TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

CELL TYPE IDENTIFICATION, DIFFERENTIAL EXPRESSION ANALYSIS AND TRAJECTORY INFERENCE IN SINGLE-CELL TRANSCRIPTOMICS

Johannes Smolander

TURUN YLIOPISTON JULKAISUJA – ANNALES UNIVERSITATIS TURKUENSIS
SARJA – SER. F OSA – TOM. 25 | TECHNICA – INFORMATICA | TURKU 2023

TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

CELL TYPE IDENTIFICATION, DIFFERENTIAL EXPRESSION ANALYSIS AND TRAJECTORY INFERENCE IN SINGLE-CELL TRANSCRIPTOMICS

Johannes Smolander

TURUN YLIOPISTON JULKAISUJA – ANNALES UNIVERSITATIS TURKUENSIS
SARJA – SER. F OSA – TOM. 25 | TECHNICA – INFORMATICA | TURKU 2023

University of Turku

Faculty of Technology
Department of Computing
Computer Science
Doctoral Programme in Technology

Supervised by

Professor, Laura L. Elo
Turku Bioscience Centre, University of Turku and Åbo Akademi University

Reviewed by

Assistant Professor, Kelly Street
Keck School of Medicine, University of Southern California

Doctor, Irene Papatheodorou
European Bioinformatics Institute, European Molecular Biology Laboratory, Wellcome Trust Genome Campus

Opponent

Associate Professor, Mark D. Robinson
Department of Molecular Life Sciences, University of Zurich

The originality of this publication has been checked in accordance with the University of Turku quality assurance system using the Turnitin OriginalityCheck service.

ISBN 978-951-29-9392-5 (Print)
ISBN 978-951-29-9393-2 (PDF)
ISSN 2736-9390 (Painettu/Print)
ISSN 2736-9684 (Sähköinen/Online)
Painosalama, Turku, Finland 2023

To my friends and family

UNIVERSITY OF TURKU
Faculty of Technology
Department of Computing
Computer Science
SMOLANDER, JOHANNES: Cell type identification, differential expression analysis and trajectory inference in single-cell transcriptomics
Doctoral dissertation, 129 pp.
Doctoral Programme in Technology
February 2023

ABSTRACT

Single-cell RNA-sequencing (scRNA-seq) is a cutting-edge technology that enables to quantify the transcriptome, the set of expressed RNA transcripts, of a group of cells at the single-cell level. It represents a significant upgrade from bulk RNA-seq, which measures the combined signal of thousands of cells. Measuring gene expression by bulk RNA-seq is an invaluable tool for biomedical researchers who want to understand how cells alter their gene expression due to an illness, differentiation, external stimulus, or other events. Similarly, scRNA-seq has become an essential method for biomedical researchers, and it has brought several new applications previously unavailable with bulk RNA-seq.

scRNA-seq has the same applications as bulk RNA-seq. However, the single-cell resolution also enables cell annotation based on gene markers of clusters, that is, cell populations that have been identified based on machine learning to be, on average, dissimilar at the transcriptomic level. Researchers can use the cell clusters to detect cell-type-specific gene expression changes between conditions such as case and control groups. Clustering can sometimes even discover entirely new cell types. Besides the cluster-level representation, the single-cell resolution also enables to model cells as a trajectory, representing how the cells are related at the cell level and what is the dynamic differentiation process that the cells undergo in a tissue.

This thesis introduces new computational methods for cell type identification and trajectory inference from scRNA-seq data. A new cell type identification method (ILoReg) was proposed, which enables high-resolution clustering of cells into populations with subtle transcriptomic differences. In addition, two new trajectory inference methods were developed: scShaper, which is an accurate and robust method for inferring linear trajectories; and Totem, which is a user-friendly and flexible method for inferring tree-shaped trajectories. In addition, one of the works benchmarked methods for detecting cell-type-specific differential states from scRNA-seq data with multiple subjects per comparison group, requiring tailored methods to confront false discoveries.

KEYWORDS: Single-cell RNA sequencing, transcriptome, cell type identification, trajectory inference, differential expression

TURUN YLIOPISTO
Teknillinen tiedekunta
Tietotekniikan laitos
Tietojenkäsittelytiede
SMOLANDER, JOHANNES: Cell type identification, differential expression analysis and trajectory inference in single-cell transcriptomics
Väitöskirja, 129 s.
Teknologian tohtoriohjelma
Helmikuu 2023

TIIVISTELMÄ

Yksisoluinen RNA-sekvensointi on huipputeknologia, joka mahdollistaa transkriptomin eli ilmentyneiden RNA-transkriptien laskennallisen määrittämisen joukolle soluja yhden solun tarkkuudella, ja sen kehittäminen oli merkittävä askel eteenpäin perinteisestä bulkki-RNA-sekvensoinnista, joka mittaa tuhansien solujen yhteistä signaalia. Bulkki-RNA-sekvensointi on tärkeä työväline biolääketieteen tutkijoille, jotka haluavat ymmärtää miten solut muuttavat geenien ilmentymistä sairauden, erilaistumisen, ulkoisen ärsykkeen tai muun tapahtuman seurauksena. Yksisoluisesta RNA-sekvensoinnista on vastaavasti kehittynyt tärkeä työväline tutkijoille, ja se on tuonut useita uusia sovelluksia.

Yksisoluisella RNA-sekvensoinnilla on samat sovellukset kuin bulkki-RNA-sekvensoinnilla, mutta sen lisäksi se mahdollistaa solujen tunnistamisen geenimarkkerien perusteella. Geenimarkkerit etsitään tilastollisin menetelmin solupopulaatioille, joiden on tunnistettu koneoppimisen menetelmin muodostavan transkriptomitasolla keskenään erilaisia joukkoja eli klustereita. Tutkijat voivat hyödyntää soluklustereita tutkimaan geeniekspressioeroja solutyyppien sisällä esimerkiksi sairaiden ja terveiden välillä, ja joskus klusterointi voi jopa tunnistaa uusia solutyyppejä. Yksisolutason mittaukset mahdollistavat myös solujen mallintamisen trajektorina, joka esittää kuinka solut kehittyvät dynaamisesti toisistaan geenien ilmentymistä vaativien prosessien aikana.

Tämä väitöskirja esittelee uusia laskennallisia menetelmiä solutyyppien ja trajektorien tunnistamiseen yksisoluisesta RNA-sekvensointidatasta. Väitöskirja esittelee uuden solutyyppitunnistusmenetelmän (ILoReg), joka mahdollistaa hienovaraisia geeniekspressioeroja sisältävien solutyyppien tunnistamisen. Sen lisäksi väitöskirjassa kehitettiin kaksi uutta trajektorin tunnistusmenetelmää: scShaper, joka on tarkka ja robusti menetelmä lineaaristen trajektorien tunnistamiseen, sekä Totem, joka on käyttäjäystävällinen ja joustava menetelmä puumallisten trajektorien tunnistamiseen. Lopuksi väitöskirjassa vertailtiin menetelmiä solutyyppien sisäisten geeniekspressioerojen tunnistamiseen ryhmien välillä, joissa on useita koehenkilöitä tai muita biologisia replikaatteja, mikä vaatii erityisiä menetelmiä väärien positiivisten löydösten vähentämiseen.

ASIASANAT: yksisoluinen RNA-sekvensointi, klusterointi, trajektorin tunnistus, geeniekspressio

Acknowledgements

I would like to thank my supervisor, Prof. Laura L. Elo, for recruiting me into the Elo lab research group, giving me critical support in writing, designing, and funding my research, as well as for trusting me to implement many of my own project ideas.

Thanks to Sini and Mikko for manuscript proofreading and supervision. Special thanks to Sini for also participating in designing and conducting the studies.

The powerful computational resources of the Elo lab played a vital role in performing the computational experiments. A big thank you to Esko, Sami, Jukka and Tomi who have been maintaining these services and assisting with technical issues.

During my first year of working at the Elo lab, I received mentoring from Asta, which was helpful in learning the basics of RNA-seq analysis, customer interaction, manuscript writing, and scientific work in general.

Many thanks to all members of the Elo lab for creating a positive work environment that stimulates scientific research. During my PhD studies, I have had many fruitful discussions with all members of the group.

Last but not least, I want to thank the funders. Research of the Elo lab is supported by the European Research Council (ERC), European Union's Horizon 2020 research and innovation programme, Academy of Finland, Sigrid Juselius Foundation, University of Turku, Åbo Akademi University, Biocenter Finland, and ELIXIR Finland. Special thanks to the University of Turku Graduate School (UTUGS) for personally awarding me with a two-year grant.

January 3, 2023
Johannes Smolander

Table of Contents

Abbreviations

ARI	Adjusted Rand index
ASW	Average silhouette width
AUROC	Area under the receiver operating characteristic
cDNA	Complementary DNA
CLARA	Clustering for large applications
CSPA	Cluster-based similarity partitioning algorithm
DE	Differential expression
DS	Differential state
FDR	False discovery rate
FPR	False positive rate
GAM	Generalized additive model
GNG	Growing neural gas
ICP	Iterative clustering projection
k-NN	*k*-nearest neigbor
LMDS	Landscape multi-dimensional scaling
logFC	Logarithmic fold chance
LOESS	Locally estimated scatterplot smoothing
LR	Logistic regression
MCC	Matthew's correlation coefficient
MDS	Multi-dimensional scaling
mRNA	Messenger RNA
MST	Minimum spanning tree
ncRNA	Non-coding RNA
NGS	Next-generation sequencing
PBMC	Peripheral blood mononuclear cell
PCA	Principal component analysis
QC	Quality control
RF	Random forest
RNA-seq	RNA sequencing
ROC	Receiver operating characteristic
scRNA-seq	Single-cell RNA sequencing
TMM	Trimmed mean of M values
t-SNE	*t*-distributed stochastic neighbor embedding

UMAP	Uniform manifold approximation and projection
UMI	Unique molecular identifier
VRC	Variance ratio criterion

List of Original Publications

This dissertation is based on the following original publications, which are referred to in the text by their Roman numerals:

I Johannes Smolander, Sini Junttila, Mikko S Venäläinen, Laura L Elo, ILoReg: a tool for high-resolution cell population identification from single-cell RNA-seq data, Bioinformatics, Volume 37, Issue 8, 15 April 2021, Pages 1107–1114, https://doi.org/10.1093/bioinformatics/btaa919.

II Sini Junttila*, Johannes Smolander*, Laura L Elo, Benchmarking methods for detecting differential states between conditions from multi-subject single-cell RNA-seq data, Briefings in Bioinformatics, Volume 23, Issue 5, September 2022, https://doi.org/10.1093/bib/bbac286.

III Johannes Smolander, Sini Junttila, Mikko S Venäläinen, Laura L Elo, scShaper: an ensemble method for fast and accurate linear trajectory inference from single-cell RNA-seq data, Bioinformatics, Volume 38, Issue 5, 1 March 2022, Pages 1328–1335, https://doi.org/10.1093/bioinformatics/btab831.

