same number of cells in the geometric sketch as Specter uses landmarks or cells in the selective sampling step (see the "Methods" section), whichever one is larger.

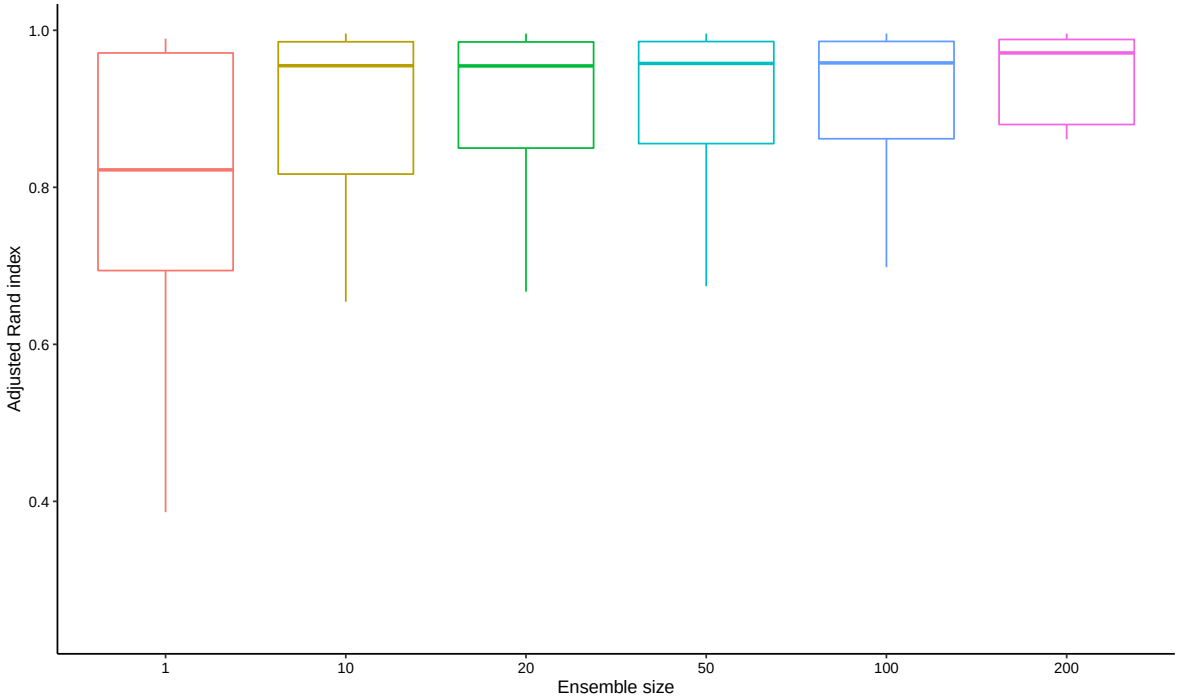

Figure S4: Accuracy of Specter vs. number of ensemble members. For each number of ensemble members, the box plot shows minimum, maximum, median, and first and third quartiles of ARI scores achieved by Specter on the 24 simulated data sets.

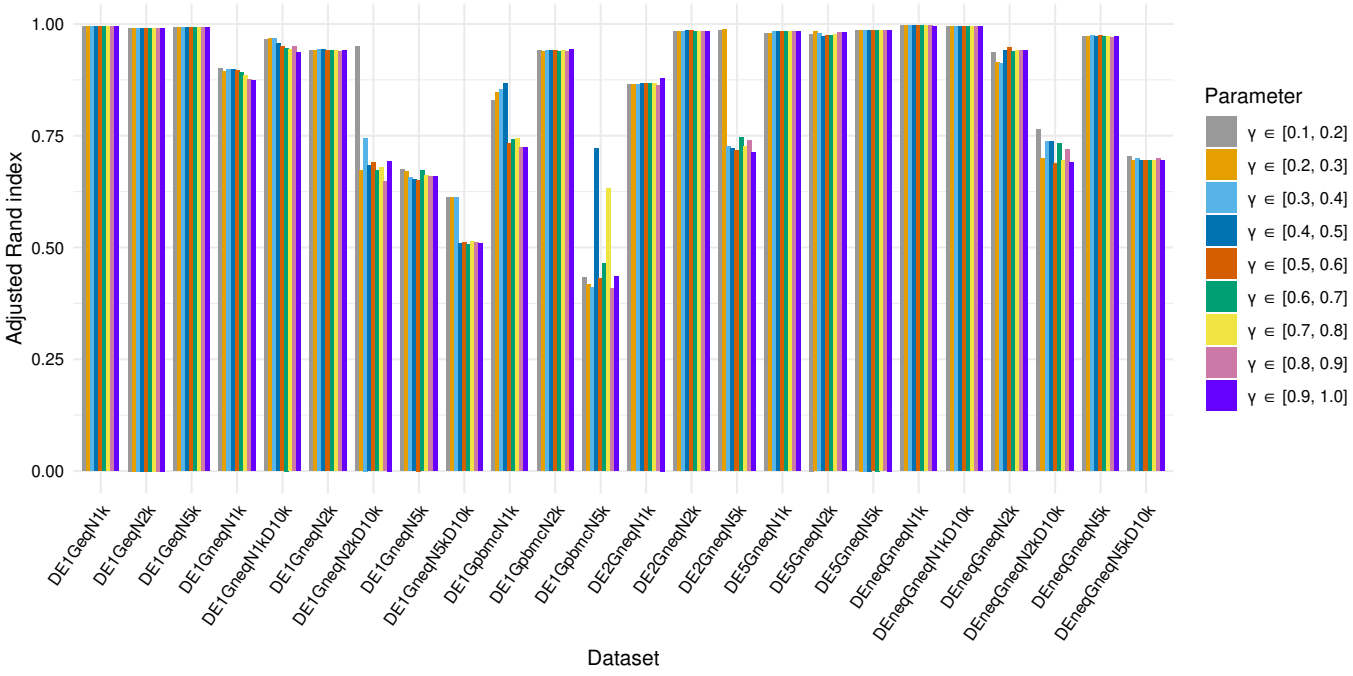

Figure S5: Robustness of Specter to choice of parameter γ. Across 24 synthetic data sets, Specter computed 50 ensemble members using different ranges for parameter γ. By default, Specter selects a $\gamma \in [0.1, 0.2]$ for each ensemble member.

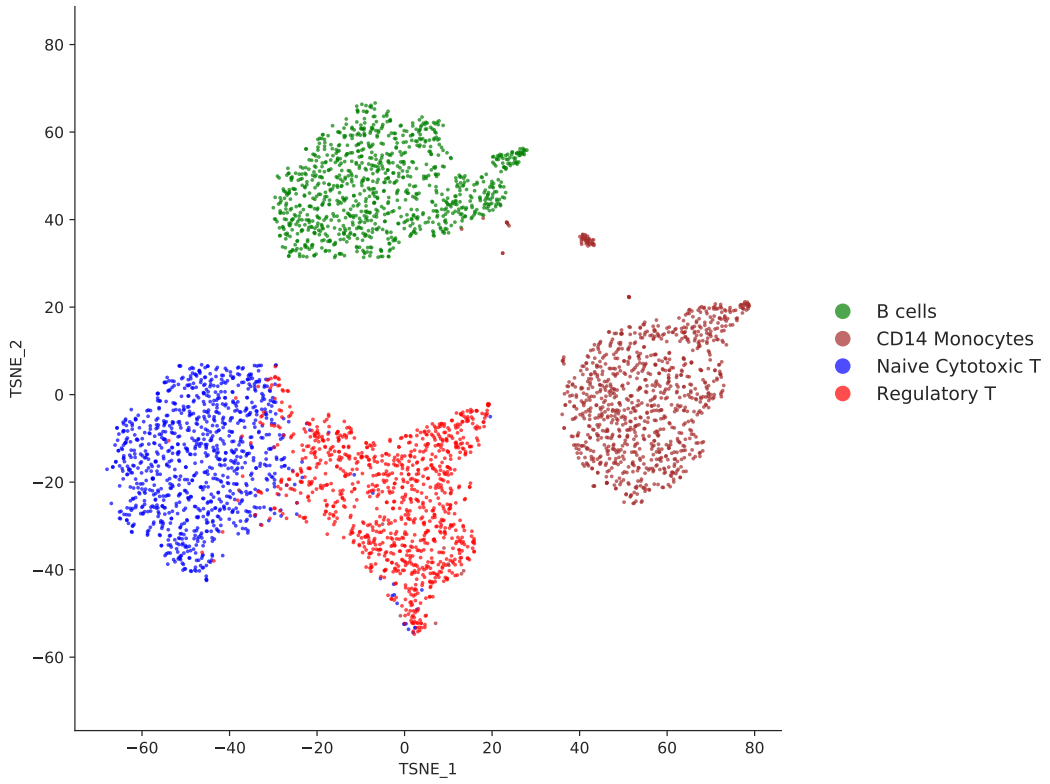

Figure S6: t-SNE visualization of the Zhengmix4eq dataset. Naive cytotoxic T cells and regulatory T cells partly overlap.

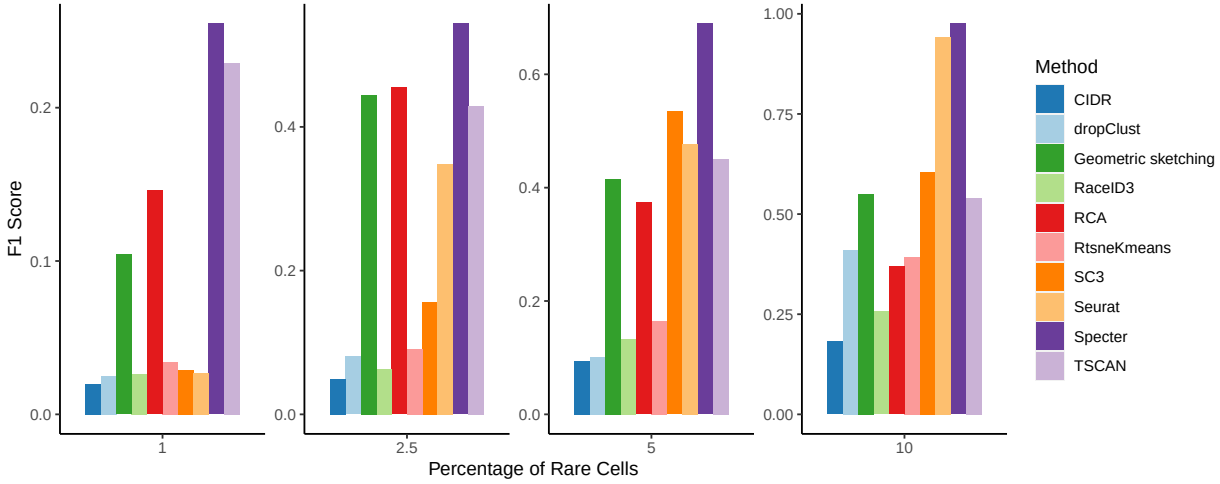

Figure S7: Sensitivity to rare population of overlapping cell types. Naive cytotoxic T cells and regulatory T cells taken from the Zhengmix4eq data set overlap in the t-SNE projection shown in Figure S6. We randomly downsampled naive cytotoxic and regulatory T cells to comprise 1%, 2.5%, 5%, and 10% of the total number of cells and repeated this experiment five times for each group. Average F_1 scores are shown over the 10 runs, with adjusted F_1 score ranges for each subsample size. For geometric sketching, the average F_1 score was taken over 10 random trials with a sketch size of 10% of the full data.

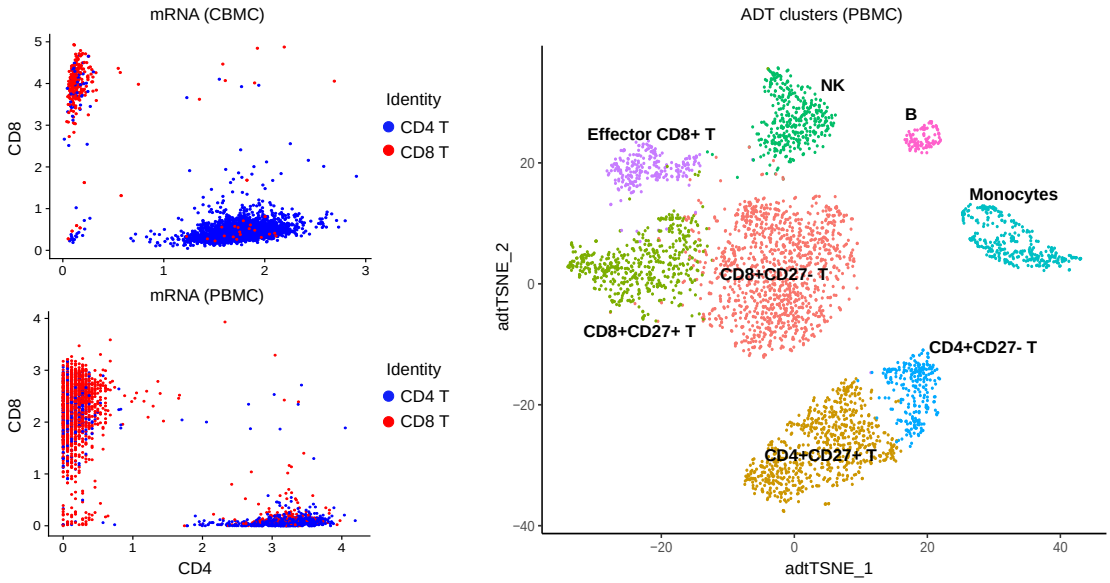

Figure S8: Seurat unimodal clustering. *left:* CBMCs (top) and PBMCs (bottom) with coordinates of protein expression (ADT) along CD4 and CD8 axis. Colors denote clusters computed by Seurat based on mRNA expression which contain a mix of CD4 T cells and CD8 T cells. *right:* t-SNE visualization of clusters identified by Seurat from protein expression (ADT) of PBM cells. $CD14^+$ and $FCGR3A^+$ monocytes cannot be discriminated, megakaryocytes are not detected.

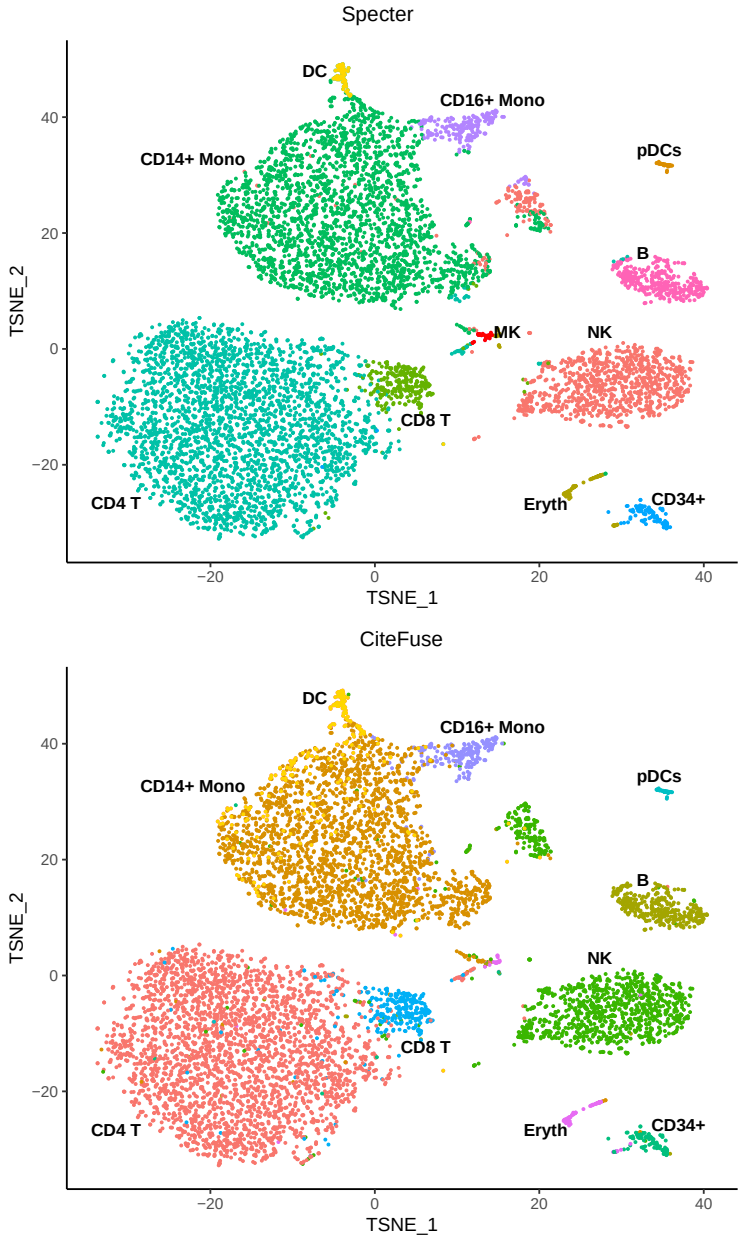

Figure S9: Comparison of multimodal clusterings of CBM cells as computed by Specter (top) and CiteFuse (bottom). Despite an overall high agreement between the two clusterings (ARI 0.94), only Specter detects a rare population of megakaryocytes (red).