IV Johannes Smolander, Sini Junttila, Laura L Elo, Totem: a user-friendly tool for clustering-based inference of tree-shaped trajectories from single-cell data. Manuscript.

* shared first author.

The original publications have been reproduced with the permission of the copyright holders.

1 Introduction

RNA sequencing (RNA-seq) is the gold-standard technique for quantifying a specimen's transcriptome, the complete set of expressed RNA transcripts [1]. RNA-seq superseded the microarray as the leading transcriptome profiling technology due to its superior ability to measure the whole transcriptome, which was a significant improvement over microarrays that use predefined probes to measure the expression of predefined genes based on hybridization [2]. Initially, RNA-seq could only be used to quantify the expression in cell bulks consisting of thousands of cells, and it was not possible to demultiplex the sequenced RNA fragments, i.e., map the individual RNA fragments to their parent cells. Therefore, the experiments needed to be carefully conducted to ensure that the sequenced specimens contained only specific cell types. The cell isolation step was prone to errors, and the resulting transcript signal would represent only the joint signal of the cells instead of individual cells. However, this "bulk RNA-seq" technology remains widely used to this day.

Single-cell RNA-seq (scRNA-seq) was developed to address the limitations of bulk RNA-seq by enabling the measurement of the transcriptomes of individual cells. From the large set of developed scRNA-seq protocols [3] that can extract the RNA material from cells and transform it into a DNA library that can be sequenced and demultiplexed, the first protocols were plate-based methods that were limited to a small number of cells, typically 96, but were able to produce a full-transcriptome profile of each cell [4]. Shortly after, droplet-based systems [5; 6] emerged, which were able to profile a significantly higher number of cells, thousands or millions, at the cost of a more limited transcriptome coverage for each cell. At the moment, the droplet-based systems such as 10X Chromium are the market leader.

The single-cell resolution enables several applications that were not possible with bulk RNA-seq. In particular, the cell-level transcriptome profile enables using unsupervised learning, specifically clustering, to partition the cells into subsets (clusters) that have high within-cluster similarity but low between-cluster similarity. If the clustering is performed optimally, the clusters represent an accurate segregation of the actual cell types of the sequenced sample. The clusters can be subsequently identified based on gene markers identified by performing statistical testing between the clusters [7; 8; 9].

Cell type identification from scRNA-seq data based on clustering has been a central research topic from the early days of the technology [10; 11; 12; 13; 14].

The choice of the clustering algorithm is not the only critical step, and the preprocessing steps, i.e., normalization, quality control, and dimensionality reduction, that are performed prior to clustering all significantly impact the clustering result. In **Publication I** of this thesis, we introduced a cell type identification method, ILoReg, designed to improve the identification of cell types with subtle transcriptomic differences. ILoReg uses a novel iterative machine learning algorithm, the iterative clustering projection (ICP), to find clusters of cells that maximize the similarity between the clustering and its projection by a logistic regression model. The logistic regression model performs feature selection that selects highly variable genes essential for segregating different cell types, helping to reduce noise and segregate cell types that are separable by a small number of highly expressed genes. After convergence, ILoReg uses the cluster probabilities of the logistic regression model as features that are inputted to principal component analysis (PCA) and subsequently clustered, and the ICP is hence acting as a feature extraction step between normalization and clustering. We used a human peripheral blood mononuclear cell (PBMC) dataset [5] and a human pancreas dataset [15] to investigate the cell populations that can be identified visually based on a nonlinear embedding, such as *t*-distributed stochastic neighbor embedding (*t*-SNE). In addition, we evaluated the clustering performance of ILoReg and four other scRNA-seq cell type identification methods [9; 11; 12; 14].

In **Publication II**, we continued with a topic downstream of cell type identification by investigating how to optimally perform cell-type-specific differential expression (DE) analysis between conditions, such as case and control patients, when both conditions include multiple subjects or other biological replicates. DE analysis of multi-subject scRNA-seq data is prone to false discoveries [16; 17; 18; 19; 20] because the cells form a hierarchical structure in which cells within each subject are more similar in their expression than cells between the subjects. Statistical tests popular in scRNA-seq DE analysis, such as Wilcoxon rank-sum test and MAST [21], assume that the cells originate from a single population of statistically independent samples, leading to inflated *p*-values and spurious findings. Due to the single-cell resolution, the naïve methods that do not model subjects in any way can, for example, report a positive finding when a gene is upregulated in only one of the subjects. We benchmarked naïve DE analysis methods with pseudo-bulk methods [17; 19; 20] that aggregate the data at the subject level and mixed models [18; 19; 22] that model the subjects as a random effect. We also considered a fourth method type, the latent variable models of the Seurat toolkit [9; 23], which can be used to account for various batch effects in DE analysis with the naïve methods.

While the cluster-level representation of cells is helpful, it is deficient because it does not describe how the cell types are related. For example, in a sample that consists of CD4+ T cells, it would also be informative to model how the CD4+ T cell subsets, such as naïve CD4+ T cells, Th17 T cells, and Th1 T cells, are connected in the underlying biological process. Moreover, the cluster-level representation can be crude because there can often be additional transition states between cell types that are not included in the clustering [24]. Frequently, the differentiation is a slow, gradual process with an indefinite number of intermediate states [25], and the process is most accurately modeled at the single-cell level. A significant number of trajectory inference methods have been developed to address these limitations, which enable modeling the cells in more depth as a trajectory [26].

In **Publications III** and **IV** of this thesis, we introduced two new trajectory inference methods, scShaper and Totem, that aim to improve or fill caveats in the current state-of-the-art trajectory inference methods [27; 28; 29; 30]. The first method, scShaper, is an ensemble method for linear trajectory inference that combines multiple weak trajectory models to create a robust, fast, and accurate model. scShaper generates the single models by optimizing a linear path through a graph of which nodes are clusters computed using the *k*-means algorithm. The path optimization is performed using a degree-constrained Kruskal's algorithm. The main aim was to develop an accurate, general-purpose algorithm for inferring linear paths through data that would work better than the principal curves algorithm [31], which is a method that is widely used in trajectory inference to infer paths through trajectory lineages [28].

The second trajectory inference method, Totem, was designed to provide an easy-to-use, intuitive interface to trajectory optimization. Although many trajectory inference methods have been developed [26], easily exceeding one hundred at the moment, the generalizability of these methods is rather weak, meaning there are no one-size-fits-all methods that work in every dataset. Therefore, to find an optimal trajectory for a single-cell dataset, a lot of parameter tuning and method testing are likely to be needed. Totem addresses this issue by generating a large number of clustering results that are used to construct a catalog of minimum spanning trees (MST), from which the user can select trajectories for further analysis that are biologically sensible. To facilitate the clustering-selection process, we developed a metric called cell connectivity, which helps to visually locate transition states and branching points from two-dimensional embeddings such as *t*-SNE. To thoroughly benchmark scShaper and Totem, we used a benchmarking framework that consists of hundreds of datasets, which were of both simulated and real origin [26].

1.1 Aims

The main aim of this thesis was to develop new computational methods for cell type identification and trajectory inference from scRNA-seq data that improve the current state-of-the-art methods. The second aim was to benchmark methods for DE analysis of scRNA-seq data. The more specific aims of this thesis can be described as follows:

1. Develop a method for scRNA-seq clustering that can segregate cell types with subtle transcriptomic differences

2. Benchmark methods cell-type-specific DE analysis of multi-subject, multi-condition scRNA-seq data

3. Develop a method for trajectory inference that is robust, fast, and performs accurately with linear trajectories of varying complexity

4. Develop a method for inferring tree-shaped trajectories from scRNA-seq data that is user-friendly and flexible

5. Make the benchmarking codes available for everyone to reproduce the analyses

6. Release the novel computational methods as free, open-source R packages

1.2 Content

Chapter 2 of the thesis briefly explains the background of scRNA-seq as a technology and the basic steps and goals of scRNA-seq data analysis. Since cell type identification, trajectory inference, and DE analysis are our main focus in this thesis, **Chapter 2** primarily focuses on them. **Chapter 3** describes how the three new computational methods introduced in **Publications I**, **III**, and **IV** work. **Chapter 4** goes through the computational methods, data, and performance evaluation methods that were used in the works. **Chapter 5** presents the main results. In **Chapter 6**, we discuss the novelty and scientific importance of this thesis, the main limitations of it, what could have been improved, and future prospects of the new computational methods. Finally, we summarize the four publications in **Chapter 7**.

2 Background

This chapter describes the background of single-cell RNA sequencing (scRNA-seq). We begin by explaining what RNA sequencing is, its applications, and the main differences between the bulk RNA-seq technology, which measures the transcriptome of a population of cells, and the scRNA-seq technology, which measures the transcriptome of individual cells. Later, we go through the basic steps of scRNA-seq data analysis. Since the number of scRNA-seq protocols that can generate scRNA-seq data is large [3], and the analysis practices vary between the protocols, we focus here on droplet-based data, specifically 10X Chromium data [5], which was the primary data type in the original publications of this thesis. Regarding the data analysis steps discussed, we emphasize more differential expression (DE) analysis, trajectory inference, and cell type identification, which are the main themes of this thesis. Other relevant steps, namely pre-processing, normalization, quality control, and joint analysis of multiple datasets, are discussed briefly.

2.1 Single-cell RNA sequencing technology

RNA sequencing (RNA-seq) is a technology that is based on the general concept of next-generation sequencing (NGS), sometimes also called massive parallel sequencing, which is a broad term for sequencing technologies that enable fast, parallel sequencing of millions of DNA fragments [1]. The NGS technologies differ by the type of DNA fragments that are sequenced. For example, in RNA-seq, the sequenced DNA fragments are complementary DNA (cDNA) fragments that are reverse transcribed from RNA fragments. Other commonly used NGS technologies include whole-genome sequencing (WGS), which sequences DNA fragments of the whole genome [32]; whole exome sequencing (WES), which sequences DNA fragments of the exome [33], i.e. the protein-coding parts of the genome; and chromatin immunoprecipitation sequencing (ChIP-seq), which uses the ChIP technology to extract certain proteins that are linked to DNA and sequences the extracted DNA [34].

The collection of RNA present in a cell or a population of cells is called the transcriptome, which comprises several types of RNA. When a cell performs protein synthesis to synthesize a protein from a DNA segment (gene), the gene is first transcribed into a messenger RNA (mRNA) and then translated into a protein. Measuring levels of different mRNA sequences present in a cell or a population of cells enables quantifying gene expression, one of the main applications of RNA-seq and the only one that we focus on in this thesis. To mention a few other essential RNA-seq applications, RNA-seq can also be used to study alternative splicing of pre-mRNAs [35; 36] or for variant discovery to detect single nucleotide variants or other mutations [37; 38]. Besides being able to quantify the expression of protein-coding genes, RNA-seq can also measure the expression of non-coding RNAs (ncRNA) that are not translated into proteins [39; 40].

When an RNA-seq experiment is performed to measure mRNA expression, the lab personnel must first prepare the cDNA library, which consists of cDNA copied from mRNA through the process of reverse transcription. The process begins with the extraction of mRNAs from the transcriptome, which involves the isolation of the RNAs that have the poly(A) tail at the 3' end of the RNA, enabling to separate mRNAs from ribosomal (rRNAs) and transfer RNAs (tRNAs) that lack the tail. The reverse transcriptase enzyme is used to generate the cDNA from the filtered mRNA, which is usually amplified using the DNA polymerase enzyme [41; 42].

When RNA-seq and other NGS technologies were first introduced, they were used to study the transcriptome of populations of cells that comprise thousands or millions of cells. These RNA-seq experiments are now commonly referred to as bulk RNA-seq experiments to distinguish them from single-cell RNA-seq (scRNA-seq) experiments, which measure the expression of individual cells. When bulk RNA-seq experiments are performed, the researcher must carefully plan which cell types to include in the sequenced sample. This step requires lab validation to ensure that the correct cell types are being investigated and is prone to errors. In contrast, in scRNA-seq experiments, the cells can be identified by their gene expression in the analysis step, which is generally accurate as long as the gene expression signal is strong enough to segregate the cell types.

Because each cell has its own set of transcriptomic processes, it is more informative to measure the transcriptome of all cells individually than as a single, mutual signal of all the cells. However, one common issue with scRNA-seq technologies is that they often cannot measure the whole transcriptome. This issue is mainly related to many droplet-based systems, such as 10X Chromium, which prioritize the cell count over the sequencing depth [3] and can often capture only the highly expressed genes. Furthermore, the 10X Chromium system can sequence only one end of the transcript (5' or 3') but not both of them at the same time. In contrast, plate-based technologies such as Smart-seq2 [4] prioritize the sequencing depth over the cell count. High-depth sequencing is beneficial because it enables more accurate detection of lowly expressed genes, whereas a higher cell number allows more accurate detection of rare cell types. However, a high cell number also improves the probability of capturing weaker signals of the lowly expressed genes, and it also increases the statistical power (sensitivity) in differential expression analysis, which is used study gene expression differences between cell populations [43].

2.2 Analysis of single-cell RNA sequencing data

This section gives a brief overview of the main steps of scRNA-seq data analysis. We begin by describing the pre-processing of scRNA-seq data, which involves the steps needed to generate the gene expression count matrix from the raw sequencing images. In the next part, we go through downstream analysis steps that are required to transform the gene expression counts into actual knowledge that researchers can utilize to make findings. These steps include quality control, normalization, cell type identification, differential expression analysis, visualization, joint analysis of multiple datasets, and trajectory inference.

In recent years, the complexity of scRNA-seq analysis has grown substantially [44], and therefore we do not discuss all essential topics. We leave out important topics such as multimodal data analysis, which consist of additional modalities (data layers) alongside RNA-seq [45; 46; 47; 48] and many downstream analysis steps, such as pathway analysis [49; 50], regulatory network inference [51], cell abundance analysis [52; 53], and ligand-target analysis [54].

2.2.1 Pre-processing

RNA-seq data analysis comprises a multitude of steps, from the analysis of the raw sequencing images to the downstream analysis of gene counts. The downstream analysis is commonly referred to as the analysis phase after the initial, standardized pre-processing steps that generate the gene expression count matrix [55]. Pre-processing steps that are common in all RNA-seq analyses include the generation of the FASTQ files from the raw image files generated by the sequencing instrument, such as the BCL files by Illumina sequencers [56], quality control to ensure adequate data quality, read alignment to a reference genome to determine the genomic location of each read [57; 58; 59], and read assignment to a reference annotation to generate the gene expression count matrix from the aligned reads [49].

The pre-processing of scRNA-seq data includes several notable differences compared to bulk RNA-seq data. Since the cDNA library of scRNA-seq data is a pooled mixture of cDNAs from multiple cells, the cDNA reads need to be mapped (demultiplexed) to their parent cells based on DNA barcodes (cell barcodes) added to the transcript-coding sequences in the lab preparation step. The cell barcode is typically at least 16 bases in length and is often accompanied by a unique molecular identifier (UMI) barcode, which is unique for each transcript [60]. The UMI technology aims to remove the confounding effect caused by the amplification step that generates copies of the cDNAs randomly and causes a skewed distribution of the transcripts compared to the original transcriptome. The UMI also helps combat the overabundance of zero measurements in scRNA-seq data [61], that is, zero inflation. The scRNA-seq systems that utilize the UMI technology, such as Chromium by 10X

Genomics, will instead count the UMIs, and the counts are hence commonly referred to as UMI counts.

Pre-processing pipelines such as Cellranger by 10X Genomics count the cell and UMI barcodes for all the sequenced reads and assign the reads to cells and genes to generate the cell-gene UMI count matrix. Initially, the counting is performed for every possible cell barcode that the protocol allows, which comprises millions of cell barcodes. The counting is based on measuring the similarity of two sequences using distance methods such as the Levenshtein distance [62]. The counting is sensitive to sequence errors, and thus, a few mismatches in the bases are typically allowed [63]. Finally, the cells are ordered into a descending order based on the sum of the UMI counts, and the positive cells are selected from the left side of the elbow point, a point at which the UMI count sum will begin to decline sharply. The cells on the right side of the curve are considered background from empty droplets and excluded. Pre-processing pipelines such as Cellranger generate helpful summaries that automatically alert quality issues. For example, for Chromium v3 data, the total number of reads per cell should be close to 50,000 reads, or at minimum 20,000 [5].

2.2.2 Downstream analysis

While the pre-processing steps are mostly standardized for each scRNA-seq protocol, the downstream analysis that uses the pre-processed cell-gene expression count matrix involves many steps that can be customized in numerous ways [44; 45]. The downstream analysis steps are specifically tailored for each dataset depending on the researcher's questions and aims for the data. For this reason, it is also the most challenging part of scRNA-seq data analysis. In this section, we describe the steps that are practically always included in downstream analysis: quality control, normalization, cell type identification, visualization, differential expression analysis, and joint analysis of multiple datasets. In addition, we go through the background of trajectory inference, which is one of the main topics of this thesis besides differential expression analysis and cell type identification.

Quality control

The first step is always quality control in which problematic cells and genes are filtered out [64]. These include cells with a high proportion of reads mapped to the mitochondrial genes, typically above 5-10%. However, the optimal threshold can also vary between tissues, and therefore caution is recommended when removing any cells. Outlier cells with an abnormally high or low number of expressed genes or UMI counts are usually discarded. These cell filtering steps involve visual inspection of the distribution of the quality parameters with violin plots and are adjusted individually for each scRNA-seq dataset. Furthermore, there exists doublet detection tools that can be used to automatically detect heterotypic doublets, which are droplets that have gene expression signal from two different cell types [65].

A good practice is to identify the cells that are being removed based on clustering and gene markers and determine for each identified cell type individually if the cell type should be discarded. It can sometimes be a good idea to perform the quality control for each sample separately if the quality attributes differ between the samples substantially [66].

Compared to the cell filtering, gene filtering is more standardized, and non-expressing genes or genes that are expressed in only a small number cells (e.g., fewer than 3) are typically removed.

Normalization

After quality control, the next necessary step is normalization, which aims to simultaneously remove errors caused by technical factors in the preparation step of a scRNA-seq experiment and preserve the underlying biological signal in the count data [67]. Normalization methods used in bulk RNA-seq, such as trimmed mean of M values (TMM) [68], and the counts per million (CPM), are generally not as such applicable to scRNA-seq normalization [69; 70]. In widely used scRNA-seq toolkits, such as Seurat [9] and Scanpy [8], the default normalization method is LogNormalize, which divides the counts by the sum of the counts for each cell and multiplies the ratios by a scaling factor, which is usually 10,000 for UMI counts. The method closely relates to the CPM bulk RNA-seq normalization method. However, its scaling factor is instead million because bulk RNA-seq measures amplified counts, and its sequencing depth is higher than in UMI counts. As in bulk RNA-seq normalization, the logarithmic transformation is commonly applied, with a pseudo-count value (usually 1) added to the normalized counts before the transformation.

sctransform is another widely used normalization method, which builds a negative binomial regression model, models the cell sequencing depth as a covariate in a generalized linear model, and uses Pearson's residuals of the regression model as the normalized data [67]. In contrast to LogNormalize, sctransform does not require pseudo-count addition or log transformation. scran [69] and SCnorm [70] are normalization methods that have been developed explicitly for scRNA-seq data and demonstrated to outperform bulk RNA-seq normalization methods in an independent study [71]. Overall, the range of scRNA-seq normalization methods that bioinformaticians actively use is relatively limited.

Cell type identification

Identification of cell types is another essential step that is performed in every scRNA-seq data analysis. In some rare cases, the cell type identities can already be available before the analysis if a plate-based system is used [4; 72]. However, in droplet-based systems, such as 10X Chromium, the cell identities are always unknown and need to be identified based on the gene expression signal. Two main approaches exist for cell type identification. The first is the unsupervised approach that uses clustering to segregate the cells into clusters, followed by statistical analysis to find gene markers that are differentially expressed between the clusters. The process requires a person who can interpret the gene markers to assign the gene markers to the cell types.

The second, more automatized approach uses supervised learning to predict the cell types with a model that has been trained using reference data [73; 74; 75; 76]. The reference data is typically from an older, independent study, which includes a cell type annotation similar to the query data of which cells are being annotated. This approach is convenient because it removes the need to find the correct clusters and interpret their gene markers. However, even the supervised models cannot be guaranteed to work accurately, and validation with gene markers is strongly recommended. In addition, the supervised annotation methods can generally only annotate cell types that are also present in the reference data. Many annotators also claim to be able to automatically predict novel cell types, which is easier the more deviant the new cell type is at the transcriptomic level compared to the reference cell types. Therefore, the two approaches are not really meant to be used mutually exclusively but together to compensate each other and facilitate cell type identification.

The unsupervised approach to cell type identification involves multiple steps before clustering. Clustering algorithms such as the Louvain community detection [77] are not optimal for datasets with tens of thousands of features. In addition to the high running time and memory requirement for such large datasets, it also becomes difficult to accurately measure distances between cells that are close to each other at the transcriptomic level. In machine learning, this phenomenon is known as ”the curse of dimensionality” [10]. To mitigate the issue, developers have implemented steps into their pipelines that reduce the dimensionality before clustering. The two ways to reduce the dimensionality are feature selection, which selects a subset of the original features, and feature extraction, which transforms the features into a smaller set of new features that aim to maintain the cell distances of the original data. In scRNA-seq data analysis, it is common to use both approaches.

The popular Seurat scRNA-seq analysis toolkit selects highly variable genes (HVG) using a local regression (LOESS) model, which adjusts the feature variance so that genes that are expressed in rare cells are not underrepresented in the selection, scales the selected feature to unit variance, and then applies principal component analysis (PCA) on the scaled features. Typically, the number of features in the PCA-transformed data matrix varies from 5 to 50, which is selected based on the elbow plot that visualizes the variance of each principal component (PC) or the jackstraw method that measures the statistical significance of each PC.