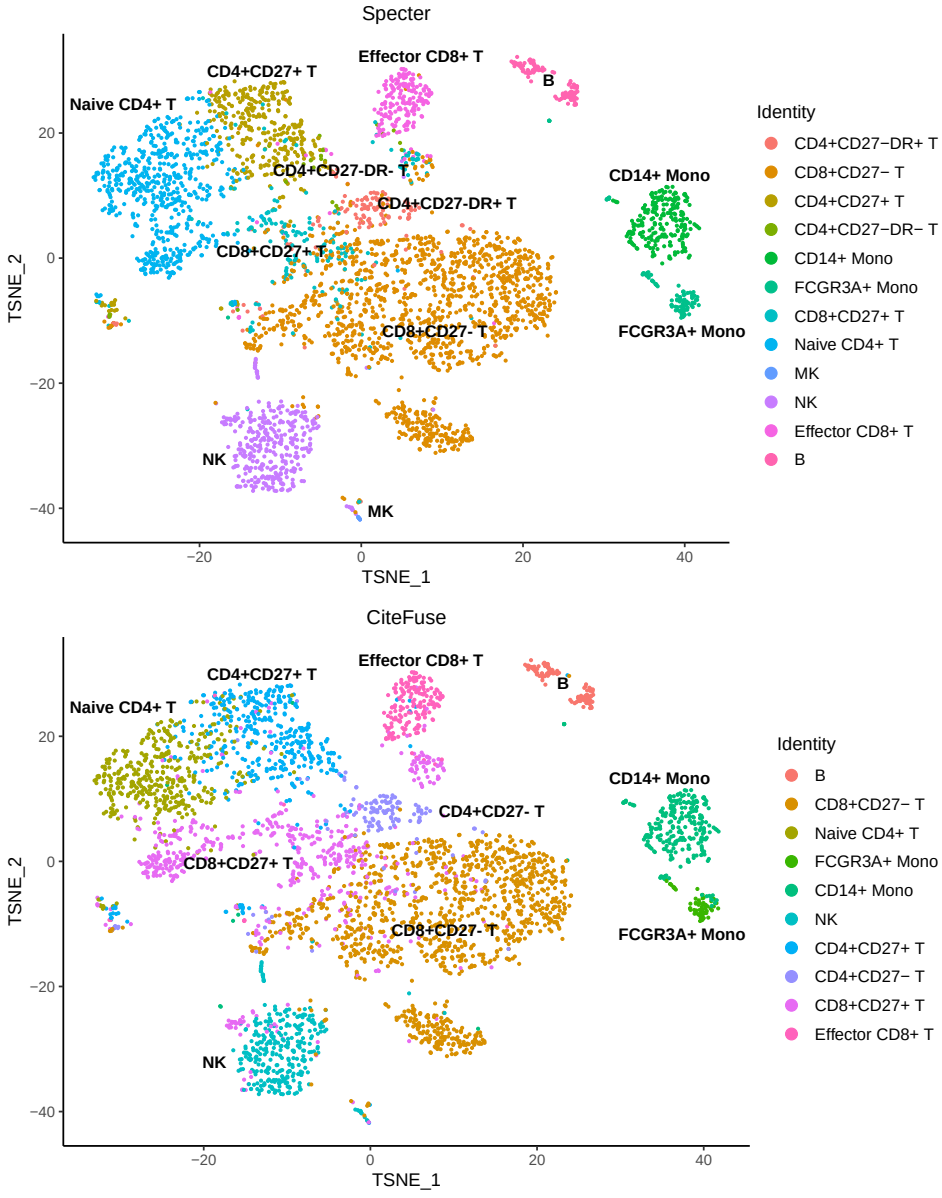

Figure S10: Comparison of multimodal clusterings of PBM cells as computed by Specter (top) and CiteFuse (bottom). Despite an overall high agreement between the two clusterings (ARI 0.86), only Specter detects a rare population of megakaryocytes and can discriminate between $CD27^{-}DR^{+}$ and $CD27^{-}DR^{-}$ subpopulations of $CD4^{+}$ memory T cells.

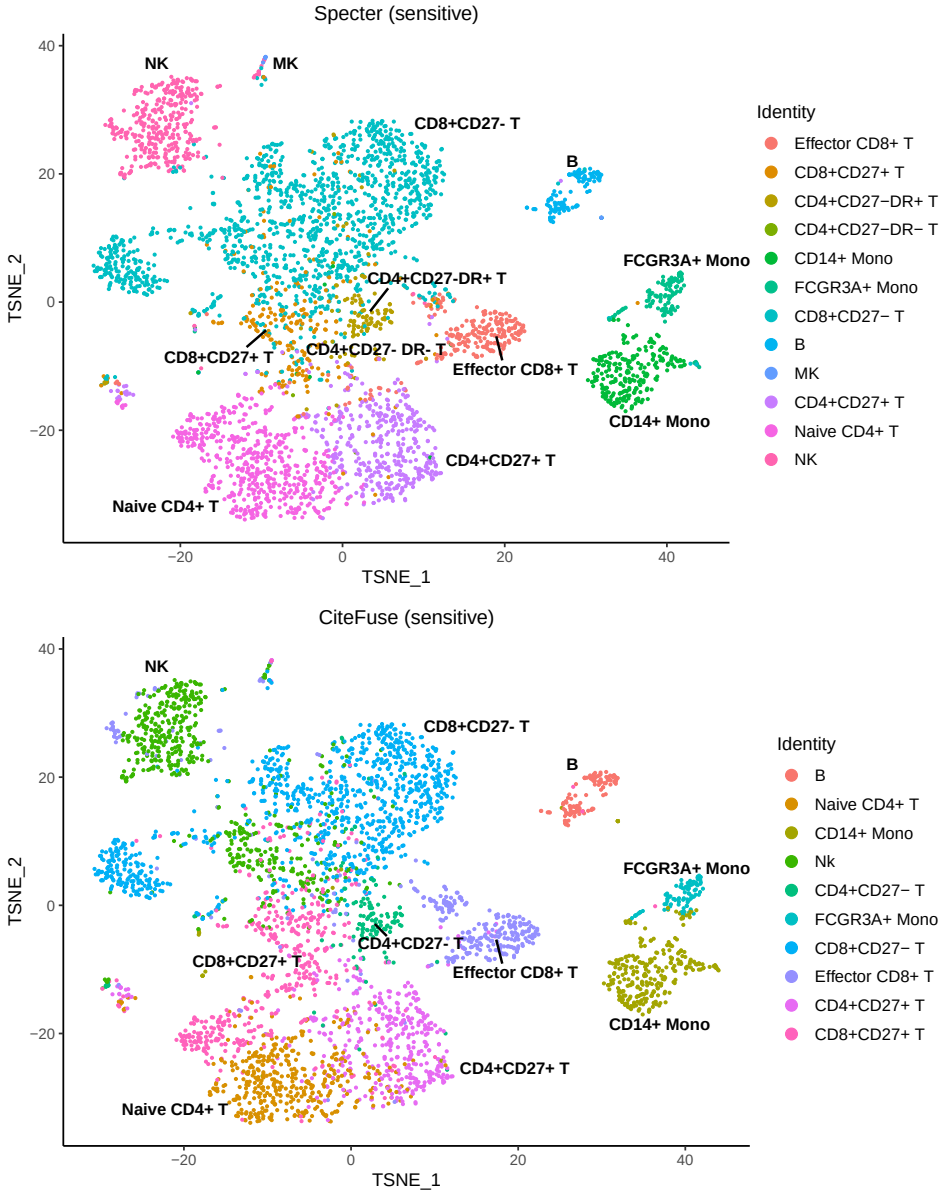

Figure S11: Comparison of multimodal clusterings of PBM cells. Here, Specter (top) and CiteFuse (bottom) use slightly more conservative parameters in the doublet removal (eps = 190, minPts = 10). Again, only Specter is able to discriminate between $CD27^{-}DR^{+}$ and $CD27^{-}DR^{-}$ subpopulations of $CD4^{+}$ memory T cells and detects a rare population of megakaryocytes.

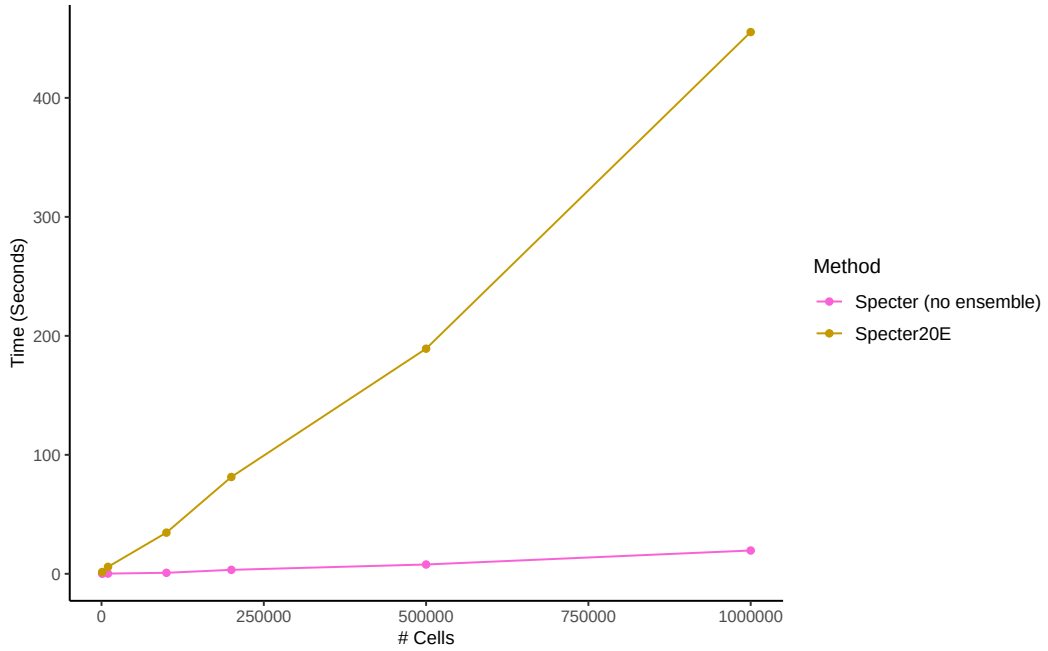

Figure S12: Linear-time complexity of Specter. CPU times in seconds (single threaded) are shown for the core algorithm of Specter (no ensemble) and Specter using a clustering ensemble of size 20. Different size data set were simulated using Splatter containing 1k, 10k, 100k, 200k, 500k, and 1 million cells.

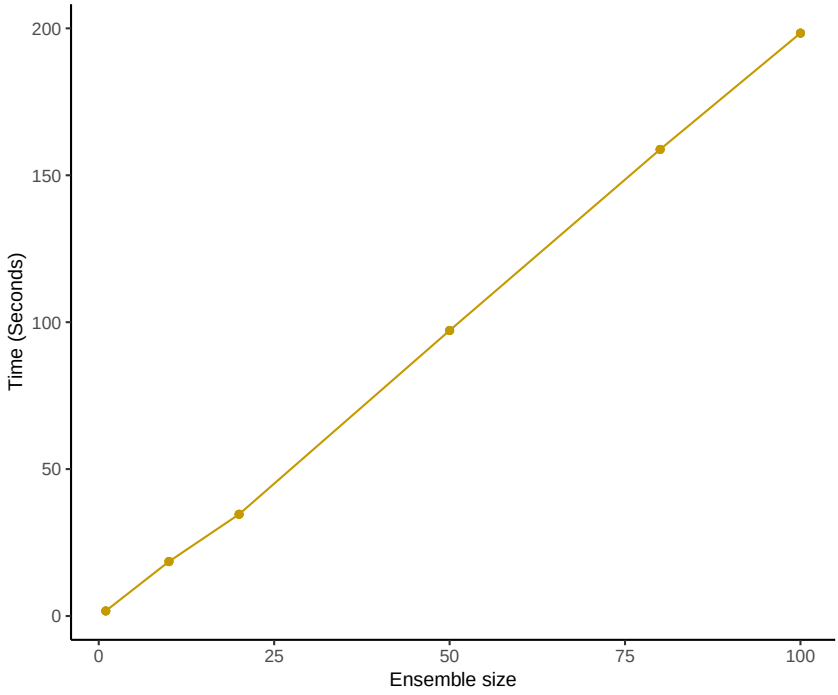

Figure S13: Linear increase in running time with number of ensemble members. CPU times in seconds (single threaded) are shown for Specter using an increasing number of ensemble members on a simulated data set containing 100,000 cells.

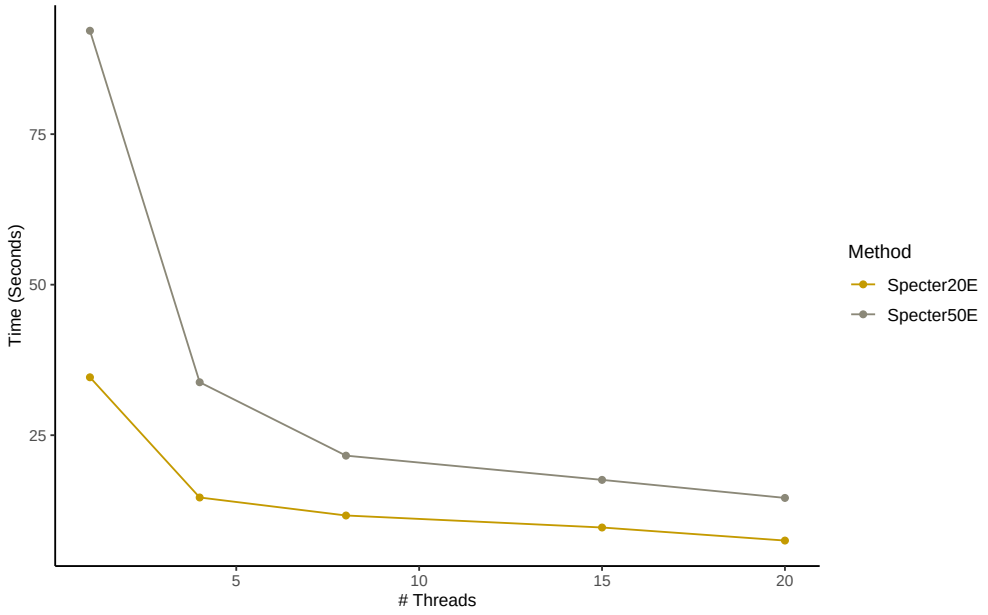

Figure S14: Specter speed-up with number of threads. CPU times in seconds are shown for Specter using an increasing number of threads on a simulated data set containing 100,000 cells. 20 or 50 clustering ensemble members were used.

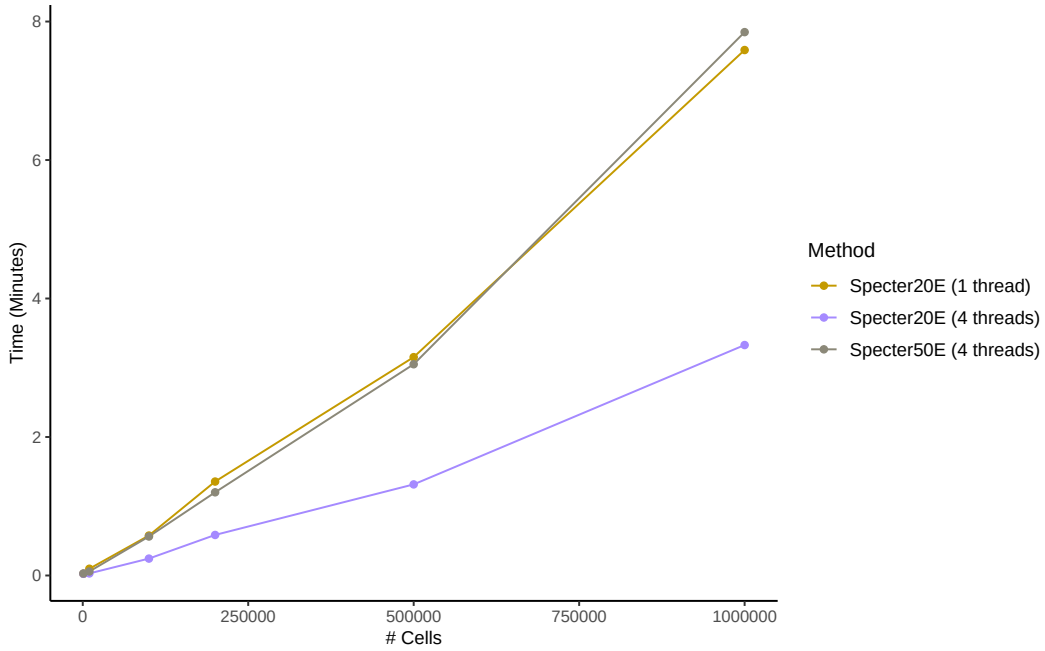

Figure S15: Increase in running time for fixed number of threads. CPU times in minutes are shown for Specter using 20 or 50 clustering ensemble members and 1 or 4 threads.

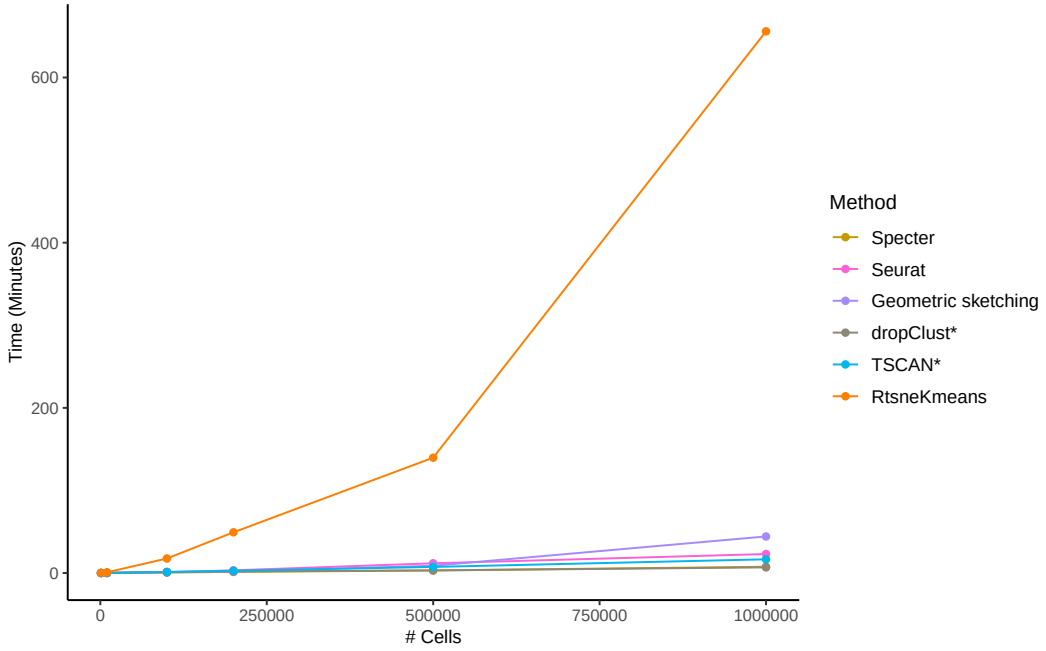

Figure S16: Runtime comparison between methods as a function of sample size. CPU times are shown in minutes on different numbers of cells sampled from a simulated data set containing 1 million cells. Seurat was run with a call to the more efficient SCANPY implementation of the Louvain clustering algorithm. *Running times exclude preprocessing for all methods except TSCAN and dropClust, whose implementation did not allow to isolate the core algorithm.