In clustering, the cells of the dimensionally reduced data matrix are grouped into clusters (communities) using a clustering algorithm. The graph-based clustering algorithms, such as Louvain [77] and Leiden [78], are popular because they are used in Seurat [9] and Scanpy [8], the two by far most popular scRNA-seq data analysis toolkits. The graph-based clustering algorithms differ from many commonly used clustering algorithms, such as *k*-means, *k*-medoids, and the Gaussian Mixture Model (GMM), because they do not generate a pre-defined number of clusters. Instead, the algorithms infer an optimal clustering number for each dataset, which is higher for datasets with more distinct communities. However, the resolution parameter, which is a positive number typically between 0.2 and 2.0, can be increased to increase the probability of finding more clusters. SC3 [11] is a consensus clustering algorithm that was initially popular when the scRNA-datasets were smaller but has now been largely superseded by graph-based clustering algorithms with better scalability.

Visualization

To visualize scRNA-seq data so that it accurately represents the cell heterogeneity of the dataset, the dimensionally reduced matrix is further transformed into a two-dimensional representation, or sometimes three-dimensional, using non-linear dimensionality reduction methods, most commonly *t*-distributed stochastic neighbour embedding (*t*-SNE) [79] or Uniform Manifold Approximation and Projection (UMAP) [80]. The features of the transformed data matrix (embeddings) are visualized as a scatter plot, which provides a general overview of the cell types that are identifiable in the dataset. The visualization is useful for assessing whether the clustering needs to be adjusted to include a different cell population composition, to study relationships between different covariates, such as time point or individual, or to visualize the expression of marker genes, which facilitates cell type identification. While *t*-SNE and UMAP are the most popular visualization methods in scRNA-seq data analysis, autoencoders [81] can also be used to visualize data [82]. However, it is more common to use the latent variables of an autoencoder as input to *t*-SNE or UMAP [83; 84], in which case the autoencoders are only used to perform data integration and other analysis steps required before visualization.

Gene marker discovery and differential expression analysis

To find gene markers for the cell clusters, we need to compare the expression levels of each gene between the cell types. The most common approach is the one-vs-rest approach, in which the expression levels of one cell type are compared with the expression levels of the rest of the cells, i.e., other cell types combined. The comparison involves statistical testing using methods that test some hypothesis, such as whether the mean ranks of two populations differ (Wilcoxon rank-sum test). In the Seurat toolkit, the Wilcoxon rank-sum test is the default method, but it also includes several other methods, such as DESeq2 and Student's *t*-test, which are common in DE analysis of bulk RNA-seq data. MAST [21] is an example of a method specifically designed for single-cell data and has become widely used. When finding gene markers for cell types, it is common to include only markers that are positively expressed compared to the other cell types (positive markers) and discard markers that are negatively expressed (negative markers). Similarly, genes that are expressed in only a tiny proportion of the cells in either population (less than 10%) or have a low logarithmic fold-change ($\texttt{logFC} < 0.25$), i.e., the logarithm-transformed ratio of the two means, are usually discarded to accelerate the analysis. If the normalized dataset is log-transformed, logFC is simply the difference between the two population means.

Since the gene marker discovery involves testing tens of thousands of genes, the *p*-values must be adjusted for multiple comparisons to decrease the number of false positive findings. Seurat and Scanpy use the Bonferroni correction [85] because the adjusted *p*-values stay constant if the gene filtering criteria are changed prior to testing, assuming the total number of expressed genes is always used to correct the values. The Bonferroni correction is a conservative correction method, meaning it can more effectively reduce false positive findings (type II error) than other correction methods, such as the Benjamini–Hochberg procedure [86]. However, its ability to reduce false negative findings (type I error) is low compared to the Benjamini–Hochberg procedure. When changing the correction method from Bonferroni to Benjamini–Hochberg, it must be kept in mind that the adjusted *p*-values can change depending on how the genes are filtered.

The process of DE analysis between any two cell populations is virtually the same as for the gene marker discovery. However, if the objective is not to find gene markers to identify cell types, it is also relevant to consider the genes with a negative logFC in the comparison. Furthermore, there are special considerations if the dataset includes cells from multiple subjects or batches. The issues of joint analysis of multiple datasets are discussed in the next subsection.

Joint analysis of multiple datasets

When scRNA-seq was introduced, the first datasets were mainly prepared using a single cDNA library, replicate, and condition. However, researchers soon began to create more complex scRNA-seq experiments that included multiple subjects and different covariates, such as conditions, time points, age, and sex. Around the same time, there also started to be a growing need to compare datasets between studies.

When joint analysis of multiple datasets is performed, the covariates often create a hierarchical structure in the data in which the cells with similar covariate characteristics are clustered closer together. This can sometimes be a favourable outcome if the cells are clustered based on a condition central to the study's aims, which implies that the condition is causing changes in gene expression. However, in many cases, cells being clustered based on a covariate is harmful because it hinders the unsupervised cell type identification that is performed using clustering, and the DE analysis that is performed within each cell type based on the clustering. Researchers have developed data integration methods to mitigate this issue by removing batch effects prior to clustering and visualization [9; 87; 88]. The data integration methods aim to transform the data so that the cell types are clustered based on cell types, not batches. Data integration is practical even in cases where the segregation of cell types based on a covariate is favourable to demonstrate a hypothesis because it facilitates DE analysis within cell types.

As in unsupervised cell type identification, the DE analysis is also more complicated when a scRNA-seq experiment includes multiple subjects or covariates that cause batch effects in the data. Basic statistical tests such as the Wilcoxon rank-sum test assume that the data points in both comparison populations are statistically independent. However, this assumption is only rarely valid in experiments that include cells originating from multiple subjects or other biological replicates. In statistics, the issue of dependent observations in populations is known as the pseudoreplication bias [89], which can cause false positive findings [16]. To alleviate the pseudoreplication bias in DE analysis of scRNA-seq data, researchers have introduced methods [17; 18; 19; 22] that account for the subjects in the DE analysis model. The two main approaches for multi-subject scRNA-seq data analysis are the pseudo-bulk methods that aggregate the counts at the subject level and the mixed models that model the subjects as a random effect. In **Publication II**, we benchmarked various methods for DE analysis of multi-condition, multi-subject scRNA-seq data.

Trajectory inference

While the discrete cluster-level representation of the cells is helpful for applications such as cell type identification and cell-type-specific DE analysis, it does not utilize the full potential of the single-cell resolution. Cells can have additional transition states [24] that are not part of the clustering, and the clustering does not model how the cell types differentiate from one another. Sometimes cell differentiation can be a gradual process with many intermediate states, and the differentiation is most accurately modelled at the single-cell level.

To model dynamic processes with scRNA-seq data, many trajectory inference methods, also known as pseudotemporal ordering methods, have been developed [26], the current number easily exceeding hundred. A single-cell trajectory is a dynamical representation of the cells that models how the cells transition between states at the discrete or continuous level, and pseudotime is the measure of cell differentiation in the trajectory, a pseudotime of 0 being the starting point and the highest possible pseudotime being the endpoint [90]. Trajectories can have different shapes, also called topologies [27], the simplest one being linear or cycle, and the most complex ones having multiple disconnected parts and multiple branching points at which cells diverge. Trees represent a topology type in which all cell types are connected without forming cycles or disconnected parts. If a trajectory has multiple endpoints to which cells diverge, it is said to diverge. In contrast, if it has multiple starting points from which cells converge, the trajectory is said to converge.

Since the number of trajectory inference methods is so large, it is not straightforward to provide a comprehensive summary of them all. In general, all trajectory inference methods require pre-processing steps that reduce the dimensionality, e.g., from 10,000 to 3, prior to the trajectory construction, while aiming to preserve the cell heterogeneity after the transformation. The dimensionality reduction can be performed similarly as in cell type identification, using feature selection and feature extraction. Another shared feature of trajectory inference methods is the requirement of a user-specified startpoint, a cell or cluster, from which the pseudotime is calculated.

Monocle [90] was one of the earliest trajectory inference methods for scRNA-seq data, and it models the trajectory as a minimum spanning tree (MST) of cells. Its successor, Monocle 2 [91], utilizes the reversed graph embedding (RGE) algorithm [92] to enable more robust and accurate trajectory inference compared to the first version of Monocle. Monocle 3 [93] was inspired by PAGA [94], which is a method that uses *k*-nearest neighbour (*k*-NN) graphs and the Louvain community detection algorithm to create a trajectory in which connected, neighbouring Louvain communities (clusters) have more neighbouring cells in the kNN graph than would be expected under a statistical model. The three versions of Monocle and PAGA are among the most widely used and cited trajectory inference methods.

Slingshot [28] is another widely used trajectory inference method. It builds a minimum spanning tree (MST) for a clustering to create a milestone network that models how the milestones (clusters) differentiate from each other in the trajectory. In the second phase, Slingshot uses the simultaneous principal curves algorithm [31; 28] to infer lineages with respect to a user-specified starting cluster, creating single-cell resolution pseudotime in the process. Slingshot allows the user to freely decide the input clustering, but automated clustering optimization methods such as the average silhouette width (ASW) [95] can be used as well [26].

The methods have significant differences in terms of the topology of the trajectory that can be modelled. Slingshot is limited to diverging tree-shaped trajectories, which can also be disconnected after a post-publication update, whereas Monocle 3 and PAGA can be used to model trajectories that have cycles and both converging and diverging parts [96]. scShaper is a trajectory inference method that was introduced in **Publication III**, and it is limited to linear trajectories.

A significant limitation of the above-mentioned trajectory inference methods is that they require the starting cell or cluster to be specified by the user. However, RNA velocity methods [97; 98] overcome this limitation by leveraging information about spliced and unspliced mRNA to infer the direction of the trajectory. While the RNA velocity methods have been largely successful, their usage can be challenging with cell types relatively close to each other at the transcriptomic level, such as immune cells. In addition, the RNA velocity methods do not generally work with multi-subject, multi-condition data [99]. In contrast, trajectory inference methods that do not estimate RNA velocity can easily be combined with data integration methods [55].

Compared to cell type identification methods, trajectory inference methods can be challenging to use because the topologies for which they have been designed vary, and the default parameters do not often work optimally with every dataset [26]. Method testing and parameter tuning can be time-consuming, requiring biological knowledge to select the method and parameter combination that provides the most optimal result. To facilitate trajectory optimization, helpful frameworks such as dyno [26] have been developed, which provide a user-friendly interface for selecting an appropriate method. In **Publication IV**, we proposed our solution to this challenge by introducing Totem, a trajectory inference method that aims to simplify the inference of tree-shaped trajectories by leveraging a large set of dissimilar clustering results. Totem models each clustering as an MST and estimates the cell connectivity based on the connectivity of the clusters in the MSTs, which helps to visually locate transition states present in the data. The cell connectivity combined with a user-friendly interface enables efficient trajectory optimization in a way that does not require in-depth knowledge of the underlying methodology.