A.2 Supplemental Figures: Sphetcher

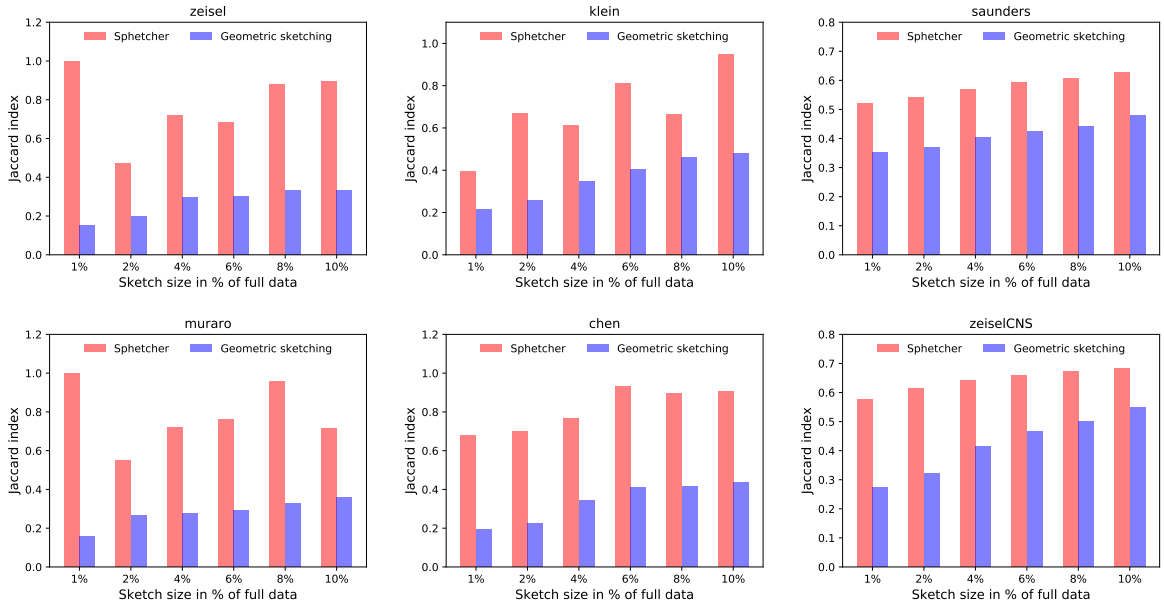

Figure S17: Comparison of Jaccard index. We compare the composition of the sketches computed in different random trials. The Jaccard index is computed for all pairs of random trials and the average is taken over all pairs for a given sketch size. The Jaccard index measures the similarity of two sketches by dividing the number of cells that they have in common by the total number of cells contained in either of the sketches. The Jaccard index ranges from 0 to 1, where 0 indicates that the two sketches have no cells in common, while 1 indicates identical sketches. Sphetcher returns highly similar sketches in different random trials, while the set of cells contained in geometric sketches can vary considerably between runs. In addition, these different geometric sketches differ in the quality of representation of the original transcriptomic space. Note that the similarity of geometric sketches returned in different runs slowly increases with larger sample size, since the algorithm has fewer choices to pick a cell in smaller boxes. In contrast, Sphetcher's random tie breaking between equal-sized sets does not depend on the sample size and thus provides highly stable sketches even for small numbers of cells.

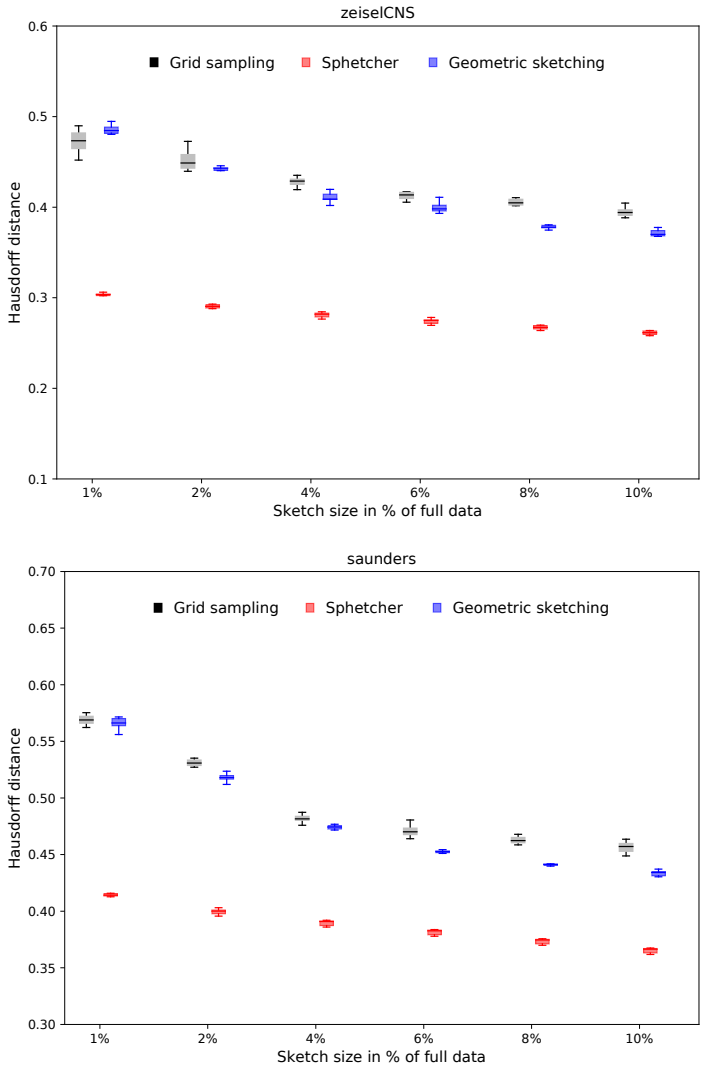

Figure S18: Comparison of Hausdorff distances. The naïve grid sampling strategy alone, which is part of our hybrid alternative for very large datasets, achieves competitive Hausdorff distances to geometric sketching on datasets zeiselCNS and saunders, especially for small sketch sizes. For each sketch size, the results of 10 random trials are shown.

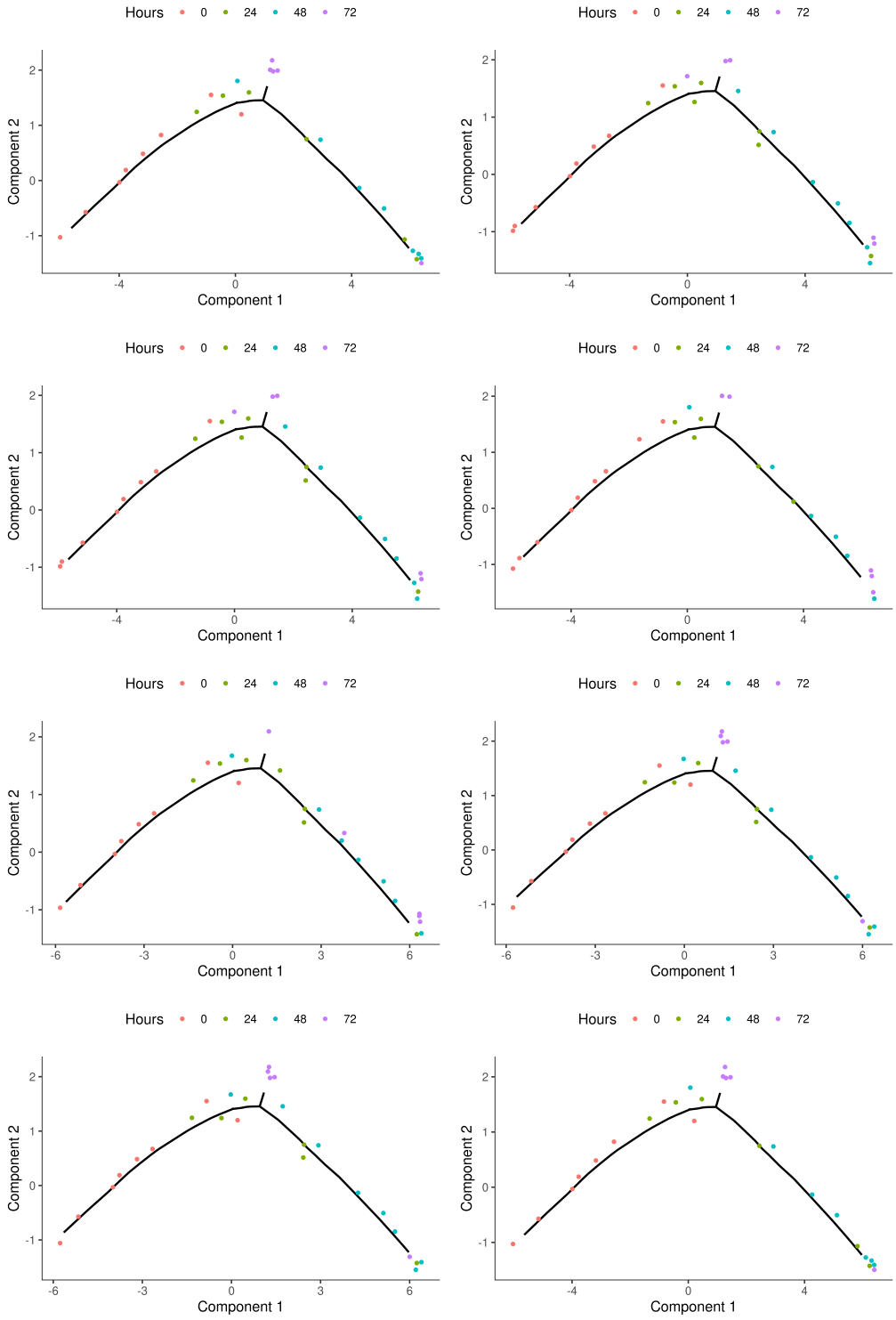

Figure S19: HSMM differentiation trajectories reconstructed by Monocole 2 from Sphetcher's sketch with fairness constraints. In 8 trials, Sphetcher did not include 'outlier' cells when its fairness model requires to include at least 4 cells from each time point. For outlier cells inferred pseudotime and actual collection time disagree. At the same time, cells collected at time point 72 in the final state are consistently retained.

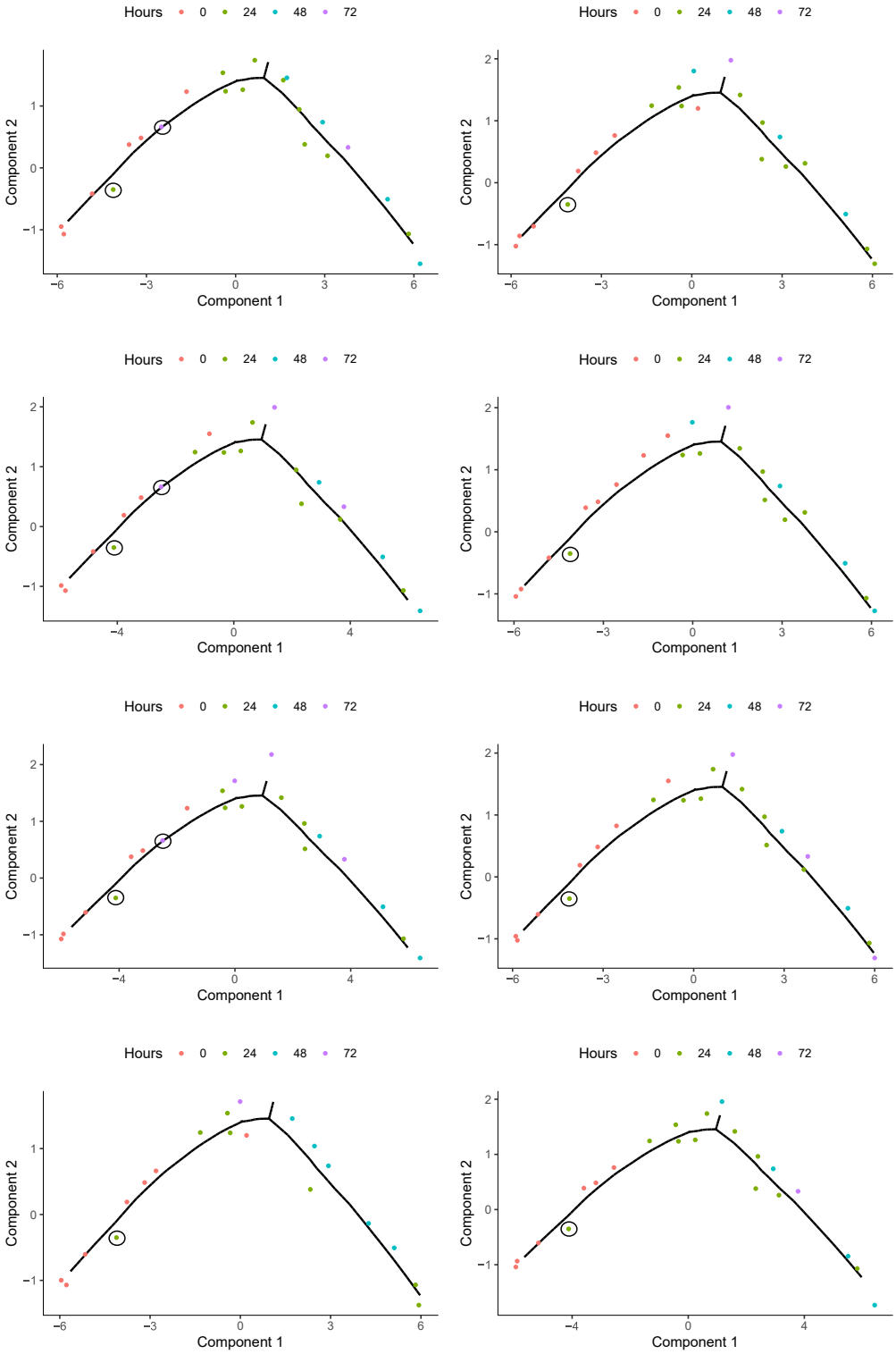

Figure S20: HSMM differentiation trajectories reconstructed by Monocole 2 from geometric sketches. In each of 8 trials, geometric sketches included outlier cells (black circles) for which inferred pseudotime and actual collection time disagree. At the same time, in only a single case a cell in final state collected at time point 72 is retained.

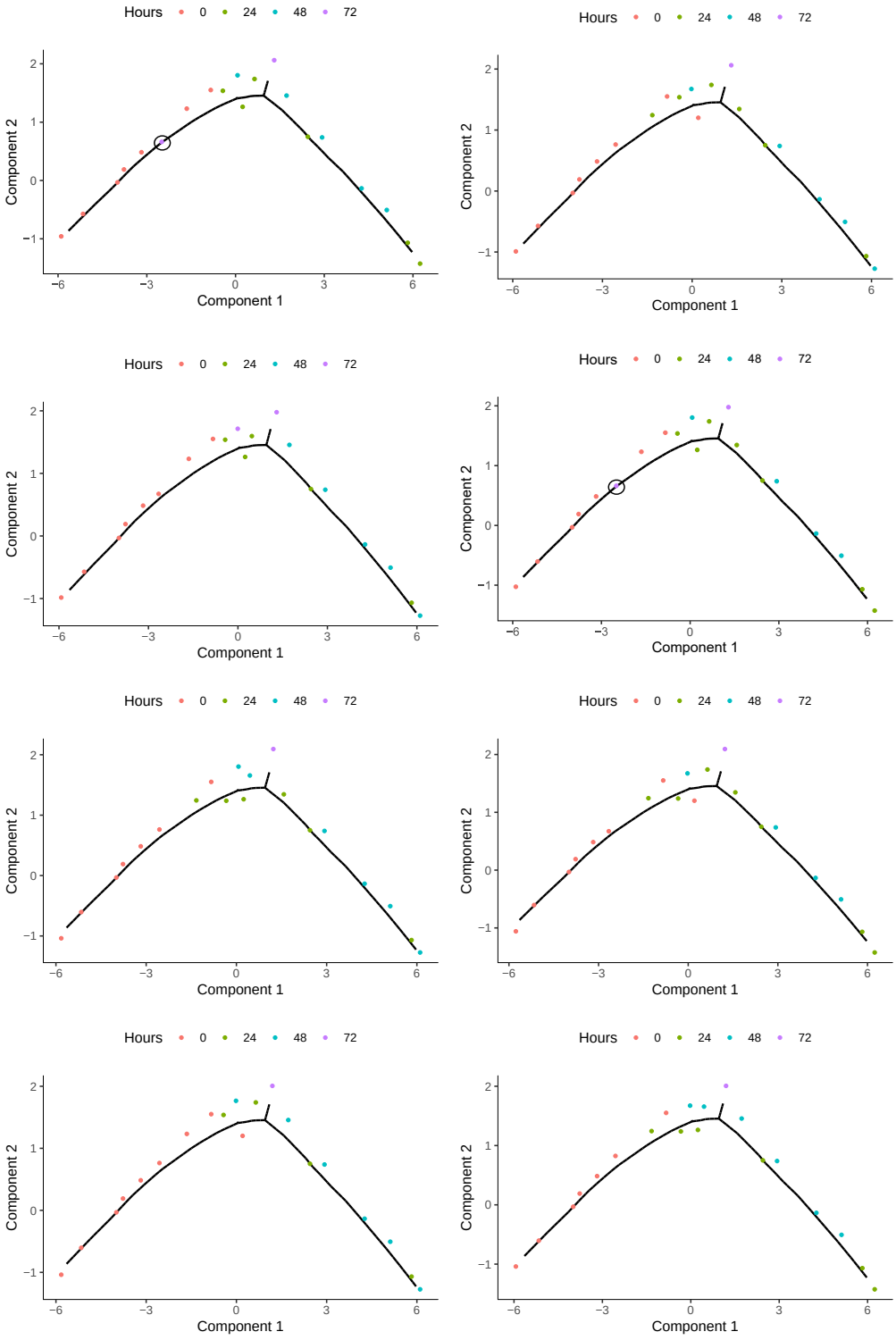

Figure S21: HSMM differentiation trajectories reconstructed by Monocole 2 from Sphetcher's sketch without fairness constraints. In 2 out of 8 trials, spherical sketches included outlier cells (black circles) for which inferred pseudotime and actual collection time disagree. At the same time, cells in final state collected at time point 72 are lost in each trial.

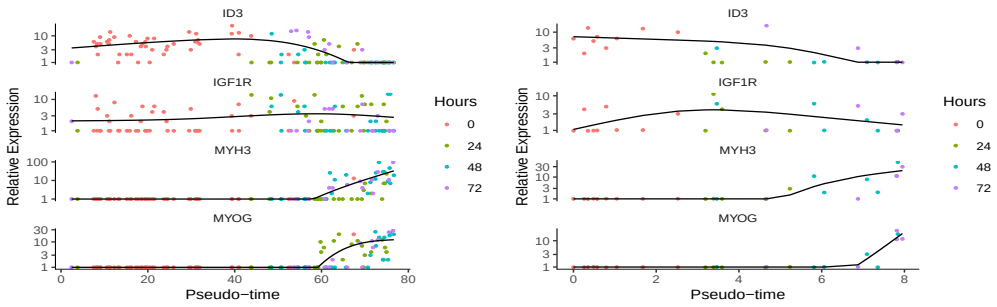

Figure S22: Gene expression dynamics. Expression dynamics along pseudotime were computed by Monocle 2 from full data (left) and from the sketch produced by Sphetcher with fairness constraints (right) for genes ID3, IGF1R, MYH3, and MYOG.

A.3 Supplemental Figures: Jvis

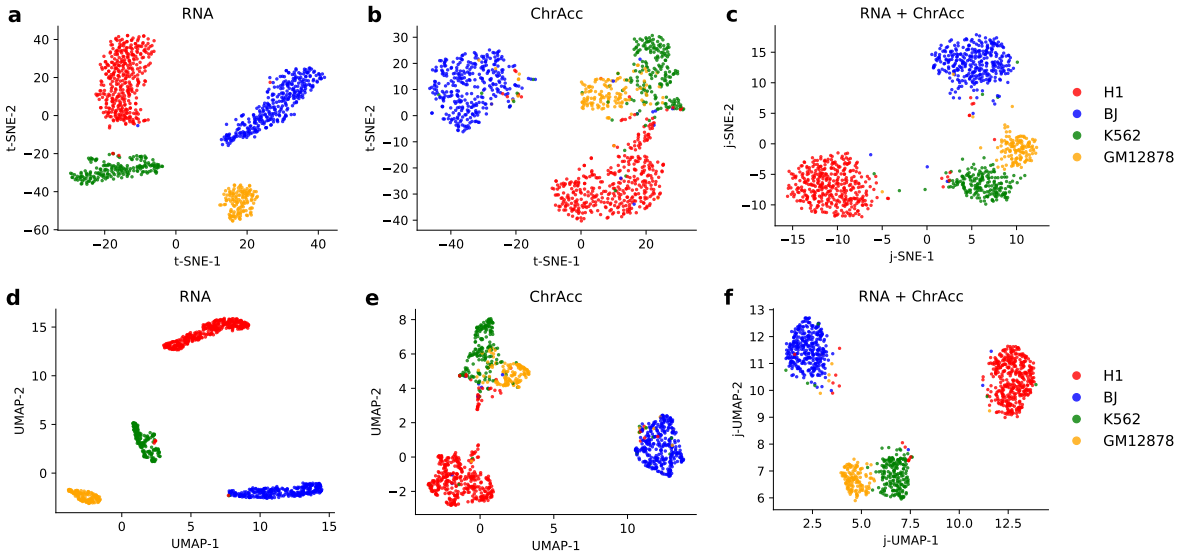

Figure S23: Unimodal and multimodal visualization of SNARE-seq measurements. Accessible chromatin (ChrAcc) and gene expression was measured simultaneously in single cell from human cell lines BJ, H1, K562, and GM12878. *First row:* Comparison of t-SNE and j-SNE. Conventional t-SNE embedding of measurements of (**a**) RNA or (**b**) accessible chromatin. (**c**) Joint embedding of both modalities (RNA and chromatin accessibility) by j-SNE. *Second row:* Comparison of UMAP and j-UMAP. Conventional UMAP of (**d**) RNA or (**e**) accessible chromatin. (**f**) Joint embedding of both modalities by j-UMAP.

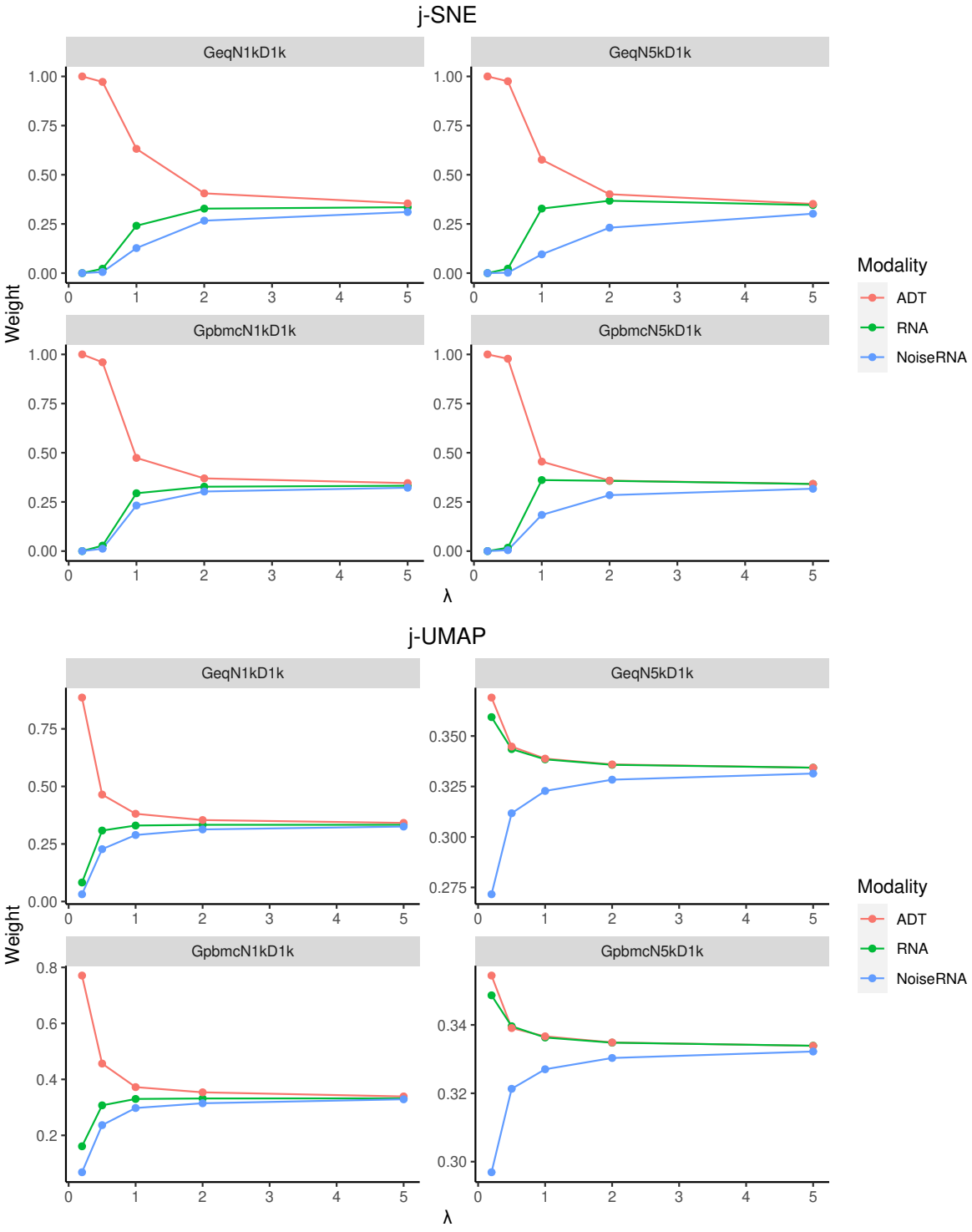

Figure S24: Weights of modalities ADT, RNA, and NoiseRNA computed in j-SNE (top) and j-UMAP (bottom) as a function of regularization coefficient λ ranging from $0.2 - 5$. The noise level in the four simulated data sets is held constant at 0.4.

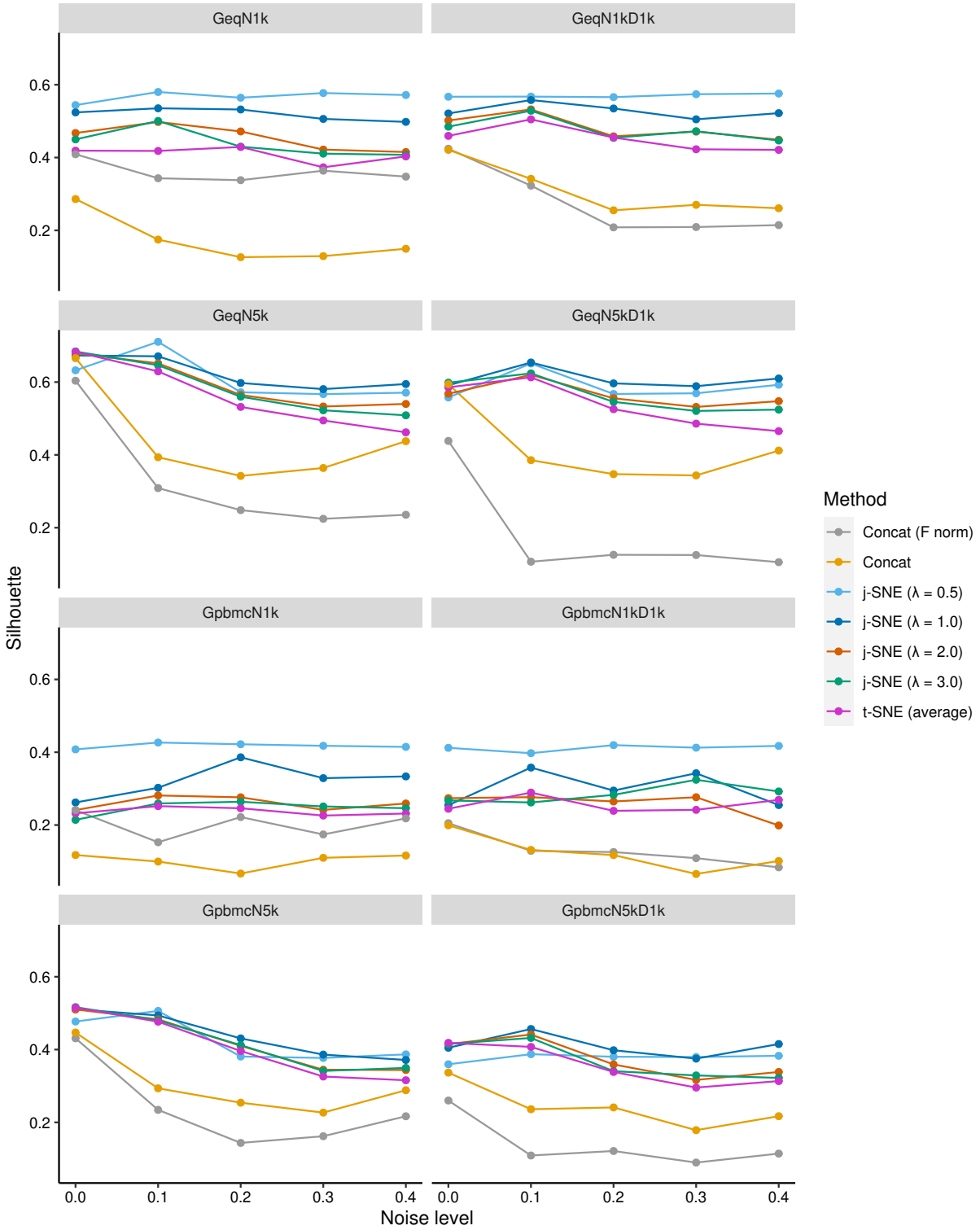

Figure S25: Silhouette scores of embeddings computed by j-SNE and alternative methods on eight simulated data sets. Values are shown as a function of noise for different regularization coefficient λ used in j-SNE. Conventional t-SNE is run for uniform weights assigned to each modality ($\alpha_i = 1/3$) (t-SNE (average)), or on concatenated modalities (Concat) that are optionally normalized by the Frobenius norm (Concat (F norm)).

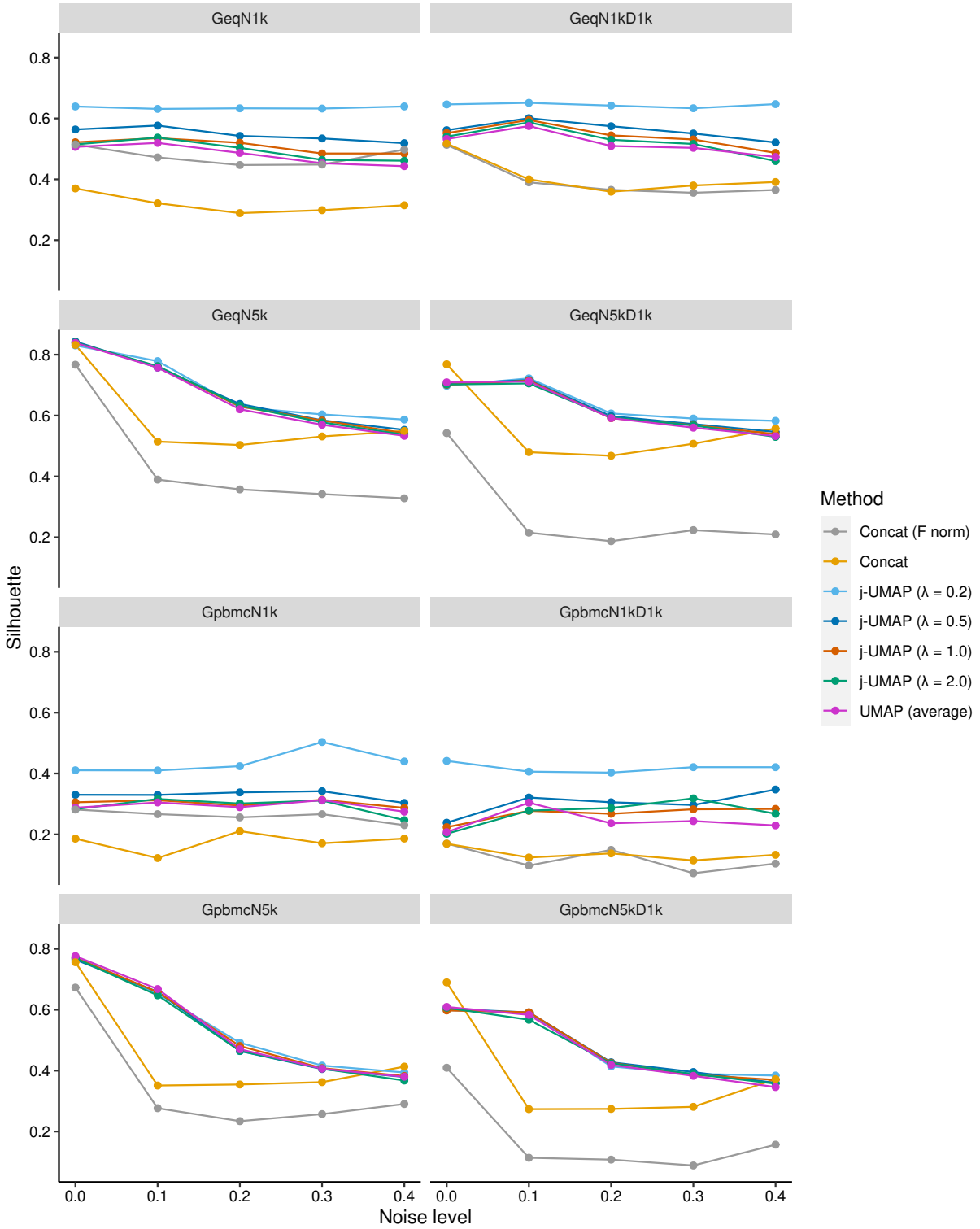

Figure S26: Silhouette scores of embeddings computed by j-UMAP and alternative methods on eight simulated data sets. Values are shown as a function of noise for different regularization coefficient λ used in j-UMAP. Conventional UMAP is run for uniform weights assigned to each modality ($\alpha_i = 1/3$) (UMAP (average)), or on concatenated modalities (Concat) that are optionally normalized by the Frobenius norm (Concat (F norm)).

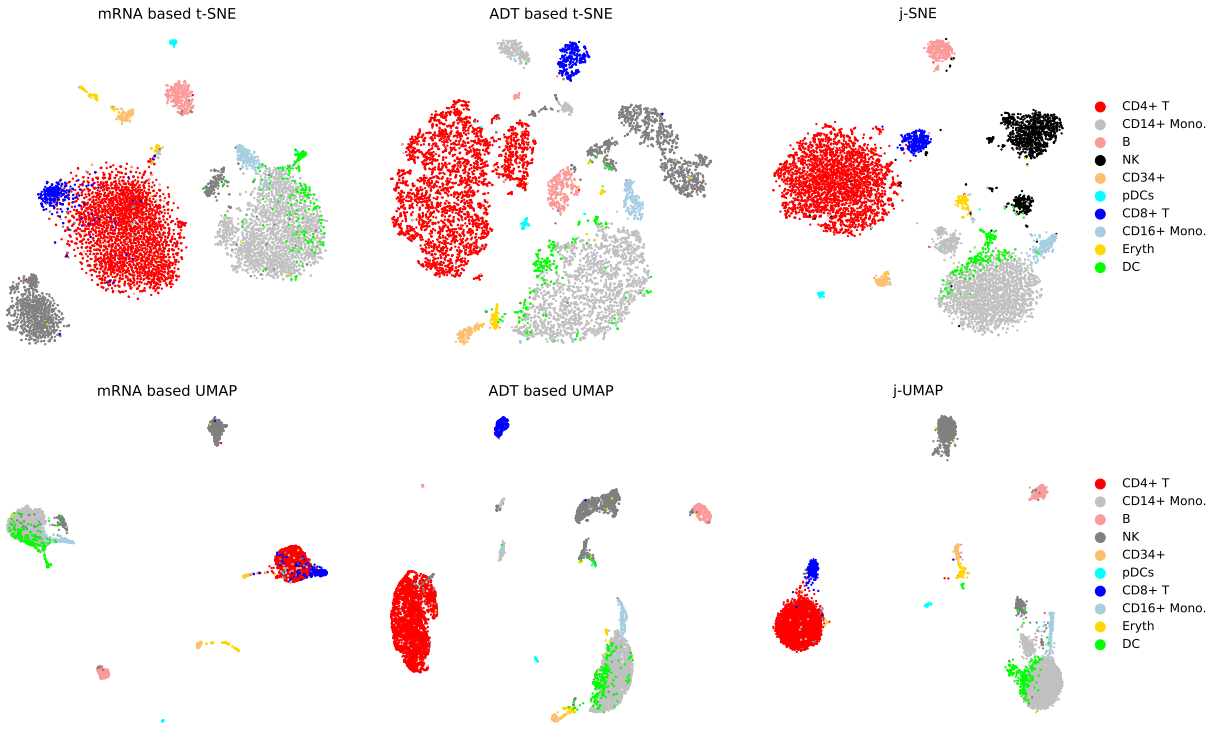

Figure S27: t-SNE/j-SNE (top row) and UMAP/j-UMAP (bottom row) visualizations of CBM cells. Cluster labels were identified by CiteFuse. Embeddings were computed from mRNA measurements alone (left), protein expression (ADT) alone (middle), or jointly from both (right).