After a trajectory model has been built, DE analysis can be performed to identify genes that change along cell pseudotime. This task differs from the DE analysis of clusters that compares two sets of expression values in which the observations (cells) are unordered. Generalized additive models (GAM) are a popular approach to performing DE analysis along pseudotime [90; 98; 100]. The dynverse pipeline for trajectory inference analysis [26] includes a method that ranks the genes by their feature importance score based on a trained random forest regression model. switchDE is a method that uses a likelihood ratio test for a sigmoidal expression model to find monotonic trends [101], and ImpulseDE2 has a similar operating principle, but it can also capture non-monotonic trends [102]. scGTM is a recently introduced model that can model versatile gene expression trends more interpretably compared to the previous methods [103].

3 Computational methods

In this chapter, we briefly describe how the three novel computational methods introduced in **Publications I**, **III** and **IV** work.

3.1 ILoReg

In **Publication I**, we introduced ILoReg, a cell type identification method developed for detecting cell types with subtle transcriptomic differences from scRNA-seq data. Here we refer to cell type identification as the process that includes all or almost all steps required to identify cell types from scRNA-seq data: quality control to filter poor-quality cells and lowly expressed genes, normalization, dimensionality reduction, clustering, visualization, and gene marker discovery. The process is unsupervised, meaning the cell types are identified without external scRNA-seq data with a cell type annotation, which is the pre-requisite of supervised cell annotation methods [9; 73; 74; 104].

Figure 1 illustrates the cell type identification workflow of ILoReg. ILoReg assumes that quality control and normalization have been performed beforehand using external software, such as Seurat [9; 23]. The key difference compared to other cell type identification tools is the iterative clustering projection (ICP) step (**Figure 1a**), which is applied on a normalized gene expression matrix before principal component analysis (PCA). In scRNA-seq analysis toolkits such as Seurat, this step involves the selection of highly variable genes (HVG), usually between 1000 and 3000 genes. The purpose of the HVG selection is to reduce noise prior to PCA and decrease run time [105].

The ICP algorithm (**Figure 1a**) is an iterative algorithm that seeks a clustering of cells that has a high predictability when trained and projected using a logistic regression (LR) model, with the training performed using a subset of the whole dataset. Starting from random cluster labels, S, ICP creates a balanced training set that has the same number of cells, $n = \lceil Nd/k \rceil$, in each cluster, with N denoting the number of cells in the whole dataset, k the number of clusters, and d the hyperparameter that controls the size of the training set with respect to the whole dataset (by default, $k = 15$, $d = 0.3$).

The ICP algorithm trains an LR model [106] using the training set and predicts the cluster labels of the whole dataset, yielding the projected clustering S'. Cluster-

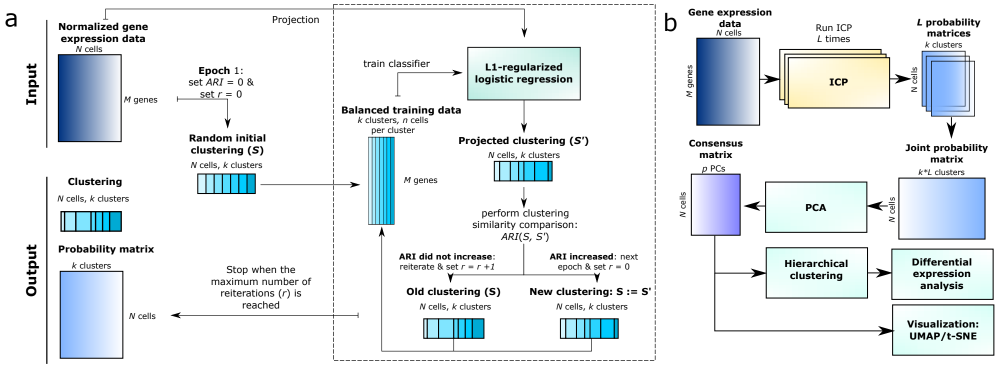

Figure 1. Process diagrams of (a) Iterative Clustering Projection (ICP) and (b) ILoReg. PCA = Principal Component Analysis. ARI = Adjusted Rand Index. UMAP = Uniform Manifold Approximation and Projection. *t*-SNE = *t*-distributed Stochastic Neighbor Embedding. Adapted from **Publication I**.

ing similarity of S and S' is assessed using the adjusted Rand index (ARI) [107], for which 1 implies perfect similarity and 0 perfect dissimilarity. If ARI increases from the starting value of 0, the clustering S is replaced by the projected clustering S', and the iteration continues with the updated clustering. Otherwise, the iteration, with the random training sampling step included, will be repeated r times (by default, $r = 5$) with the old clustering, and the clustering S and its LR-estimated probability matrix are returned as output when the number of reiterations reaches r. Every time the clustering S is updated because ARI increases, the reiteration counter is reset to zero.

The fourth hyperparameter of the ICP algorithm is the cost of constraints parameter (C) of the LR model, which controls the trade-off between the training accuracy and the stringency of L1-regularized feature selection, with $C = 0.3$ being the default value and a lower value increasing the stringency, i.e., fewer features are selected.

To increase the robustness of the ICP algorithm to its hyperparameters and reduce the impact of the stochastic training sampling on the result, we developed a consensus (ensemble) algorithm (**Figure 1b**) that aggregates the ICP probabilities from L ICP runs using PCA, where each ICP run is initialized using a different random seed. Finally, ILoReg clusters the $N \times p$-dimensional PCA-transformed data matrix using Ward's method for hierarchical clustering and visualizes it using t-distributed stochastic neighbor embedding (t-SNE) or uniform manifold approximation and projection (UMAP). The number of principal components (p) can be selected using the elbow plot, which visualizes the standard deviation of each principal component [9; 8]. Finally, the gene markers are identified using the Wilcoxon rank-sum test and the one-versus-rest approach, in which cells of each cluster are compared with the cells of the rest of the clusters [9; 8].

The rationale behind the ICP algorithm is to obtain a set of cluster labels that produce a well-generalizable LR model, that is, a model that can predict the correct labels when trained with only a subset of the data. In supervised learning, the fundamental aim is to train a classification or regression model with a subset of the data and optimize it to generalize to new, independent observations. In this regard, the processes resemble each other. Although the ICP algorithm is an unsupervised clustering method, the discrete cluster labels are, in the end, not required because only the cluster probabilities are used in the consensus method. Therefore, the ICP algorithm can be considered a soft, fuzzy clustering method. The ICP algorithm also bears a resemblance to autoencoders [82; 83; 108; 109; 110; 111; 112; 113] , which learn a small number of new features from data, i.e., the latent layer features. Similarly, the autoencoder activation functions can also be LR models.

3.2 scShaper

scShaper is a fast, robust, and accurate method for inferring linear paths through single-cell data. The principle under which it operates is based on the graph theory. scShaper generates a set of dissimilar clustering results using the *k*-means algorithm and aims to find a path through each clustering that minimizes the weights between the neighboring clusters (**Figure 2**). When the path-optimized labels replace the initial cluster labels, we obtain discrete pseudotime in which the numeric cluster labels and the path are correlated. Since finding the globally optimal path is an NP-hard problem, solving it requires a brute-force search, which is highly inefficient to compute. Therefore, we utilize a greedy algorithm inspired by Kruskal's algorithm for finding the minimum spanning tree (MST). By limiting the maximum degree of the graph to two, we obtain the degree-constrained Kruskal's algorithm. which can be used for path finding. However, unlike Kruskal's algorithm for finding MSTs, the degree-constrained Kruskal's algorithm is sensitive to the input sequence of the clusters. Therefore, the result can change depending on the order in which the clusters are inputted to the algorithm.

To obtain a trajectory that is insensitive to the input sequence of the clusters, continuous instead of discrete, and less sensitive to the choice of the number of clusters in *k*-means, we developed an ensemble (consensus) method that aggregates multiple sets of discrete pseudotime into a consensus solution. By default, scShaper clusters the data matrix 99 times, with the number of clusters (*k*) varying from 2 to 100, and then aggregates the discrete pseudotime sets by performing PCA and selecting the discrete pseudotime sets that have higher PCA loadings for the first principal component than the second component. scShaper scales the selected discrete pseudotimes by min-max scaling, flips the pseudotimes based on the signs of the variables in the PCA loadings, and averages the selected pseudotimes. The aggregation approach provided better overall performance than using the first principal component directly

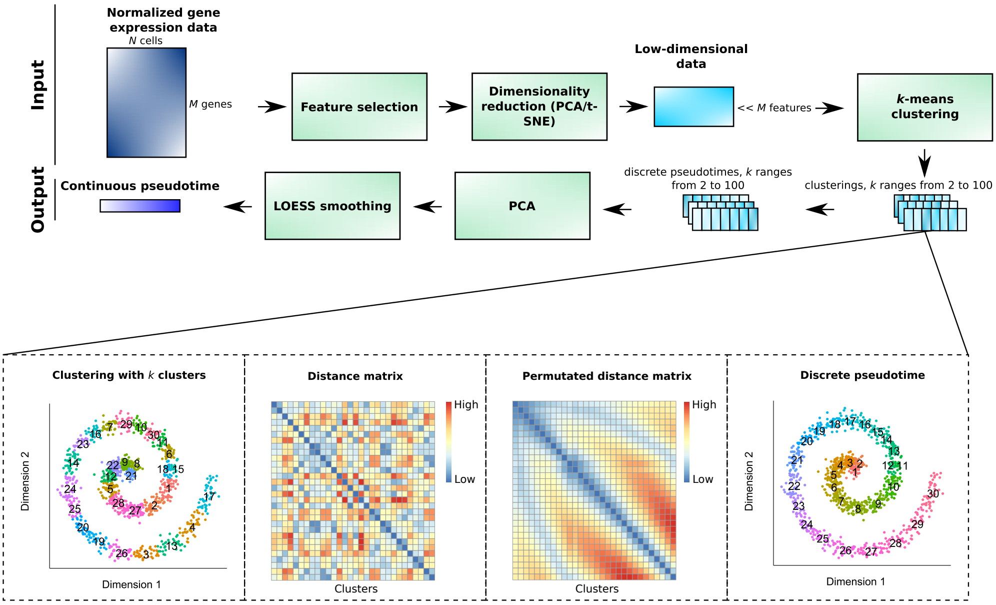

Figure 2. Process diagram of scShaper. PCA = Principal Component Analysis. *t*-SNE = *t*-distributed Stochastic Neighbor Embedding. LOESS = Locally Estimated Scatterplot Smoothing. *k* denotes the number of clusters. Adapted from **Publication III**.

(**Supplementary Fig. 3** of **Publication III**). Finally, scShaper applies LOESS (Locally Estimated Scatterplot Smoothing) to perform smoothing on the scaled and averaged pseudotime.