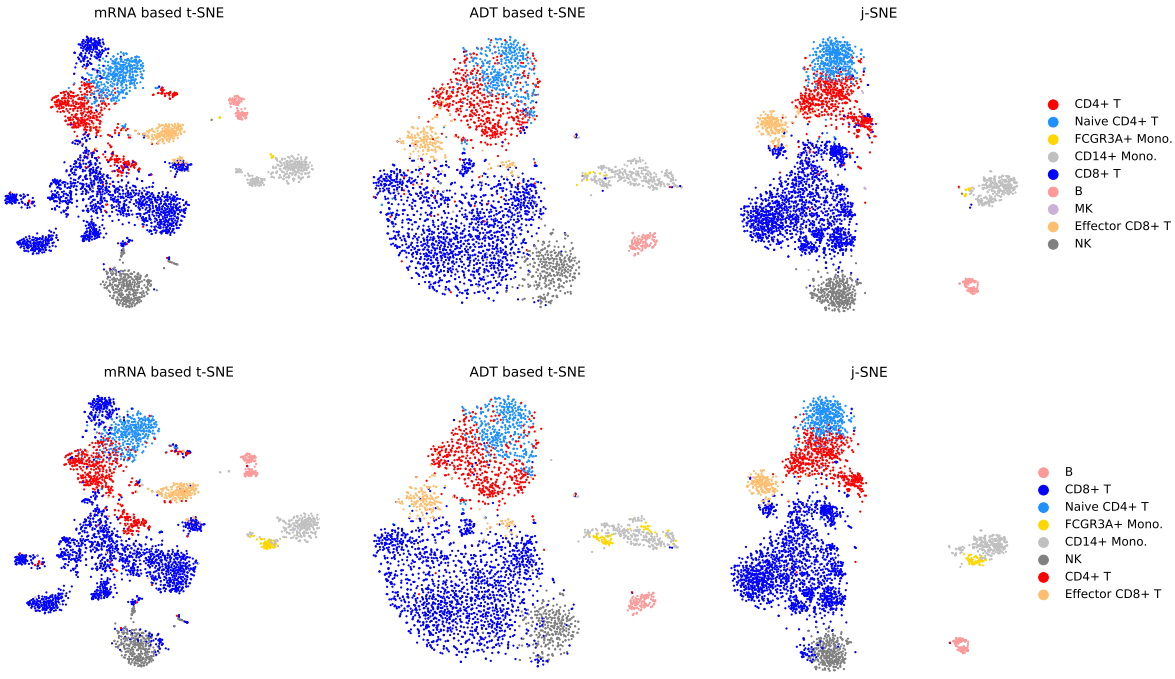

Figure S28: t-SNE/j-SNE visualizations of PBM cells. Cluster labels were identified by Specter (top row) or CiteFuse (bottom row). Embeddings were computed from RNA measurements alone (left), protein expression (ADT) alone (middle), or jointly from both (right).

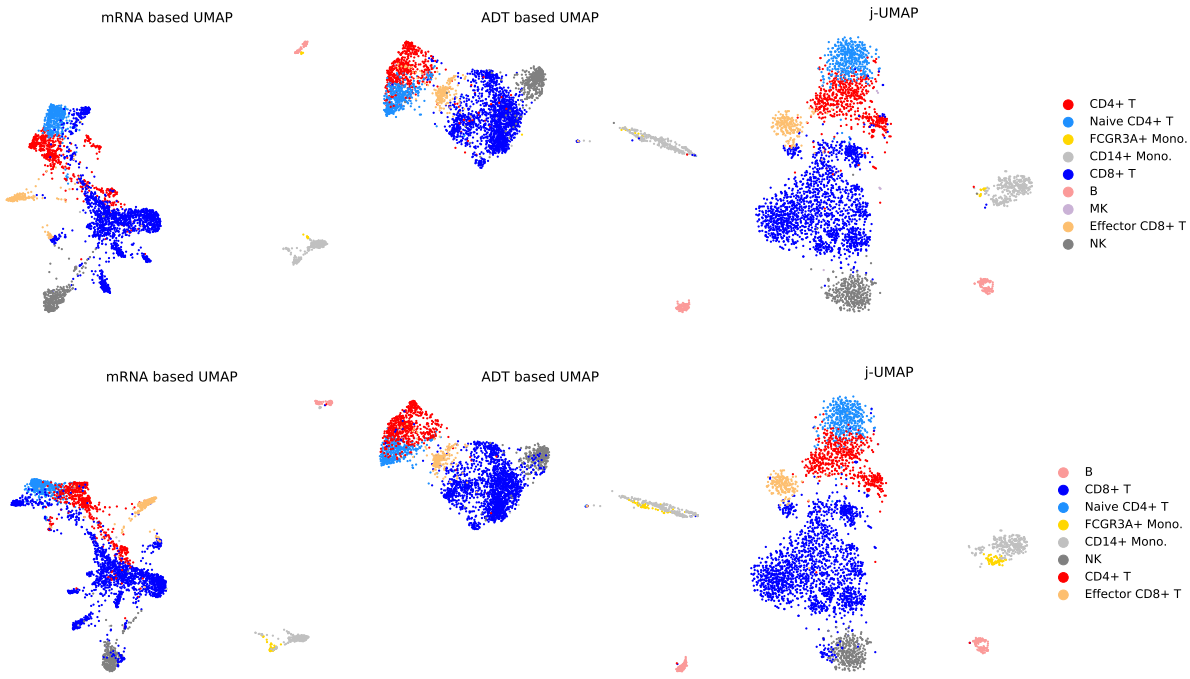

Figure S29: UMAP/j-UMAP visualization of PBM cells. Cluster labels were identified by Specter (top row) or CiteFuse (bottom row). Embeddings were computed from RNA measurements alone (left), protein expression (ADT) alone (middle), or jointly from both (right).

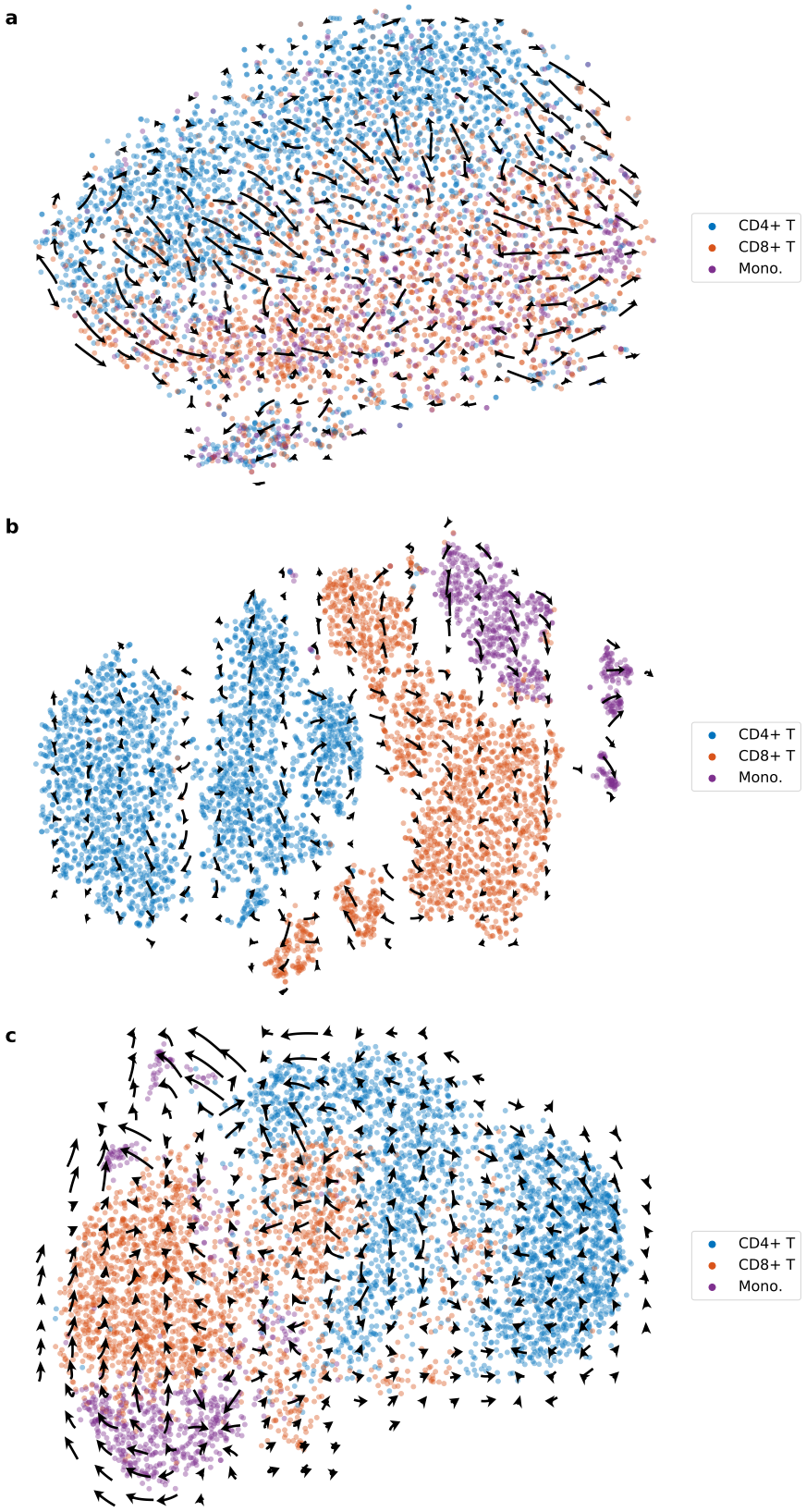

Figure S30: Protein acceleration in ECCITE-seq data set CTCL projected into transcriptome-based t-SNE (a), j-SNE (b), and j-UMAP (c) embeddings.

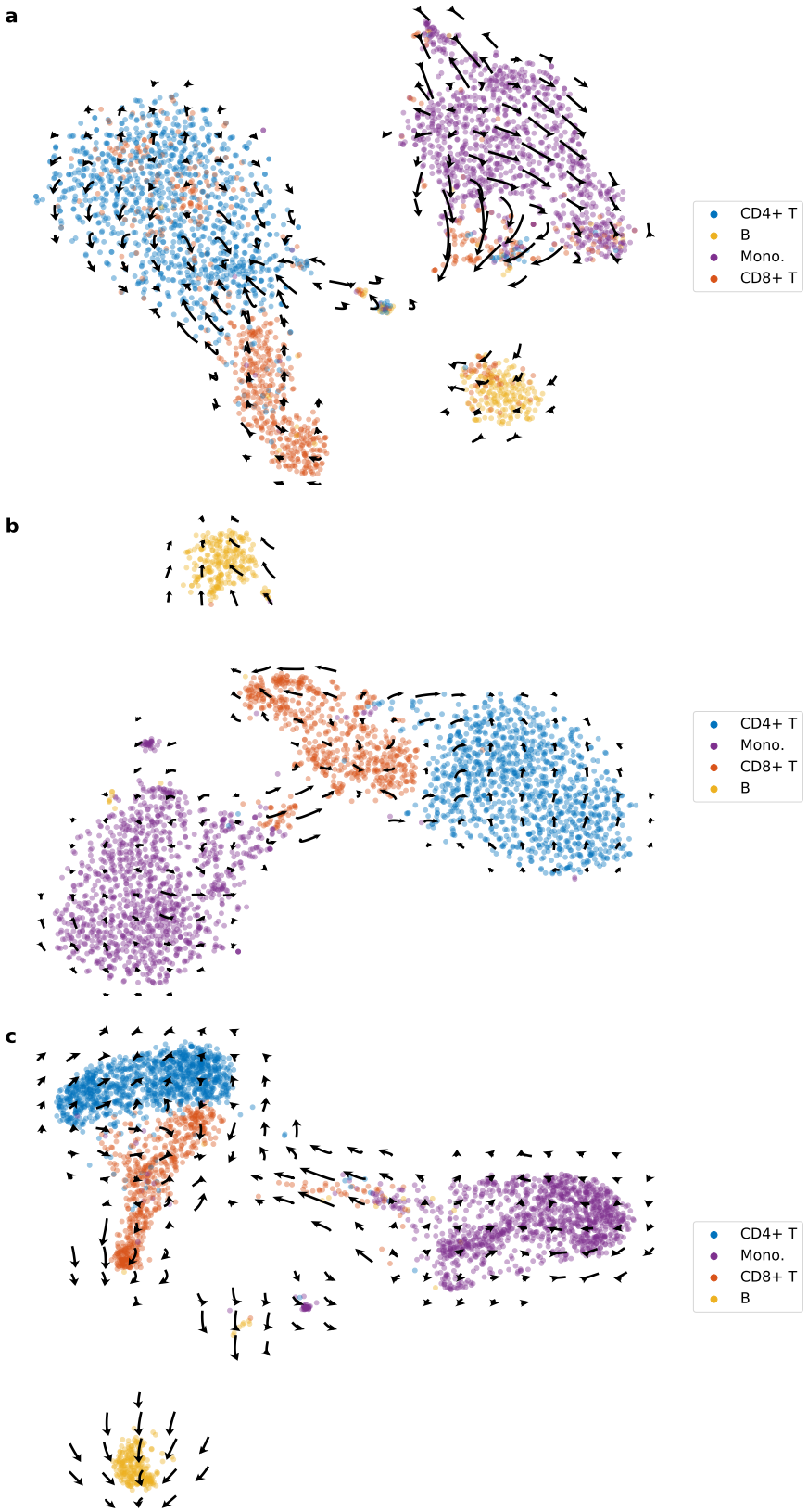

Figure S31: Protein acceleration in REAP-seq data set projected into transcriptome-based t-SNE (a), j-SNE (b), and j-UMAP (c) embeddings.

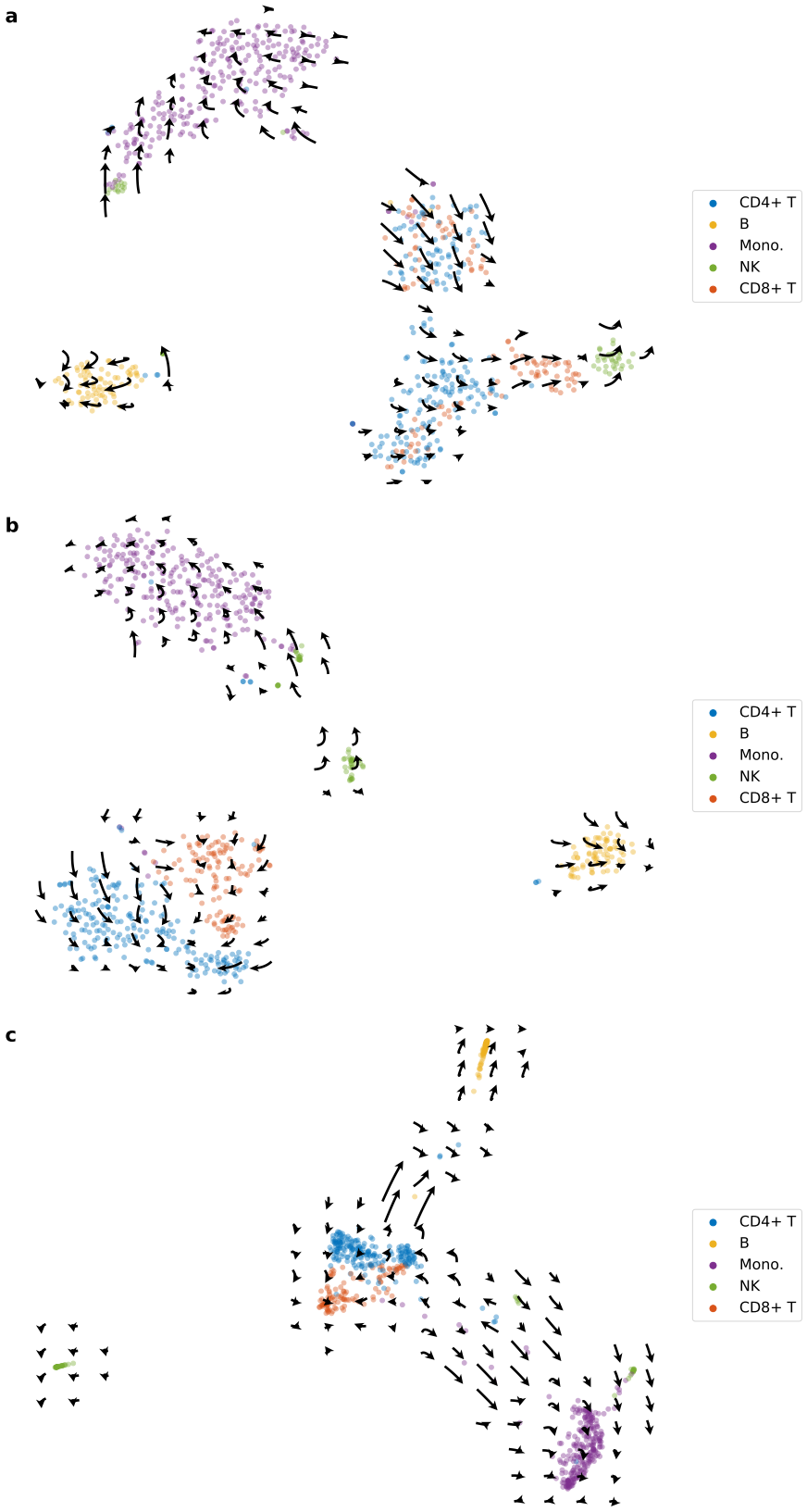

Figure S32: Protein acceleration in 10X 1k data set projected into transcriptome-based t-SNE (a), j-SNE (b), and j-UMAP (c) embeddings.

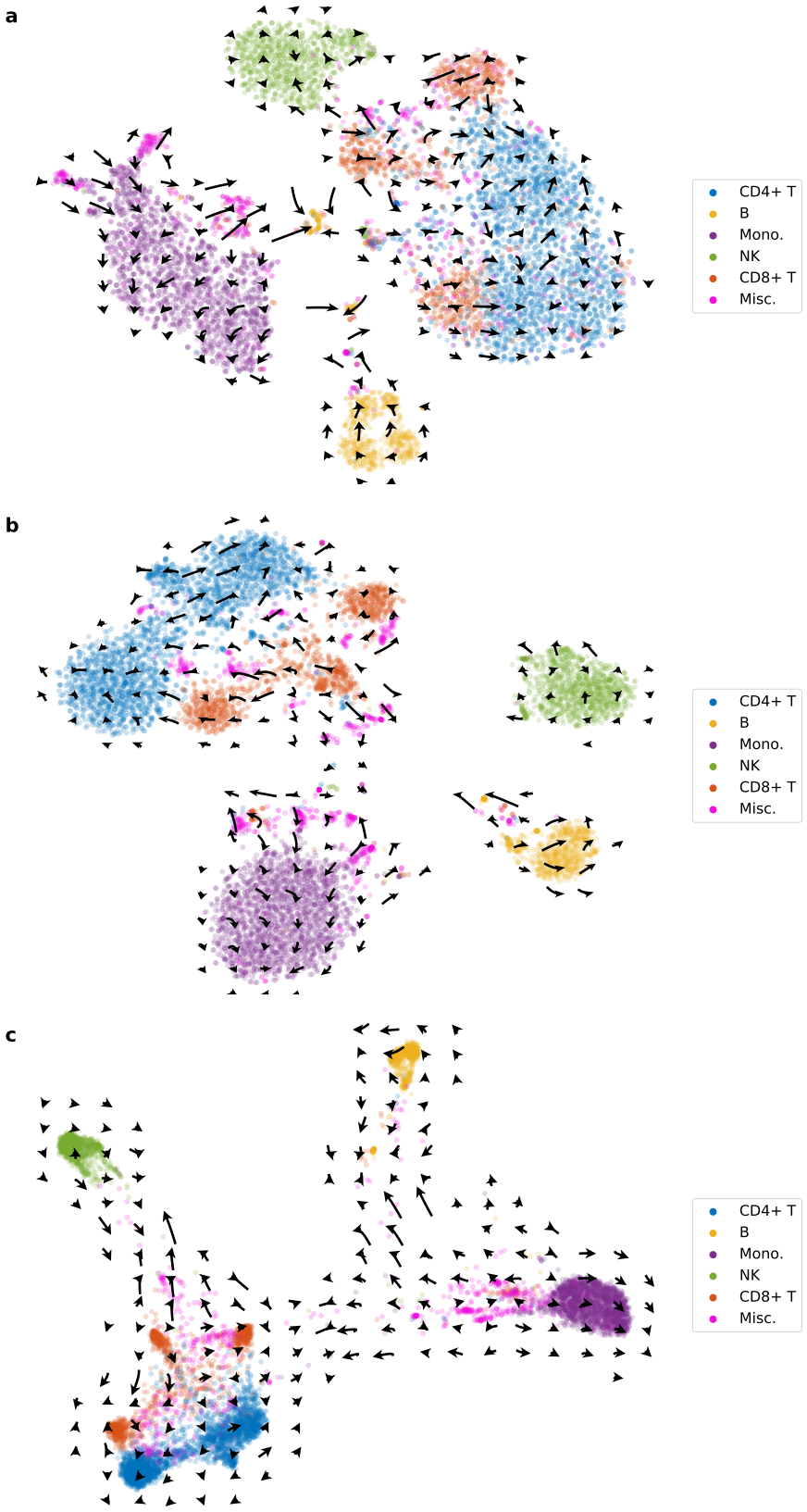

Figure S33: Protein acceleration in 10X 10k data set projected into transcriptome-based t-SNE (a), j-SNE (b), and j-UMAP (c) embeddings.

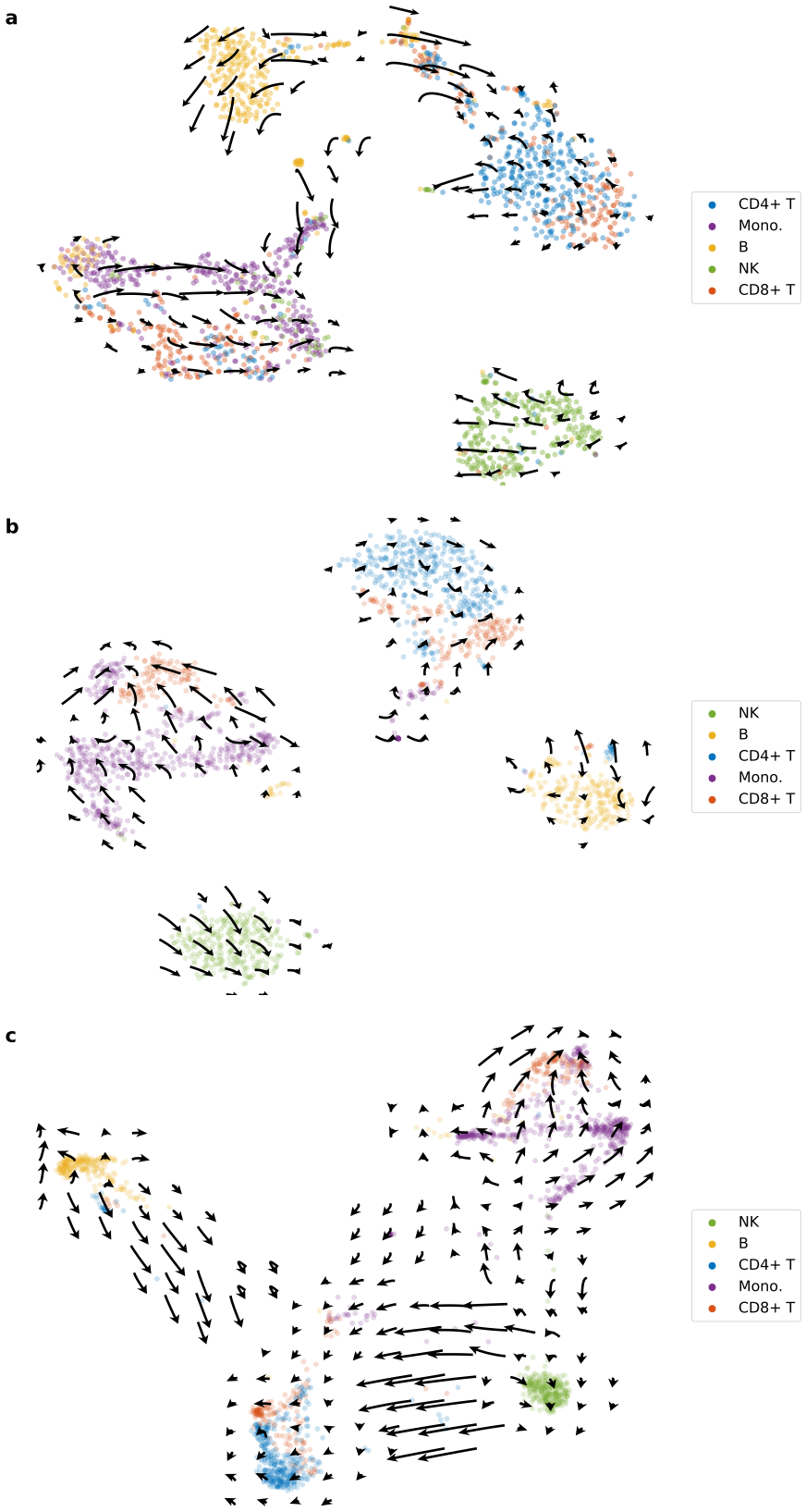

Figure S34: Protein acceleration in CITE-seq data set projected into transcriptome-based t-SNE (a), j-SNE (b), and j-UMAP (c) embeddings.

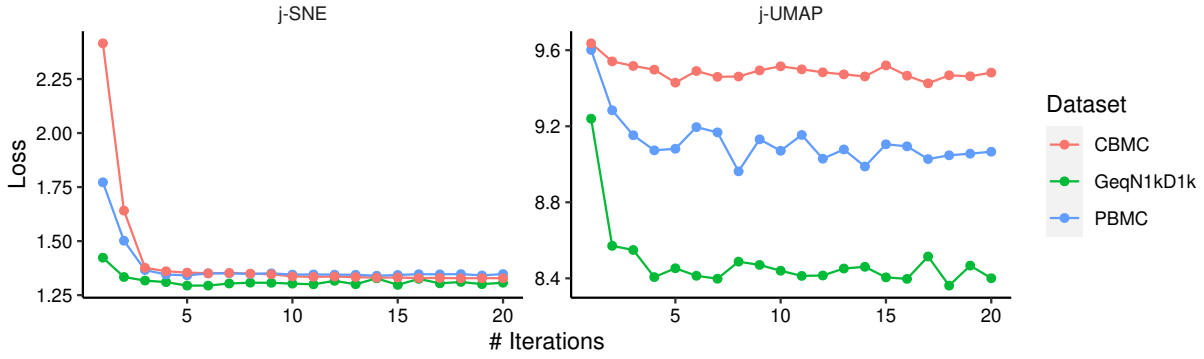

Figure S35: Change in loss function across iterations in j-SNE and j-UMAP on 2 real data set and one simulated data set.

Appendix B

Supplementary Tables

Table S1: Overview of the simulated data sets used in this study. Names listed in the left-most column are used throughout the text. Data sets were simulated using Splatter (Zappia et al., 2017) and vary in number of cells (#Cells), number of genes (#Genes), number of clusters (k), the probability with which a given gene is differentially expressed in one of the cell types (marker genes), and the relative abundance of cell types that were either equal, unequal, or based on cell type abundances among peripheral blood mononuclear cells (PBMCs) in healthy individuals.

Name	# Cells (N)	# Genes (D)	k	Probabilities of gene DE	Relative abundances (G)
DE1GeqN1k	1,000	1,000	5	(0.01, 0.01 0.01, 0.01, 0.01)	(0.2, 0.2, 0.2, 0.2, 0.2)
DE1GeqN2k	2,000	1,000	5		
DE1GeqN5k	5,000	1,000	5		
DEneqGneqN1k	1,000	1,000	5	(0.01, 0.01 0.02, 0.02, 0.05)	(0.01, 0.05, 0.14, 0.3, 0.5)
DEneqGneqN2k	2,000	1,000	5		
DEneqGneqN5k	5,000	1,000	5		
DEneqGneqN1kD10k	1,000	10,000	5	(0.01, 0.01 0.02, 0.02, 0.05)	(0.01, 0.05, 0.14, 0.3, 0.5)
DEneqGneqN2kD10k	2,000	10,000	5		
DEneqGneqN5kD10k	5,000	10,000	5		
DE1GneqN1k	1,000	1,000	5	(0.01, 0.01 0.01, 0.01, 0.01)	(0.01, 0.05, 0.14, 0.3, 0.5)
DE1GneqN2k	2,000	1,000	5		
DE1GneqN5k	5,000	1,000	5		
DE1GneqN1kD10k	1,000	10,000	5	(0.01, 0.01 0.01, 0.01, 0.01)	(0.01, 0.05, 0.14, 0.3, 0.5)
DE1GneqN2kD10k	2,000	10,000	5		
DE1GneqN5kD10k	5,000	10,000	5		
DE2GneqN1k	1,000	1,000	5	(0.02, 0.02 0.02, 0.02, 0.02)	(0.01, 0.05, 0.14, 0.3, 0.5)
DE2GneqN2k	2,000	1,000	5		
DE2GneqN5k	5,000	1,000	5		
DE5GneqN1k	1,000	1,000	5	(0.05, 0.05 0.05, 0.05, 0.05)	(0.01, 0.05, 0.14, 0.3, 0.5)
DE5GneqN2k	2,000	1,000	5		
DE5GneqN5k	5,000	1,000	5		
DE1GpbmcN1k	1,000	1,000	5	(0.01, 0.01 0.01, 0.01, 0.01)	PBMCs: DC: 0.02, NK: 0.2, B: 0.1 Mono: 0.08, T: 0.6
DE1GpbmcN2k	2,000	1,000	5		
DE1GpbmcN5k	5,000	1,000	5		
RareCellExp1	4,000	1,000	2	(0.01, 0.01)	(0.5, 0.5)
RareCellExp2	10,000	1,000	2	(0.01, 0.01)	(0.9, 0.1)

Table S2: Comparison of running times in minutes on simulated data. Data sets of different size were simulated using Splatter. *Running times exclude preprocessing for all methods except TSCAN and dropClust, whose implementation did not allow to isolate the core algorithm. Specter used 20 ensemble members and was run with a single thread (as all other methods). The last column (Specter+Pre) shows the total running time of Specter and all its preprocessing steps, including log-transformation, selection of highly variable genes (500), and PCA.

#Cells	Specter	Seurat	dropClust*	Geosketch	RtsneKmeans	TSCAN*	Specter+Pre
1k	0.02	0.04	0.04	0.10	0.14	0.06	0.02
10k	0.1	0.15	0.24	0.02	0.88	0.20	0.1
100k	0.58	1.00	1.01	1.38	17.61	1.23	0.61
200k	1.36	3.27	1.89	1.75	49.31	2.79	1.40
500k	3.15	11.80	3.14	8.81	139.69	7.39	3.25
1m	7.59	23.00	6.83	44.29	655.95	16.61	7.77

Table S3: Comparison of running times on three largest real data sets. Running times of Specter, Seurat, dropClust, the geometric sketching (Gsketch) based Louvain clustering, TSCAN, and RtseKmeans are reported in minutes (rounded) on the 3 largest real data sets used in this study. *Running times exclude preprocessing for all methods except TSCAN and dropClust, whose implementation did not allow to isolate the core algorithm. Specter used 50 ensemble members and was run with 20 threads. The last column (Specter+Pre) shows the total running time of Specter and all its preprocessing steps, including log-transformation, selection of highly variable genes (2000), and PCA.

Data set	#Cells	Specter	Seurat	dropClust*	Gsketch	TSCAN*	RtsneKmeans	Specter+Pre
CNS	464,713	1	11	2	7	3	89	3
saunders	665,385	2	18	3	19	8	193	4
trapnell	2,026,641	15	79	12	400	100	1225	23

Bibliography

Sara Aibar, Carmen Bravo González-Blas, Thomas Moerman, Vân Anh Huynh-Thu, Hana Imrichova, Gert Hulselmans, Florian Rambow, Jean-Christophe Marine, Pierre Geurts, Jan Aerts, Joost van den Oord, Zeynep Kalender Atak, Jasper Wouters, and Stein Aerts. Scenic: single-cell regulatory network inference and clustering. *Nature Methods*, 14(11): 1083–1086, 2017. doi: 10.1038/nmeth.4463.

Ayelet Alpert, Lindsay S Moore, Tania Dubovik, and Shai S Shen-Orr. Alignment of single-cell trajectories to compare cellular expression dynamics. *Nature Methods*, 15(4):267–270, 2018. doi: 10.1038/nmeth.4628.

Georg Anegg, Haris Angelidakis, Adam Kurpisz, and Rico Zenklusen. A technique for obtaining true approximations for k-center with covering constraints. In *Integer Programming and Combinatorial Optimization*. Springer International Publishing, 2020.

Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M. Pérez, and Inigo Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256, 2013. doi: 10.1016/j.patcog.2012.07.021.

Ricard Argelaguet, Stephen J. Clark, Hisham Mohammed, L. Carine Stapel, Christel Krueger, Chantriolnt-Andreas Kapourani, Ivan Imaz-Rosshandler, Tim Lohoff, Yunlong Xiang, Courtney W. Hanna, Sebastien Smallwood, Ximena Ibarra-Soria, Florian Buettner, Guido Sanguinetti, Wei Xie, Felix Krueger, Berthold Göttgens, Peter J. Rugg-Gunn, Gavin Kelsey, Wendy Dean, Jennifer Nichols, Oliver Stegle, John C. Marioni, and Wolf Reik. Multi-omics profiling of mouse gastrulation at single-cell resolution. *Nature*, 576 (7787):487–491, 2019. doi: 10.1038/s41586-019-1825-8.

Erick Armingol, Adam Officer, Olivier Harismendy, and Nathan E. Lewis. Deciphering cell–cell interactions and communication from gene expression. *Nature Reviews Genetics*, 22(2):71–88, 2021. doi: 10.1038/s41576-020-00292-x.

David Arthur and Sergei Vassilvitskii. K-means++: the advantages of careful seeding. In *In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.

Sayan Bandyapadhyay, Tanmay Inamdar, Shreyas Pai, and Kasturi R. Varadarajan. A constant approximation for colorful k-center. *CoRR*, abs/1907.08906, 2019.

Josh Barnes and Piet Hut. A hierarchical o(n log n) force-calculation algorithm. *Nature*, 324(6096):446–449, 1986. doi: 10.1038/324446a0.

Maayan Baron, Adrian Veres, Samuel L Wolock, Aubrey L Faust, Renaud Gaujoux, Amedeo Vetere, Jennifer Hyoje Ryu, Bridget K Wagner, Shai S Shen-Orr, Allon M Klein, Douglas A Melton, and Itai Yanai. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Systems*, 3(4):346–360.e4, 2016. doi: 10.1016/j.cels.2016.08.011.

Mayank Bawa, Tyson Condie, and Prasanna Ganesan. Lsh forest: Self-tuning indexes for similarity search. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, page 651–660, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595930469. doi: 10.1145/1060745.1060840.

Volker Bergen, Marius Lange, Stefan Peidli, F. Alexander Wolf, and Fabian J. Theis. Generalizing rna velocity to transient cell states through dynamical modeling. *Nature Biotechnology*, 38(12):1408–1414, 2020. doi: 10.1038/s41587-020-0591-3.

Fernando H. Biase, Xiaoyi Cao, and Sheng Zhong. Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell rna sequencing. *Genome Research*, 24(11): 1787–1796, 2014. doi: 10.1101/gr.177725.114.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

Sebastian Böcker, Stefan Canzar, and Gunnar W. Klau. The generalized robinson-foulds metric. In *Workshop on Algorithms in Bioinformatics (WABI)*, 2013.

Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. A unifying perspective on neighbor embeddings along the attraction-repulsion spectrum. *ArXiv e-prints at https://arxiv.org/abs/2007.08902*, 2020.

Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnoly*, 36(5):411–420, 2018. doi: 10.1038/nbt.4096.

Davide Cacchiarelli, Xiaojie Qiu, Sanjay Srivatsan, Anna Manfredi, Michael Ziller, Eliah Overbey, Antonio Grimaldi, Jonna Grimsby, Prapti Pokharel, Kenneth J Livak, Shuqiang Li, Alexander Meissner, Tarjei S Mikkelsen, John L Rinn, and Cole Trapnell. Aligning Single-Cell Developmental and Reprogramming Trajectories Identifies Molecular Determinants of Myogenic Reprogramming Outcome. *Cell Systems*, 7(3):258–268.E3, 2018. doi: 10.1016/j.cels.2018.07.006.

Deng Cai and Xinlei Chen. Large scale spectral clustering with landmark-based representation. *AAAI*, pages 313–318, 2011. doi: 10.1109/TCYB.2014.2358564.

Junyue Cao, Jonathan S. Packer, Vijay Ramani, Darren A. Cusanovich, Chau Huynh, Riza Daza, Xiaojie Qiu, Choli Lee, Scott N. Furlan, Frank J. Steemers, Andrew Adey, Robert H. Waterston, Cole Trapnell, and Jay Shendure. Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science*, 357(6352):661–667, 2017. doi: 10.1126/science.aam8940.