Trajectory inference requires the same pre-processing steps that are performed before clustering and visualization. The gene expression count data must be filtered to remove poor-quality cells and lowly expressed genes. Normalization is required to reduce skewness caused by the varying library sizes between cells. In addition, dimensionality reduction steps are needed for the same reasons as in cell type identification: to improve run time and mitigate ”the curse of dimensionality”. The upstream analysis steps can be customized as the user sees best, but, by default, scShaper performs dimensionality reduction using PCA with 50 components and *t*-SNE with three components.

3.3 Totem

A comprehensive comparison of trajectory inference methods [26] suggested that trajectory inference methods have weak generalizability; that is, the methods do not perform accurately with every dataset. Therefore, the user will likely need to test several different tools and adjust their parameters to obtain accurate trajectories. The testing process can be arduous for users unfamiliar with the underlying methodology.

To facilitate the optimization process, we developed a new trajectory inference method, Totem, which enables user-friendly inference of tree-shaped trajectories from single-cell data. **Figure 3** visualizes the workflow of Totem. Like scShaper, Totem assumes that the pre-processing steps, i.e., QC, normalization, HVG selection, and feature selection, have been performed beforehand, with the only exception being the feature extraction, which can also be performed using the Totem R package.

Totem generates a large set of dissimilar clustering results (by default, 10,000) using a k-medoids algorithm, CLARA [114], and models the differentiation network (milestone network [26]) of each clustering as a minimum spanning tree (MST). The number of clusters (k) ranges from 3 to 20, by default, and can be adjusted by the user. The MST is obtained by finding a tree that minimizes the sum of the edge weights, where the weights are the distances between the clusters calculated using a Mahalanobis-like distance metric [115; 28]. For each MST, Totem calculates connectivity of each cluster by counting the number of edges each cluster has in the MST graph and dividing it by the number of clusters. The cluster-level connectivity is scaled by the maximum connectivity of the MST, and the scaled cluster-level connectivity is transformed into cell-level connectivity values based on the cluster membership of each cell. Finally, cell connectivity is obtained by averaging all cell-level connectivity vectors across the different MSTs using the arithmetic mean.

The cell connectivity helps to give a general overview of the milestone transitions that are present in the trajectory. For example, it can be used to compare the topologies between different dimensionality reduction methods, such as PCA, multidimensional scaling (MDS) or t-SNE, which could not otherwise be compared due to their high dimensionality. With the cell connectivity as a guidance, the user can browse the MSTs and choose the ones for further analysis that accurately model the transition states and are biologically sensible. Totem selects the most optimal MST using the variance ratio criterion (VRC), with the cell connectivity used as input to it.

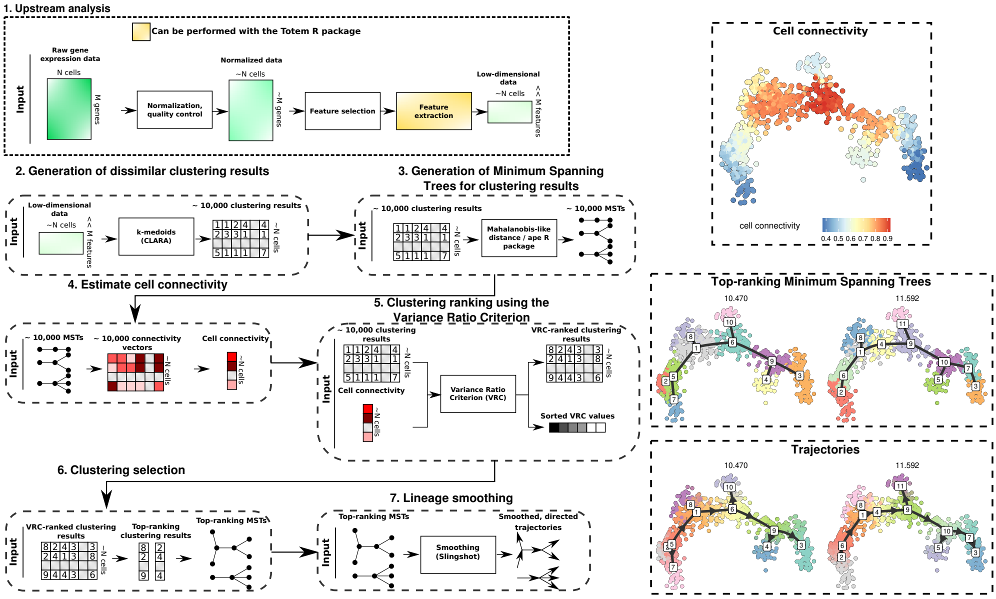

Figure 3. Process diagram of Totem.

4 Materials

In this chapter, we go through the materials that were used in all publications of this thesis, including the computational methods that were compared, the benchmark data, and the performance evaluation methods.

4.1 Benchmarked computational methods

In this section, we briefly describe the computational methods that were benchmarked with the novel computational methods introduced in the publications of this thesis.

4.1.1 Cell type identification methods

An extensive number of cell type identification methods have been developed, which enable identification of cell types in an unsupervised manner from scRNA-seq data based on gene marker signatures of cell clusters [10; 116; 117]. In addition to ILoReg, our method introduced in **Publication I**, we considered four additional cell type identification methods (CIDR [12], RaceID3 [14], SC3 [11], and Seurat [9]), which were popular among the scRNA-seq research community at the time, and especially Seurat continues to be. The methodologies of these methods vary considerably in terms of the pre-processing steps performed prior to clustering (**Table 1**). To mention a few notable differences, CIDR is the only method that uses imputation to replace missing values (dropouts), a step that is still controversial among researchers as to whether it brings any significant benefit to the analysis of scRNA-seq data [71; 118], and the method for selecting the optimal number of clusters is different for each cell type identification method.

4.1.2 Differential state detection methods

In **Publication II**, we benchmarked methods for detecting differential states (DS) between conditions (e.g., knock-out versus wild-type or sick versus healthy) from scRNA-seq data, which include multiple subjects or other biological replicates per condition. The methods that were compared in the publication can be divided into two main categories (**Figure 3**): pseudo-bulk methods and single-cell methods.

Table 1. Properties of the cell type identification methods compared in **Publication I**. *SC3 is a clustering algorithm, but *t*-SNE and UMAP can be used to visualize using the scater R package [119]. ICP = Iterative Clustering Projection. PCA = Principal Component Analysis. MDS = Multi-Dimensional Scaling. CSPA = Cluster-based Similarity Partitioning Algorithm. kNN = *k*-nearest neighbors. *t*-SNE = *t*-distributed stochastic neighbor embedding. UMAP = Uniform Manifold Approximation and Projection. VRC = Variance Ratio Criterion.

	ILoReg	**CIDR**	**RaceID3**	**SC3**	**Seurat**
Feature selection	L1-regularization in ICP	-	HVG	HVG	HVG
Feature extraction	ICP + PCA	MDS	-	Three distance matrices + PCA, Laplacian eigenmap	PCA
Other preprocessing steps	-	imputation	random-forest-based reclassification for a separate outlier detection step	-	-
Clustering method	hierarchical	hierarchical	*k*-medoids	*k*-means + CSPA + hierarchical	graph-based (Louvain)
Visualization method	*t*-SNE, UMAP	MDS	*t*-SNE, kNN graph	none*	*t*-SNE, UMAP
Method for selecting the number of clusters	silhouette	VRC	saturation	random matrix theory	resolution-controlled
Reference	[120]	[12]	[14]	[10]	[9]

The pseudo-bulk [17; 19; 20; 121] methods aggregate the gene expression data within each cell type (cluster) and subject (or other biological replicate) by either averaging the counts that have been normalized at the single-cell level (mean aggregation) or by summing the raw counts and then applying bulk normalization (sum aggregation). After the aggregation, the pseudo-bulk methods use statistical tests from bulk RNA-seq analysis, such as Limma [122] or ROTS [123], to perform the differential expression analysis.

The single-cell methods include naïve methods that do not model the subjects in any way, as well as mixed models [18; 19; 22; 93] that model the subjects as a random effect. The naïve methods comprise classical statistical tests, such as the Wilcoxon rank-sum test, and DS analysis developed explicitly for scRNA-seq data, such as the MAST two-part hurdle model [21]. An additional single-cell method category (”Other” in **Figure 3**) includes methods that were primarily designed as batch effect correction methods, such as ComBat [124] or Seurat’s latent variable models [9], and so far, there has not been supportive evidence about their applicability to multi-subject DS analysis [18; 125].

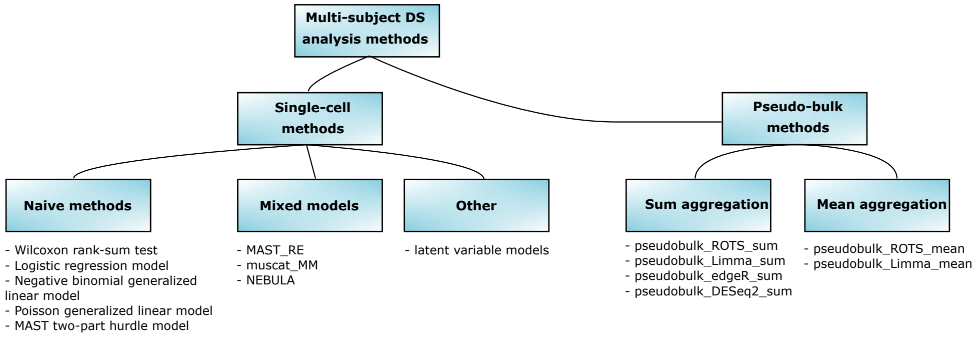

Figure 4. Overview of the differential state (DS) detection methods compared in **Publication II**.

4.1.3 Trajectory inference methods

From the large number of trajectory inference methods that have been published, we included methods that performed well in a benchmark study by Saelens et al. [26]. To benchmark scShaper, which is a method introduced in **Publication III** for linear trajectory inference, we considered the best-performing methods in the comparison that are restricted to linear trajectories, i.e., Comp 1 [26], Elpilinear [29], Embeddr [126], and SCORPIUS [30]. Comp 1 uses the first principal component as pseudotime. In contrast, the other three methods use the principal curves algorithm [31] or the elastic principal graphs algorithm [127] to perform the pseudotime estimation (**Table 2**). In both **Publications III** and **IV**, we also considered the popular Slingshot method [28], which is a method for inferring tree-shaped trajectories based on a user-provided clustering and a low-dimensional embedding. Slingshot models the differentiation network as an MST and estimates pseudotime using the simultaneous principal curves algorithm. We also considered TinGa [27], which is a more recently introduced method that utilizes the growing neural gas (GNG) algorithm [128] and can also infer trajectories that have cycles and disconnected parts.

Table 2. Properties of the trajectory inference methods compared in **Publications III** and **IV**. In Publication **IV**, we compared only Slingshot, TinGa, and Totem. *Elpilinear uses the elastic principal graphs algorithm, which is related to the principal curves algorithm. **SCORPIUS, Slingshot, and Totem use a clustering to infer the milestone network, whereas scShaper uses an ensemble of clustering results to estimate pseudotime. ***All methods except Elpilinear and Embeddr can be used with any dimensionally reduced data matrix. PCA = Principal Component Analysis. MDS = Multi-Dimensional Scaling. *t*-SNE = *t*-distributed stochastic neighbor embedding. LMDS = Landscape MDS.