Junyue Cao, Darren A Cusanovich, Vijay Ramani, Delasa Aghamirzaie, Hannah A Pliner, Andrew J Hill, Riza M Daza, Jose L McFaline-Figueroa, Jonathan S Packer, Lena Christiansen, Frank J Steemers, Andrew C Adey, Cole Trapnell, and Jay Shendure. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science*, 361(6409):1380–1385, 2018. doi: 10.1126/science.aau0730.

Junyue Cao, Malte Spielmann, Xiaojie Qiu, Xingfan Huang, Daniel M Ibrahim, Andrew J Hill, Fan Zhang, Stefan Mundlos, Lena Christiansen, Frank J Steemers, Cole Trapnell, and Jay Shendure. The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566(7745):496–502, 2019. doi: 10.1038/s41586-019-0969-x.

Junyue Cao, Diana R. O'Day, Hannah A. Pliner, Paul D. Kingsley, Mei Deng, Riza M. Daza, Michael A. Zager, Kimberly A. Aldinger, Ronnie Blecher-Gonen, Fan Zhang, Malte Spielmann, James Palis, Dan Doherty, Frank J. Steemers, Ian A. Glass, Cole Trapnell, and Jay Shendure. A human cell atlas of fetal gene expression. *Science*, 370(6518):eaba7721, 2020. doi: 10.1126/science.aba7721.

M. Emre Celebi, Hassan A. Kingravi, and Patricio A. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40(1):200–210, 2013. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2012.07.021.

Renchao Chen, Xiaoji Wu, Lan Jiang, and Yi Zhang. Single-cell rna-seq reveals hypothalamic cell diversity. *Cell Reports*, 18(13):3227–3241, 2017. doi: 10.1016/j.celrep.2017.03.004.

Song Chen, Blue B. Lake, and Kun Zhang. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nature Biotechnology*, 37(12):1452–1457, 2019. doi: 10.1038/s41587-019-0290-0.

H. Chouikhi, M. Charrad, and N. Ghazzali. A comparison study of clustering validity indices. In *2015 Global Summit on Computer Information Technology (GSCIT)*, pages 1–4, 2015.

Stephen J. Clark, Ricard Argelaguet, Chantriolnt-Andreas Kapourani, Thomas M. Stubbs, Heather J. Lee, Celia Alda-Catalinas, Felix Krueger, Guido Sanguinetti, Gavin Kelsey, John C. Marioni, Oliver Stegle, and Wolf Reik. scnmt-seq enables joint profiling of chromatin accessibility dna methylation and transcription in single cells. *Nature Communications*, 9(1):781, 2018. doi: 10.1038/s41467-018-03149-4.

Matthew Collin, Naomi McGovern, and Muzlifah Haniffa. Human dendritic cell subsets. *Immunology*, 140(1):22–30, 2013. doi: 10.1111/imm.12117.

Sara L. Colpitts, Nicole M. Dalton, and Phillip Scott. Il-7 receptor expression provides the potential for long-term survival of both cd62l high central memory t cells and th1 effector cells during leishmania major infection. *Journal of Immunology*, 182(9):5702–5711, 2009. doi: 10.4049/jimmunol.0803450.

Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcześniak, Daniel J. Gaffney, Laura L. Elo, Xuegong Zhang, and Ali Mortazavi. A survey of best practices for rna-seq data analysis. *Genome Biology*, 17(1):13, 2016. doi: 10.1186/s13059-016-0881-8.

Graham Cormode, Howard Karloff, and Anthony Wirth. Set cover algorithms for very large datasets. *CIKM*, pages 479–488, 2010. doi: 10.1145/1871437.1871501.

Francis Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–563, 1970. doi: 10.1038/227561a0.

D. Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20 (4):364–366, April 1977. doi: 10.1093/comjnl/20.4.364.

Qiaolin Deng, Daniel Ramsköld, Björn Reinius, and Rickard Sandberg. Single-cell rna-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, 343 (6167), 2014. doi: 10.1126/science.1245316.

Van Hoan Do, Mislav Blažević, Pablo Monteagudo, Luka Borozan, Khaled Elbassioni, Sören Laue, Francisca Rojas Ringeling, Domagoj Matijević, and Stefan Canzar. Dynamic pseudotime warping of complex single-cell trajectories. In *RECOMB 2019. Lecture Notes in Computer Science*, 2019.

Van Hoan Do, Francisca Rojas Ringeling, and Stefan Canzar. Linear-time cluster ensembles of large-scale single-cell rna-seq and multimodal data. *Genome Research*, 2021. doi: 10.1101/gr.267906.120.

H. E. Driver and A. L. Kroeber. Quantitative expression of cultural relationships. 1932.

Angelo Duò, Mark D. Robinson, and Charlotte Soneson. A systematic performance evaluation of clustering methods for single-cell rna-seq data. *F1000Research*, 7:1141, 2018. doi: 10.12688/f1000research.15666.2.

Christopher T. Fincher, Omri Wurtzel, Thom de Hoog, Kellie M. Kravarik, and Peter W. Reddien. Cell type transcriptome atlas for the planarian schmidtea mediterranea. *Science*, 360(6391), 2018. doi: 10.1126/science.aaq1736.

Chamith Y. Fonseka, Deepak A. Rao, Nikola C. Teslovich, Ilya Korsunsky, and et al. Mixed-effects association of single cells identifies an expanded effector cd4+ t cell subset in rheumatoid arthritis. *Science Translational Medicine*, 10(463), 2018. doi: 10.1126/scitranslmed.aaq0305.

Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004. doi: 10.1109/TPAMI.2004.1262185.

Ana L.N. Fred and Anil K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005. doi: 10.1109/TPAMI.2005.113.

Saskia Freytag, Luyi Tian, Ingrid Lönnstedt, Milica Ng, and Melanie Bahlo. Comparison of clustering tools in r for medium-sized 10x genomics single-cell rna-sequencing data. *F1000Research*, 7:1297, 2018. doi: 10.12688/f1000research.15809.2.

Mubeen Goolam, Antonio Scialdone, Sarah J.L. Graham, Iain C. Macaulay, Agnieszka Jedrusik, Anna Hupalowska, Thierry Voet, and John C. Marioni. Heterogeneity in oct4

and sox2 targets biases cell fate in 4-cell mouse embryos. *Cell*, 165(1):61–74, 2016. doi: 10.1016/j.cell.2016.01.047.

Gennady Gorin, Valentine Svensson, and Lior Pachter. Protein velocity and acceleration from single-cell multiomics experiments. *Genome Biology*, 21:39, 2020. doi: 10.1186/s13059-020-1945-3.

Dominic Grün, Mauro J. Muraro, Jean-Charles Boisset, Kay Wiebrands, Anna Lyubimova, Gitanjali Dharmadhikari, Maaike van den Born, Johan van Es, Erik Jansen, Hans Clevers, Eelco J.P. de Koning, and Alexander van Oudenaarden. De novo prediction of stem cell identity using single-cell transcriptome data. *Cell Stem Cell*, 19(2):266–277, 2016. doi: 10.1016/j.stem.2016.05.010.

Xinyi Guo, Yuanyuan Zhang, Liangtao Zheng, Chunhong Zheng, Jintao Song, Qiming Zhang, Boxi Kang, Zhouzerui Liu, Liang Jin, Rui Xing, Ranran Gao, Lei Zhang, Minghui Dong, Xueda Hu, Xianwen Ren, Dennis Kirchhoff, Helge Gottfried Roider, Tiansheng Yan, and Zemin Zhang. Global characterization of t cells in non-small-cell lung cancer by single-cell sequencing. *Nature Medicine*, 24(7):978–985, 2018. doi: 10.1038/s41591-018-0045-3.

L. Hagen and A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992. doi: 10.1109/43.159993.

Laleh Haghverdi, Maren Büttner, F Alexander Wolf, Florian Buettner, and Fabian J Theis. Diffusion pseudotime robustly reconstructs lineage branching. *Nature Methods*, 13(10):845–848, 2016. doi: 10.1038/nmeth.3971.

W Nicholas Haining, Jill Angelosanto, Kathleen Brosnahan, Kenneth Ross, Cynthia Hahn, Kate Russell, Linda Drury, Stephanie Norton, Lee Nadler, and Kimberly Stegmaier. High-throughput gene expression profiling of memory differentiation in primary human t cells. *BMC Immunology*, 9(44), 2008. doi: 10.1186/1471-2172-9-44.

Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M. Mauck, Shiwei Zheng, Andrew Butler, Maddie J. Lee, Aaron J. Wilk, Charlotte Darby, Michael Zager, Paul Hoffman, Marlon Stoeckius, Efthymia Papalexi, Eleni P. Mimitou, Jaison Jain, Avi Srivastava, Tim Stuart, Lamar M. Fleming, Bertrand Yeung, Angela J. Rogers, Juliana M. McElrath, Catherine A. Blish, Raphael Gottardo, Peter Smibert, and Rahul Satija. Integrated analysis of multimodal single-cell data. *Cell*, May 2021. doi: 10.1016/j.cell.2021.04.048.

Wolfgang Härdle. Applied nonparametric regression. 1990.

J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979. ISSN 00359254, 14679876.

Josip S Herman, Sagar, and Dominic Grün. Fateid infers cell fate bias in multipotent progenitors from single-cell rna-seq data. *Nature Methods*, 15(5):379–386, 2018. doi: 10.1038/nmeth.4662.

Brian Hie, Bryan Bryson, and Bonnie Berger. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nature Biotechnology*, 37(6):685–691, 2019a. doi: 10.1038/s41587-019-0113-3.

Brian Hie, Hyunghoon Cho, Benjamin DeMeo, Bryan Bryson, and Bonnie Berger. Geometric sketching compactly summarizes the single-cell transcriptomic landscape. *Cell Systems*, 8 (6):483–493.e7, 2019b. doi: 10.1016/j.cels.2019.05.003.

Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, 1986. doi: 10.1145/5925.5933.

Tomas Hruz, Oliver Laule, Gabor Szabo, Frans Wessendorp, Stefan Bleuler, Lukas Oertle, Peter Widmayer, Wilhelm Gruissem, and Philip Zimmermann. Genevestigator v3: A reference expression database for the meta-analysis of transcriptomes. *Advances in Bioinformatics*, 2008(420747):1–5, 2008. doi: 10.1155/2008/420747.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1): 193–218, 1985.

D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15 (9):850–863, 1993. doi: 10.1109/34.232073.

Pablo A. Jaskowiak, Ricardo JGB Campello, and Ivan G. Costa. On the selection of appropriate distances for gene expression data clustering. *BMC Bioinform.*, 15(2):S2, 2014. doi: 10.1186/1471-2105-15-S2-S2.

Zhicheng Ji and Hongkai Ji. Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. *Nucleic Acids Research*, 44(13):e117–e117, 2016. doi: 10.1093/nar/gkw430.

David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, 1974. doi: 10.1016/S0022-0000(74)80044-9.

W. Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2006. doi: 10.1093/biostatistics/kxj037.

Hani Jieun Kim, Yingxin Lin, Thomas A Geddes, Jean Yee Hwa Yang, and Pengyi Yang. CiteFuse enables multi-modal analysis of CITE-seq data. *Bioinformatics*, 36(14):4137–4143, 2020. doi: 10.1093/bioinformatics/btaa282.

Taiyun Kim, Irene Rui Chen, Yingxin Lin, Andy Yi-Yang Wang, Jean Yee Hwa Yang, and Pengyi Yang. Impact of similarity metrics on single-cell rna-seq data clustering. *Brief. Bioinform.*, 20(6):2316–2326, 2018. doi: 10.1093/bib/bby076.

Vladimir Yu Kiselev, Kristina Kirschner, Michael T. Schaub, Tallulah Andrews, Andrew Yiu, Tamir Chandra, Kedar N. Natarajan, Wolf Reik, Mauricio Barahona, Anthony R. Green, and Martin Hemberg. Sc3: consensus clustering of single-cell rna-seq data. *Nature Methods*, 14:483–486, 2017.

Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 2015. doi: 10.1016/j.cell.2015.04.044.

Jon M. Kleinberg. An impossibility theorem for clustering. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 463–470. MIT Press, 2003.

Pang Wei Koh, Rahul Sinha, Amira A. Barkal, Rachel M. Morganti, Angela Chen, Irving L. Weissman, Lay Teng Ang, Anshul Kundaje, and Kyle M. Loh. An atlas of transcriptional, chromatin accessibility, and surface marker changes in human mesoderm development. *Scientific Data*, 3(1):160109, 2016. doi: 10.1038/sdata.2016.109.

Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature Methods*, 16(12):1289–1296, 2019. doi: 10.1038/s41592-019-0619-0.

Roshan M. Kumar, Patrick Cahan, Alex K. Shalek, Rahul Satija, A. Jay DaleyKeyser, Hu Li, Jin Zhang, Keith Pardee, David Gennert, John J. Trombetta, Thomas C. Ferrante, Aviv Regev, George Q. Daley, and James J. Collins. Deconstructing transcriptional heterogeneity in pluripotent stem cells. *Nature*, 516(7529):56–61, 2014. doi: 10.1038/nature13920.

Gioele La Manno, Daniel Gyllborg, Simone Codeluppi, Kaneyasu Nishimura, Carmen Salto, Amit Zeisel, Lars E Borm, Simon R W Stott, Enrique M Toledo, J Carlos Villaescusa, Peter L"onnerberg, Jesper Ryge, Roger A Barker, Ernest Arenas, and Sten Linnarsson. Molecular diversity of midbrain development in mouse, human, and stem cells. *Cell*, 167 (2):566–580.e19, 2016. doi: 10.1016/j.cell.2016.09.027.

Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E. Kastriti, Peter Lönnerberg, Alessandro Furlan, Jean Fan, Lars E. Borm, Zehua Liu, David van Bruggen, Jimin Guo, Xiaoling He, Roger Barker, Erik Sundström, Gonçalo Castelo-Branco, Patrick Cramer, Igor Adameyko, Sten Linnarsson, and Peter V Kharchenko. Rna velocity of single cells. *Nature*, 560(7719): 494–498, 2018. doi: 10.1038/s41586-018-0414-6.

Michele P Lambert, Ronghua Meng, Dawn Harper, Liqing Xiao, Michael S. Marks, and Mortimer Poncz. Megakaryocytes exchange significant levels of their alpha-granular pf4 with their environment. *Blood*, 124(21):1432–1432, 2014. doi: 10.1182/blood.V124.21.1432.1432.

Michele P. Lambert, Ronghua Meng, Liqing Xiao, Dawn C. Harper, Michael S. Marks, Anna M. Kowalska, and Mortimer Poncz1. Intramedullary megakaryocytes internalize released platelet factor 4 (pf4) and store it in alpha granules. *Journal of Thrombosis and Haemostasis*, 13(10):1888–1899, 2016. doi: 10.1111/jth.13069.

Huipeng Li, Elise T Courtois, Debarka Sengupta, Yuliana Tan, Kok Hao Chen, Jolene Jie Lin Goh, Say Li Kong, Clarinda Chua, Lim Kiat Hon, Wah Siew Tan, Mark Wong, Paul Jongjoon Choi, Lawrence J K Wee, Axel M Hillmer, Iain Beehuat Tan, Paul Robson,

and Shyam Prabhakar. Reference component analysis of single-cell transcriptomes elucidates cellular heterogeneity in human colorectal tumors. *Nature Genetics*, 49(5):708–718, 2017. doi: 10.1038/ng.3818.

Peijie Lin, Michael Troup, and Joshua W. K. Ho. Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome Biology*, 18(1):59, 2017. doi: 10.1186/s13059-017-1188-0.

Yingxin Lin and Hani Jieun Kim. Citefuse: getting started, 2020. `https://sydneybiox.github.io/CiteFuse/articles/CiteFuse.html`. Accessed 15 March 2020.

George C. Linderman, Manas Rachh, Jeremy G. Hoskins, Stefan Steinerberger, and Yuval Kluger. Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data. *Nature Methods*, 16(3):243–245, 2019. doi: 10.1038/s41592-018-0308-4.

Malte D Luecken and Fabian J Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6):e8746, 2019. doi: 10.15252/msb.20188746.

Sai Ma, Bing Zhang, Lindsay LaFave, Zachary Chiang, Yan Hu, Jiarui Ding, Alison Brack, Vinay K. Kartha, Travis Law, Caleb Lareau, Ya-Chieh Hsu, Aviv Regev, and Jason D. Buenrostro. Chromatin potential identified by shared single-cell profiling of rna and chromatin. *Cell*, 183(4):"1103–1116", 2020. doi: 10.1016/j.cell.2020.09.056.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv e-prints at https://arxiv.org/abs/1802.03426*, 2018.

Miriam Merad, Priyanka Sathe, Julie Helft, Jennifer Miller, and Arthur Mortha. The dendritic cell lineage: ontogeny and function of dendritic cells and their subsets in the steady state and the inflamed setting. *Annual Review of Immunology*, 31:563–604, 2013. doi: 10.1146/annurev-immunol-020711-074950.

Eleni P. Mimitou, Anthony Cheng, Antonino Montalbano, Stephanie Hao, Marlon Stoeckius, Mateusz Legut, Timothy Roush, Alberto Herrera, Efthymia Papalexi, Zhengqing Ouyang, Rahul Satija, Neville E. Sanjana, Sergei B. Koralov, and Peter Smibert. Multiplexed detection of proteins, transcriptomes, clonotypes and crispr perturbations in single cells. *Nature Methods*, 16(5):409–412, 2019. doi: 10.1038/s41592-019-0392-0.

Mauro J Muraro, Gitanjali Dharmadhikari, Dominic Grün, Nathalie Groen, Tim Dielen, Erik Jansen, Leon van Gurp, Marten A Engelse, Francoise Carlotti, Eelco J P. de Koning, and Alexander van Oudenaarden. A single-cell transcriptome atlas of the human pancreas. *Cell Systems*, 3(4):385–394.e3, 2016. doi: 10.1016/j.cels.2016.09.002.

Sonia Nestorowa, Fiona K Hamey, Blanca Pijuan Sala, Evangelia Diamanti, Mairi Shepherd, Elisa Laurenti, Nicola K Wilson, David G Kent, and Berthold G"ottgens. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood*, 128(8):e20–e31, 2016. doi: 10.1182/blood-2016-05-716480.

Quy H. Nguyen, Nicholas Pervolarakis, Kerrigan Blake, Dennis Ma, Ryan Tevia Davis, Nathan James, Anh T. Phung, Elizabeth Willey, Raj Kumar, Eric Jabart, Ian Driver, Jason Rock, Andrei Goga, Seema A. Khan, Devon A. Lawson, Zena Werb, and Kai Kessenbrock. Profiling human breast epithelial cells using single cell rna sequencing identifies cell diversity. *Nature Communications*, 9(1):2028, 2018. doi: 10.1038/s41467-018-04334-1.