	Comp 1	**Elpilinear**	**Embeddr**	**SCORPIUS**	**scShaper**	**Slingshot**	**TinGa**	**Totem**
Restricted to linear trajectories	yes	yes	yes	yes	yes	no	no	no
Uses principal curves *	no	no	yes	yes	no	yes	no	yes
Clustering-based **	no	no	no	yes	yes	yes	no	yes
Default feature extraction ***	PCA	PCA	Laplacian eigenmaps	MDS	PCA + *t*-SNE	PCA	LMDS	LMDS
Reference	[26]	[29]	[126]	[30]	[129]	[28]	[27]	[130]

4.2 Benchmark data

Each work included datasets that were used to benchmark the novel computational methods (**Publications I**, **III**, and **IV**) and the methods for differential state detection (**Publication II**). All datasets were either simulated or acquired from public databases.

4.2.1 Benchmark data for cell type identification

To compare the ability of ILoReg (**Publication I**) and the other cell type identification methods to visually identify cell populations from scRNA-seq data based on a two-dimensional embedding plot, such as *t*-SNE or UMAP, we used a peripheral blood mononuclear cell (PBMC) dataset (pbmc3k) that includes 3,000 cells generated using the version 1 of Chromium by 10X Genomics [5]. The pbmc3k dataset is widely used in tutorials to showcase scRNA-seq analysis methods, such as Seurat [9] and Scanpy [8]. In addition, we considered a second public dataset, which was originally extracted from human pancreatic tissue (named Baron1) [15], consisting of 2,000 cells. The Baron1 dataset included a cell type annotation created by the original study's authors, and the pbmc3k dataset included an annotation provided by the developers of the Seurat toolkit.

To benchmark ILoReg for clustering of scRNA-seq data, we considered 11 datasets from three different studies [15; 72; 131]. The Pollen dataset was the only gold-standard dataset, meaning the cell labels were known prior to the sequencing based on laboratory experiments. The remaining ten datasets were silver-standard, meaning the cell type labels were used as they were identified in the original studies based on clustering.

4.2.2 Benchmark data for differential state detection

Our comparison of methods for DS detection between conditions from multi-subject scRNA-seq data (**Publication II**) included synthetic data generated based on real scRNA-seq data as a reference. Data generated with reference-based simulation [19; 89] is meant to be more realistic than data simulated without reference. This can be achieved by modeling characteristics of individual genes and samples from the reference data. These characteristics include the mean expression of genes, library sizes of cells, and dispersion at the sample (subject) level. The simulator can then generate new multi-subject, multi-condition data that include the same characteristics as the reference, and ground truth on which genes have differential states.

To perform reference-based simulation, we used the muscat R package [19], which can simulate genes with different DS types: changes in mean expression (DE), changes in modality (DM), changes in proportions of low and high expression parts (DP), and changes in both modality and proportions (DB). The differential modality (DM) means that a dataset has a different number of expression peaks between groups, and they overlap at least partially. When a gene has both differential proportions and modality (DB), the peaks do not overlap, and their number differs. The muscat simulator uses a negative binomial generative model to simulate count data and requires control scRNA-seq samples as input. As the reference data for the muscat simulator, we used control samples from four studies [17; 132; 133; 134]. If the number of simulated subjects per group (condition) exceeds the number of control samples in the reference, the extra samples will be technical replicates from the biological replicates. Therefore, the number of simulated subjects per condition (group) was kept, at maximum, at the number of control samples, except when we performed an additional analysis in which we investigated the impact of the number of samples on the performance. Overall, the number of simulated subjects in both case-control groups varied from four to ten. logFC between the case-control groups varied from 0.5 to 1.25 for the genes with differential states, generating genes that have relatively subtle changes between conditions. In total, we simulated 54 datasets (clusters, cell types) with the reference-based simulation approach.

In addition to the reference-based simulation, we performed a simulation that does not utilize reference data [22]. As in the reference-based simulation, the reference-free simulation adds between-subject and between-cell variance into the data. However, the magnitudes of these effects are not estimated from reference data but set manually. In total, we generated 1280 datasets by varying the two variance parameters, the number of samples per case-control group, the number of cells, and the average expression. The genes with a logFC of 0 were classified as not having differential states (negative), and those genes with a logFC between 0.5 and 2 had a differential state (positive).

4.2.3 Benchmark data for trajectory inference

For the Totem and scShaper trajectory inference tools (**Publication II** and **IV**), we used the dynverse environment [26], which comprises a versatile set of R packages for running trajectory inference methods, analyzing their results, and benchmarking. The Zenodo data repository [135] of dynverse comprises close to 300 datasets in total, from which we used 69 datasets with a linear trajectory to benchmark scShaper [129] and 216 datasets with a tree-shaped trajectory to benchmark Totem [130], which includes the 69 datasets with a linear trajectory. The synthetic datasets were simulated using PROSSTT [136] (19 datasets), Splatter [137] (35 datasets), dyngen [138] (30 datasets), and dyntoy [138] (52 datasets) tools. In addition to the synthetic data, the benchmark data also included real data (80 datasets) with either silver-standard ground truth (54 datasets), meaning the cell types and their trajectories were based on earlier annotations from published works, or gold-standard ground truth (26 datasets), meaning the cell types were known prior to the sequencing, and their differentiation order is known with high certainty. In all gold-standard datasets, as well as some of the silver-standard datasets, the trajectories had only discrete pseudotime, which models the differentiation at the cell type level. However, some silver-standard datasets also had continuous, single-cell resolution pseudotime generated using a trajectory inference method.

4.3 Performance evaluation

4.3.1 Clustering performance

To measure clustering performance in **Publication I**, we used the adjusted Rand index (ARI) [107], which has been widely used in scRNA-seq studies to evaluate clustering performance [11; 12; 116; 117]. ARI is a modification the normal Rand index and adjusts the result for chance.

4.3.2 Performance evaluation of differential state detection methods

Simulated data

When performing DS detection between two cell populations, a gene can be seen as having one of two possible states: the gene either has a differential state (positive), or it does not have one (negative). Therefore, benchmarking DS detection methods involves comparing the actual, ground-truth state, i.e., the state we obtain from simulation or somehow else, with the state that is computationally predicted using a method. The comparison is performed for many genes, typically thousands, meaning the comparison is performed between two large binary variable lists. In this type of binary classification task, we can use binary classification performance metrics [139], which can be derived from a 2×2 contingency table (confusion matrix) that counts the true positives (TP), true negatives (TN), false negatives (FN), and false positives (FP) for all genes between the two lists. In **Publication II**, we used sensitivity (recall, power, true positive rate), specificity (true negative rate), precision, F1 score, and Matthew's correlation coefficient (MCC).

Sensitivity measures what proportion of the true positives were identified as positives ($TP/(TP + FN)$), which can be maximized by detecting all genes as positives. Specificity, its counterpart, measures what proportion of the true negatives were identified as negatives ($TN/(TN + FP)$), and it can similarly be maximized by not reporting any positives.

Precision measures what proportion of all positive findings were true positives ($TP/(TP+FP)$), and its drawback is that it cannot be calculated if no positives are detected. Notably, we obtain the false discovery rate (FDR) by subtracting precision from one, the metric controlled by the Benjamini-Hochberg procedure for correcting p-values for multiple testing, which was used in **Publication II**. Comparison of the estimated FDR based on how many false discoveries are found in benchmarking and the expected FDR based on the FDR cutoff enables to determine whether a method is too conservative, i.e., it produces fewer false discoveries than expected, or if it is too permissive, i.e., it produces more false discoveries than expected. We compared each method's estimated and expected FDR levels based on the results of the two simulations.

The F1 score is the harmonic mean of sensitivity and precision, providing a more general-level assessment of the performance than the two metrics alone. The MCC is another metric for evaluating overall performance. It has been suggested to be less negatively affected than the F1 score and accuracy when the class labels are imbalanced [140], a property that is commonly present in RNA-seq data and was also in our benchmark data.

In differential state detection, an intricate part is defining when a gene has a differential state. In RNA-seq data analysis, the definition is typically based upon the

p-value of a gene, or more often its value after adjustment for multiple comparisons [85; 86], and its log-transformed fold-change (logFC), i.e., the log-transformed ratio of the average expression between the two groups. In **Publication II**, we defined a gene as a positive if its $FDR \leq 0.05$. In practice, the p-value criterion is often accompanied by a logFC threshold (by default, 0.25 in Seurat) to exclude genes with only a tiny difference in the average expression between two groups. However, we decided to neglect the logFC filtering step to avoid removing weaker signals that may be decisive in determining the superiority of the methods.

A single p-value or FDR cutoff to determine positive and negative discoveries may not be conclusive enough to compare methods because the values of the binary classification metrics change depending on which cutoff is used. Therefore, it is common to perform an analysis that tests different cutoffs, calculates their corresponding metric values, and calculates a new metric that summarizes the values. The receiver operating characteristic (ROC) curve is the most widely used method to perform this type of multi-threshold performance evaluation. ROC uses multiple p-value cutoffs to calculate sensitivity and false positive rate, specificity subtracted from one, creates a curve from the values, and calculates the area under the curve (AUROC). To perform ROC analysis in **Publication II**, we used the pROC R package [141]. Alternatively, the precision-recall curve can be more suitable than ROC if a dataset has a high proportion of negatives and the positive instances are rare [142; 143; 144]. However, we did not use the precision-recall method in our work.

Mock comparison to estimate false positive rate

Besides simulation, we performed a mock comparison [145] to estimate the false positive rate (FPR), which is one minus specificity, by creating randomly assigned groups from 14 control subjects of the Liu dataset [134], and by calculating the ratio of the number of positive (false positive) findings to the number of negative (true negative) and positive findings. Using the uncorrected p-values to define the positive and negative findings, we estimated the type I error (false positives) control by comparing the estimated and expected FPR levels [146]. This is based on the notion that with a p-value of 0.05, we can expect a 5% chance that the null hypothesis was rejected incorrectly (type I error). The interpretation is similar to the FDR control analysis based on the precision from the simulation, i.e. those methods that exceed the FPR are overly permissive, and those that fall below it are overly conservative.

Reproducibility

As the final analysis in **Publication II**, we assessed reproducibility of the DS detection by comparing how similar the results between different subsets of the same dataset are. We created 100 random subsets of the Liu [134] case-control samples that were generated based on the reference-based simulation by the muscat R package. To measure reproducibility, we calculated Spearman's rank correlation coefficient between every dataset pair using the p-values.

4.3.3 Performance evaluation of trajectory inference methods

We used the dynverse trajectory inference framework [26] to benchmark trajectory inference methods in **Publications III** and **IV**. In total, dynverse comprises 17 performance evaluation metrics, but only four were used to assess the overall performance in the original study by Saelens et al. [26] and our studies.