Kazuyuki Ogawa, Yasushi Takamori, Kunou Suzuki, Masayuki Nagasawa, Shoichi Takano, Yoshihito Kasahara, Yoshiko Nakamura, Shigemi Kondo, Kazuo Sugamura, and Kinya Nagata. Granulysin in human serum as a marker of cell-mediated immunity. *European Journal of Immunology*, 33(7):1925–1933, 2003. doi: 10.1002/eji.200323977.

Anoop P. Patel, Itay Tirosh, John J. Trombetta, Alex K. Shalek, Shawn M. Gillespie, Hiroaki Wakimoto, Daniel P. Cahill, Brian V. Nahed, William T. Curry, Robert L. Martuza, David N. Louis, Orit Rozenblatt-Rosen, Mario L. Suvá, Aviv Regev, and Bradley E. Bernstein. Single-cell rna-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science*, 344(6190):1396–1401, 2014. doi: 10.1126/science.1254257.

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

Vanessa M Peterson, Kelvin Xi Zhang, Namit Kumar, Jerelyn Wong, Lixia Li, Douglas C Wilson, Renee Moore, Terrill K McClanahan, Svetlana Sadekova, and Joel A Klappenbach. Multiplexed quantification of proteins and transcripts in single cells. *Nature Biotechnology*, 35(10):936–939, 2017. doi: 10.1038/nbt.3973.

Alex A Pollen, Tomasz J Nowakowski, Joe Shuga, Xiaohui Wang, Anne A Leyrat, Jan H Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, Naveen Ramalingam, Gang Sun, Myo Thu, Michael Norris, Ronald Lebofsky, Dominique Toppani, Darnell W Kemp II, Michael Wong, Barry Clerkson, Brittnee N Jones, Shiquan Wu, Lawrence Knutsson, Beatriz Alvarado, Jing Wang, Lesley S Weaver, Andrew P May, Robert C Jones, Marc A Unger, Arnold R Kriegstein, and Jay A A West. Low-coverage single-cell mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature Biotechnology*, 32(10):1053–1058, 2014. doi: 10.1038/nbt.2967.

Gilles Puy, Nicolas Tremblay, Rémi Gribonval, and Pierre Vandergheynst. Random sampling of bandlimited signals on graphs, 2016.

Xiaojie Qiu, Qi Mao, Ying Tang, Li Wang, Raghav Chawla, Hannah A Pliner, and Cole Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nature Methods*, 14(10):979–982, 2017. doi: 10.1038/nmeth.4402.

Xiaojie Qiu, Yan Zhang, Dian Yang, Shayan Hosseinzadeh, Li Wang, Ruoshi Yuan, Song Xu, Yian Ma, Joseph Replogle, Spyros Darmanis, Jianhua Xing, and Jonathan S Weissman. Mapping vector field of single cells. *bioRxiv*, 2019. doi: 10.1101/696724.

Mostafa Rahmani and George K. Atia. Spatial random sampling: A structure-preserving data sketching tool. *IEEE Signal Processing Letters*, 24:1398–1402, 2017.

Aviv Regev, Sarah A Teichmann, Eric S Lander, Ido Amit, Christophe Benoist, Ewan Birney, Bernd Bodenmiller, Peter Campbell, Piero Carninci, Menna Clatworthy, Hans Clevers, Bart Deplancke, Ian Dunham, James Eberwine, Roland Eils, Wolfgang Enard, Andrew Farmer, Lars Fugger, Berthold Göttgens, Nir Hacohen, Muzlifah Haniffa, Martin Hemberg, Seung Kim, Paul Klenerman, Arnold Kriegstein, Ed Lein, Sten Linnarsson, Emma Lundberg, Joakim Lundeberg, Partha Majumder, John C Marioni, Miriam Merad, Musa Mhlanga, Martijn Nawijn, Mihai Netea, Garry Nolan, Dana Pe'er, Anthony Phillipakis, Chris P Ponting, Stephen Quake, Wolf Reik, Orit Rozenblatt-Rosen, Joshua Sanes, Rahul Satija, Ton N Schumacher, Alex Shalek, Ehud Shapiro, Padmanee Sharma, Jay W Shin, Oliver Stegle, Michael Stratton, Michael J T Stubbington, Fabian J Theis, Matthias Uhlen, Alexander van Oudenaarden, Allon Wagner, Fiona Watt, Jonathan Weissman, Barbara Wold, Ramnik Xavier, Nir Yosef, and Human Cell Atlas Meeting Participants. Science forum: The human cell atlas. *eLife*, 6:e27041, 2017. doi: 10.7554/eLife.27041.

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, 2007. Association for Computational Linguistics.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. doi: 10.1016/0377-0427(87)90125-7.

Wouter Saelens, Robrecht Cannoodt, Helena Todorov, and Yvan Saeys. A comparison of single-cell trajectory inference methods. *Nature Biotechnology*, 37(5):547–554, 2019. doi: 10.1038/s41587-019-0071-9.

Kazuki Sakurai, Tohru Fujiwara, Shin Hasegawa, Yoko Okitsu, Noriko Fukuhara, Yasushi Onishi, Minami Yamada-Fujiwara, Ryo Ichinohasama, and Hideo Harigae. Inhibition of human primary megakaryocyte differentiation by anagrelide: a gene expression profiling analysis. *International Journal of Hematology*, 104(2):190–199, 2016. doi: 10.1007/s12185-016-2006-2.

Rahul Satija. Using seurat with multi-modal data, 2019. `https://satijalab.org/seurat/v3.1/multimodal_vignette.html`. Accessed 15 December 2019.

Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5):495–502, 2015. doi: 10.1038/nbt.3192.

Arpiar Saunders, Evan Z. Macosko, Alec Wysoker, Melissa Goldman, Fenna M. Krienen, Heather de Rivera, Elizabeth Bien, Matthew Baum, Laura Bortolin, Shuyu Wang, Aleksandrina Goeva, James Nemesh, Nolan Kamitaki, Sara Brumbaugh, David Kulp, and Steven A. McCarroll. Molecular diversity and specializations among the cells of the adult mouse brain. *Cell*, 174(4):1015–1030.e16, 2018. doi: 10.1016/j.cell.2018.07.028.

Peter Savas, Balaji Virassamy, Chengzhong Ye, Agus Salim, Christopher P. Mintoff, Franco Caramia, Roberto Salgado, David J. Byrne, Zhi L. Teo, Sathana Dushyanthen, Ann

Byrne, Lironne Wein, Stephen J. Luen, Catherine Poliness, Sophie S. Nightingale, Anita S. Skandarajah, David E. Gyorki, Chantel M. Thornton, Paul A. Beavis, Stephen B. Fox, Phillip K. Darcy, Terence P. Speed, Laura K. Mackay, Paul J. Neeson, Sherene Loi, and Kathleen Cuningham Foundation Consortium for Research into Familial Breast Cancer (kConFab). Single-cell profiling of breast cancer t cells reveals a tissue-resident memory subset associated with improved prognosis. *Nature Medicine*, 24(7):986–993, 2018. doi: 10.1038/s41591-018-0078-7.

Alex K. Shalek, Rahul Satija, John J. Trombetta Joe Shuga, Dave Gennert, Diana Lu, Peilin Chen, Rona S. Gertner, Jellert T. Gaublomme, Nir Yosef, Schraga Schwartz, Brian Fowler, Suzanne Weaver, Jing Wang, Xiaohui Wang, Ruihua Ding, Raktima Raychowdhury, Nir Friedman, Nir Hacohen, Hongkun Park, Andrew P. May, and Aviv Regev. Single-cell rna-seq reveals dynamic paracrine control of cellular variation. *Nature*, 498(510):363–369, 2014. doi: 10.1038/nature13437.

Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. doi: 10.1109/34.868688.

Hiroyuki Shinnou and Minoru Sasaki. Spectral clustering for a large data set by reducing the similarity matrix size. *Proceedings of the Sixth International Language Resources and Evaluation*, 2008.

R. Sibson. SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, January 1973. doi: 10.1093/comjnl/16.1.30.

Rohit Singh, Brian L. Hie, Ashwin Narayan, and Bonnie Berger. Schema: metric learning enables interpretable synthesis of heterogeneous single-cell modalities. *Genome Biology*, 22(1):131, 2021. doi: 10.1186/s13059-021-02313-2.

Debajyoti Sinha, Akhilesh Kumar, Himanshu Kumar, Sanghamitra Bandyopadhyay, and Debarka Sengupta. dropClust: efficient clustering of ultra-large scRNA-seq data. *Nucleic Acids Research*, 46(6):e36–e36, 2018. doi: 10.1093/nar/gky007.

Petr Slavík. A tight analysis of the greedy algorithm for set cover. *J. Algorithms*, 25(2): 237–254, 1997. doi: 10.1006/jagm.1997.0887.

C. Allison Stewart, Carl M. Gay, Yuanxin Xi, Santhosh Sivajothi, V. Sivakamasundari, Junya Fujimoto, Mohan Bolisetty, Patrice M. Hartsfield, Veerakumar Balasubramaniyan, Milind D. Chalishazar, Cesar Moran, Neda Kalhor, John Stewart, Hai Tran, Stephen G. Swisher, Jack A. Roth, Jianjun Zhang, John de Groot, Bonnie Glisson, Trudy G. Oliver, John V. Heymach, Ignacio Wistuba, Paul Robson, Jing Wang, and Lauren Averett Byers. Single-cell analyses reveal increased intratumoral heterogeneity after the onset of therapy resistance in small-cell lung cancer. *Nature Cancer*, 1(4):423–436, 2020. doi: 10.1038/s43018-019-0020-z.

Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods*, 14(9): 865–868, 2017. doi: 10.1038/nmeth.4380.

Kelly Street, Davide Risso, Russell B. Fletcher, Diya Das, John Ngai, Nir Yosef, Elizabeth Purdom, and Sandrine Dudoit. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics*, 19(1):477, 2018. doi: 10.1186/s12864-018-4772-0.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2003.

Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive Integration of Single-Cell Data. *Cell*, 177(7):1888–1902.e21, 2019.

C. Studholme, D.L.G. Hill, and D.J. Hawkes. An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, 32(1):71–86, 1999. doi: 10.1016/S0031-3203(98)00091-0.

Valentine Svensson, Kedar Nath Natarajan, Lam-Ha Ly, Ricardo J Miragaia, Charlotte Labalette, Iain C Macaulay, Ana Cvejic, and Sarah A Teichmann. Power analysis of single-cell rna-sequencing experiments. *Nature Methods*, 14(4):381–387, 2017. doi: 10.1038/nmeth.4220.

Valentine Svensson, Roser Vento-Tormo, and Sarah A Teichmann. Exponential scaling of single-cell rna-seq in the past decade. *Nature Protocols*, 13(4):599–604, 2018. doi: 10.1038/nprot.2017.149.

Luyi Tian, Xueyi Dong, Saskia Freytag, Kim-Anh Lê Cao, Shian Su, Abolfazl JalalAbadi, Daniela Amann-Zalcenstein, Tom S. Weber, Azadeh Seidi, Jafar S. Jabbari, Shalin H. Naik, and Matthew E. Ritchie. Benchmarking single cell rna-sequencing analysis pipelines using mixture control experiments. *Nature Methods*, 16(6):479–487, 2019. doi: 10.1038/s41592-019-0425-8.

Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32(4):381–386, 2014. doi: 10.1038/nbt.2859.

Nicolas Tremblay, Gilles Puy, Rémi Gribonval, and Pierre Vandergheynst. Compressive spectral clustering. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, ICML'16, pages 1002–1011. JMLR.org, 2016.

Barbara Treutlein, Doug G. Brownfield, Angela R. Wu, Norma F. Neff, Gary L. Mantalas, F. Hernan Espinoza, Tushar J. Desai, Mark A. Krasnow, and Stephen R. Quake. Reconstructing lineage hierarchies of the distal lung epithelium using single-cell rna-seq. *Nature*, 509(7500):371–375, 2014. doi: 10.1038/nature13173.

Martin A. Turman, Toshio Yabe, Cynthia McSherry, Fritz H. Bach, and Jeffrey P. Houchins. Characterization of a novel gene (nkg7) on human chromosome 19 that is expressed in natural killer cells and t cells. *Human Immunology*, 36(1):34–40, 1993. doi: 10.1016/0198-8859(93)90006-m.

Laurens van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15(93):3221–3245, 2014.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Vipin Vijayan. Fast svd and pca, 2020. `https://www.mathworks.com/matlabcentral/fileexchange/47132-fast-svd-and-pca`. MATLAB Central File Exchange. Retrieved October 30, 2020.

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, 2007.

Bo Wang, Aziz M. Mezlini, Feyyaz Demir, Marc Fiume, Zhuowen Tu, Michael Brudno, Benjamin Haibe-Kains, and Anna Goldenberg. Similarity network fusion for aggregating data types on a genomic scale. *Nature Methods*, 11(3):333–337, 2014. doi: 10.1038/nmeth.2810.

Xinjun Wang, Zhe Sun, Yanfu Zhang, Zhongli Xu, Hongyi Xin, Heng Huang, Richard H Duerr, Kong Chen, Ying Ding, and Wei Chen. BREM-SC: a bayesian random effects mixture model for joint clustering single cell multi-omics data. *Nucleic Acids Research*, 48 (11):5814–5824, 2020. doi: 10.1093/nar/gkaa314.

F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, 2018. doi: 10.1186/s13059-017-1382-0.

Yurong Xin, Jinrang Kim, Haruka Okamoto, Min Ni, Yi Wei, Christina Adler, Andrew J Murphy, George D Yancopoulos, Calvin Lin, and Jesper Gromada. Rna sequencing of single human islet cells reveals type 2 diabetes genes. *Cell Metabolism*, 24(4):608–615, 2016. doi: 10.1016/j.cmet.2016.08.018.

Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 267–273, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581136463. doi: 10.1145/860435.860485.

Liying Yan, Mingyu Yang, Hongshan Guo, Lu Yang, Jun Wu, Rong Li, Ping Liu, Ying Lian, Xiaoying Zheng, Jie Yan, Jin Huang, Ming Li, Xinglong Wu, Lu Wen, Kaiqin Lao, Ruiqiang Li, Jie Qiao, and Fuchou Tang. Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells. *Nature Structural & Molecular Biology*, 20(9): 1131–1139, 2013. doi: 10.1038/nsmb.2660.

Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: simulation of single-cell rna sequencing data. *Genome Biology*, 18(1):174, 2017. doi: 10.1186/s13059-017-1305-0.

Luke Zappia, Belinda Phipson, and Alicia Oshlack. Introduction to splatter, 2020. `https://bioconductor.org/packages/devel/bioc/vignettes/splatter/inst/doc/splatter.html`. Accessed 15 October 2020.

Amit Zeisel, Ana B. Muñoz-Manchado, Simone Codeluppi, Peter Lönnerberg, Gioele La Manno, Anna Juréus, Sueli Marques, Hermany Munguba, Liqun He, Christer Betsholtz,

Charlotte Rolny, Gonçalo Castelo-Branco, Jens Hjerling-Leffler, and Sten Linnarsson. Cell types in the mouse cortex and hippocampus revealed by single-cell rna-seq. *Science*, 347 (6226):1138–1142, 2015. doi: 10.1126/science.aaa1934.

Amit Zeisel, Hannah Hochgerner, Peter Lonnerberg, Anna Johnsson, Fatima Memic, Job van der Zwan, Martin Haring, Emelie Braun, Lars E. Borm, Gioele La Manno, Simone Codeluppi, Alessandro Furlan, Kawai Lee, Nathan Skene, Kenneth D. Harris, Jens Hjerling-Leffler, Ernest Arenas, Patrik Ernfors, Ulrika Marklund, and Sten Linnarsson. Molecular architecture of the mouse nervous system. *Cell*, 174(4):999–1014.e22, 2018. doi: 10.1016/j.cell.2018.06.021.

Ziqi Zhang and Xiuwei Zhang. Inference of high-resolution trajectories in single cell rna-seq data from rna velocity. *bioRxiv*, 2021. doi: 10.1101/2020.09.30.321125.

Jiaping Zhao and Laurent Itti. shapedtw: Shape dynamic time warping. *Pattern Recognition*, 74:171–184, 2018. doi: 10.1016/j.patcog.2017.09.020.

Grace X. Y. Zheng, Jessica M. Terry, Phillip Belgrader, Paul Ryvkin, Zachary W. Bent, Ryan Wilson, Solongo B. Ziraldo, Tobias D. Wheeler, Geoff P. McDermott, Junjie Zhu, Mark T. Gregory, Joe Shuga, Luz Montesclaros, Jason G. Underwood, Donald A. Masquelier, Stefanie Y. Nishimura, Michael Schnall-Levin, Paul W. Wyatt, Christopher M. Hindson, Rajiv Bharadwaj, Alexander Wong, Kevin D. Ness, Lan W. Beppu, H. Joachim Deeg, Christopher McFarland, Keith R. Loeb, William J. Valente, Nolan G. Ericson, Emily A. Stevens, Jerald P. Radich, Tarjei S. Mikkelsen, Benjamin J. Hindson, and Jason H. Bielas. Massively parallel digital transcriptional profiling of single cells. *Nature Communications*, 8(1):14049, 2017. doi: 10.1038/ncomms14049.

Suijuan Zhong, Shu Zhang, Xiaoying Fan, Qian Wu, Liying Yan, Ji Dong, Haofeng Zhang, Long Li, Le Sun, Na Pan, Xiaohui Xu, Fuchou Tang, Jun Zhang, Jie Qiao, and Xiaoqun Wang. A single-cell rna-seq survey of the developmental landscape of the human prefrontal cortex. *Nature*, 555(7697):524–528, 2018. doi: 10.1038/nature25980.

Chenxu Zhu, Sebastian Preissl, and Bing Ren. Single-cell multimodal omics: the power of many. *Nature Publishing Group*, 17(1):11–14, 2020.

Christoph Ziegenhain, Beate Vieth, Swati Parekh, Björn Reinius, Amy Guillaumet-Adkins, Martha Smets, Heinrich Leonhardt, Holger Heyn, Ines Hellmann, and Wolfgang Enard. Comparative analysis of single-cell rna sequencing methods. *Molecular Cell*, 65(4):631–643.e4, 2017. doi: 10.1016/j.molcel.2017.01.023.

J. Zubin. A technique for measuring like-mindedness. *The Journal of Abnormal and Social Psychology*, 33(4):508–516, 1938.