The first of the four metrics is the correlation between geodesic distances. In the two trajectories that are compared, i.e., the ground-truth and inferred trajectories, the geodesic distance is calculated between every possible cell pair in both trajectories. The geodesic distance is a distance method that can be easily applied to milestone network models that have regions of delayed commitment. The regions of delayed commitment are trajectory regions in which a lineage diverges into multiple new lineages (bifurcating or multifurcating), and the cells that are within the regions can simultaneously have non-zero progression along more than one of the diverging lineages, which is, in contrast to linear regions in which cells are only progressing between two milestones (cell types). After the pairwise geodesic distances have been calculated in both trajectories, Pearson's correlation coefficient is calculated between the two geodesic distance lists.

The second main metric is the accuracy of differentially expressed features. dynverse uses the ranger software [147] to build a random forest (RF) regression model and assess the feature importance of each feature (gene) using the trained model. As the predictor variable that the RF model aims to predict, dynverse uses the geodesic distance, which is calculated from each cell to all milestones. A separate RF model is trained for each milestone, and the feature importance scores are averaged over the milestones. The Pearson's correlation coefficient is calculated between the ground-truth and computationally inferred average feature importance lists. To place more weight on features that are ranked higher in the ground-truth feature importance list, dynverse calculates the weighted correlation by weighting the features by their feature importance scores in the ground-truth feature importance list.

The third metric is the Hamming-Ipsen-Mikhailov distance (HIM) [148], which measures the topological accuracy, i.e., how similar the milestone network graphs are in terms of structure. The two compared trajectories are represented as adjacency matrices in which the edges are weighted based on the edge weights in the milestone network of the trajectories, which is usually the Euclidean or Mahalanobis distance between the milestone centroids. The HIM is a linear combination of the normalized Hamming distance and the normalized Ipsen-Mikhailov distance calculated between two graphs, which assess the local structural similarity and the global structural similarity of the graphs, respectively.

The fourth and final main metric is the F1 branches, which estimates accuracy of branch assignment. The cells are mapped to their nearest branches, i.e., linear segments of the trajectory between milestones, generating a discrete clustering, and clustering similarity between the ground-truth and inferred clustersets is assessed using a method based on the Jaccard index [149].

The four above-described metrics are averaged using the geometric mean, which penalizes small values. Therefore, if one of the metrics has a value close to zero and the rest of the metric values are high, the overall score will still be small. In **Publication III**, we only considered the correlation of geodesic distances and the correlation between differentially expressed features when calculating the overall score because including the two other metrics would have unfairly penalized the Slingshot and TinGa methods that can also predict non-linear trajectories. In **Publication IV**, we considered all four metrics when calculating the overall score.

5 Results

5.1 Unsupervised cell type identification using ILoReg – Publication I

To benchmark ILoReg for cell type identification from scRNA-seq data, we carried out a two-part comparison. In the first part, we evaluated the clustering accuracy of ILoReg and four other cell type identification methods: CIDR [12], RaceID3 [14], SC3 [11], and Seurat [9]. In the second part, we compared cell populations that were visually identifiable from a PBMC dataset (pbmc3k) based on two-dimensional embeddings, such as *t*-SNE, UMAP, or MDS. We performed a closer examination of the cell types that could be identified from the PBMC dataset and a pancreatic dataset using ILoReg and Seurat. In addition, we constrained the parameters of ILoReg by investigating how the parameters' adjustment affected the identifiable cell populations.

5.1.1 Benchmarking ILoReg for clustering of scRNA-seq data

To assess clustering accuracy, we used ARI [107], which measures clustering similarity between two clustersets, with 1 indicating perfect similarity and 0 indicating no similarity. For each of the 11 datasets and five clustering methods, we calculated ARI between the ground-truth and inferred clustersets. The results (**Figure 5**) suggested that ILoReg and Seurat achieved good overall performance regardless of the dataset size. SC3 performed well with the smaller datasets (Pollen, vanGalen_BM_1), but its performance was moderate for the larger datasets (Baron). Overall, CIDR performed worse than the other methods. Two of the van Galen datasets (BM_5_1 and BM_5_2) had a highly imbalanced distribution of cells between the cell types, and all methods performed poorly (ARI below 0.25) with these datasets.

5.1.2 Visualizing peripheral blood mononuclear cell populations using ILoReg

ILoReg was designed to identify cell populations from scRNA-seq data that can be difficult to identify with conventional methods such as Seurat [9], which select a set of highly variable genes (HVGs) prior to PCA. To demonstrate the ability of ILoReg to identify cell populations with subtle transcriptomic differences, we used

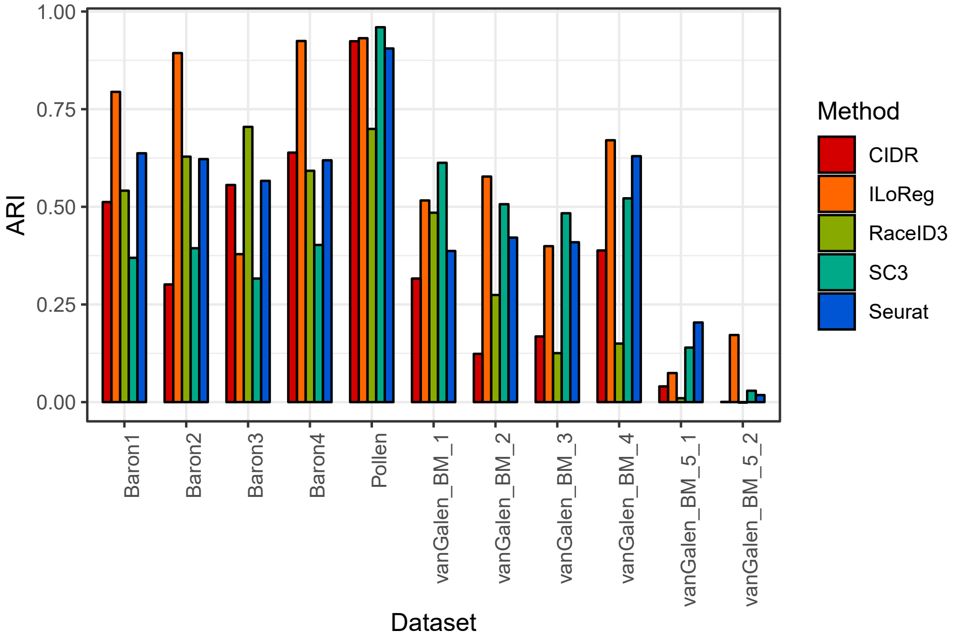

Figure 5. Evaluating clustering accuracy in **Publication I**. The benchmark data include 11 datasets from three studies. ARI = adjusted Rand index. Adapted from **Publication I**.

the pbmc3k dataset [150], which is an example dataset in tutorials of Seurat [9] and Scanpy [8]. In **Publication I**, we compared the visualizations (**Figure 5** of **Publication I**) of the five methods, which differ in the steps that are performed prior to the visualization (**Table 2**).

The results showed that ILoReg was able to identify distinct B and T cell populations that were not visible in the visualizations of the other methods. Specifically, ILoReg identified a cell population that expressed gene markers (*CCR7+*/*S100A4-*/*S100B+*) of the naïve CD8+ T cells (**Figure 4c** of **Publication I**). In addition, ILoReg identified B cell populations that expressed markers of naïve (*TCL1A+*/*CD27-*) and memory B cells (*TCL1A-*/*CD27+*) [151; 152], as well B cell subpopulations with differential expression of immunoglobulin light chain markers *IKGC*, *IGLC2*, *IGLC3*, indicating segregation of B cells with lambda and kappa light chains [153]. The *t*-SNE embeddings in **Figure 6** show the same analysis but with a more stringent feature selection ($C = 1$) and a larger training set size ($d = 0.5$). With these parameters, the small B cell subpopulations with varying expression of the light chain markers disappear, but the rest of the cell populations remain unchanged.

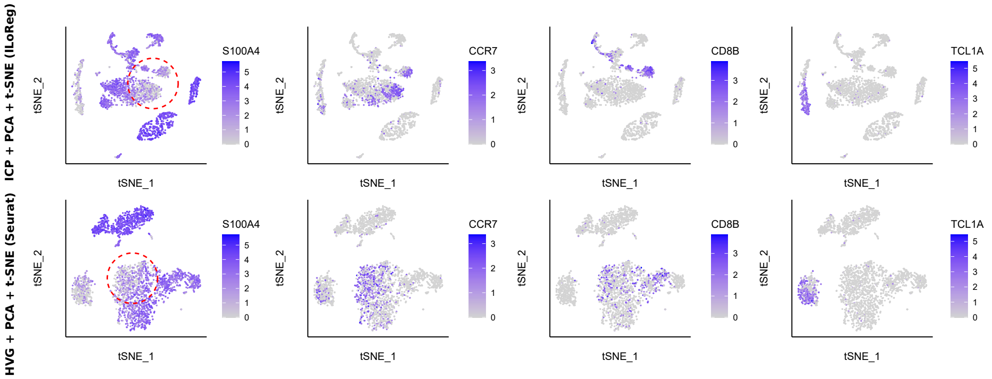

Figure 6. Comparing ILoReg and Seurat for identifying PBMC populations from *t*-SNE visualizations in **Publication I**. *S100A4-/CCR7+* indicate naïve T cells, and *S100A4+/CCR7-* indicate memory T cells. *CD8B* is a marker of CD8+ T cells, and *TCL1A* is a marker of naïve B cells. PBMC = peripheral blood mononuclear cell. ICP = iterative clustering projection. HVG = highly variable genes. PCA = principal component analysis. *t*-SNE = *t*-distributed stochastic neighbor embedding.

5.1.3 Visualizing pancreatic cell populations using ILoReg

For a human pancreatic scRNA-seq dataset (Baron1 [15]), we performed a similar comparison as for the PBMC dataset. Overall, the *t*-SNE visualization of ILoReg included more distinct cell populations than the *t*-SNE visualization of Seurat (**Figure 5** of **Publication I**). An example of such a population was the *IAPP+/MALAT1-* beta cell population, which we hypothesized to be stressed (injured) beta cells because *MALAT1* has been shown to be downregulated in injured beta cells [154]. This conclusion was supported by a functional analysis using the Metascape tool [155], which revealed enriched pathways linked to cell stress, such as endoplasmic reticulum stress [156]. **Figure 7** visualizes some of the markers that were downregulated in the injured beta cells. While the *t*-SNE visualization of Seurat included a population that had similar markers, it was clustered closely together with the rest of the beta cells. The example shows how ILoReg can more easily distinguish cell populations that are subsets of the same cell type.

5.1.4 Robustness, run time and parameters of ILoReg

The ICP is stochastic because it includes a step that selects cells randomly into the training set. To obtain robust, reproducible results with ILoReg, we developed a consensus (ensemble) method that aggregates results from multiple ICP runs (L) using PCA. **Figure 2** of **Publication I** shows that the consensus approach of ILoReg achieved more robust and better performance than single ICP runs.

The run time is an important part of algorithm performance. The ICP algorithm

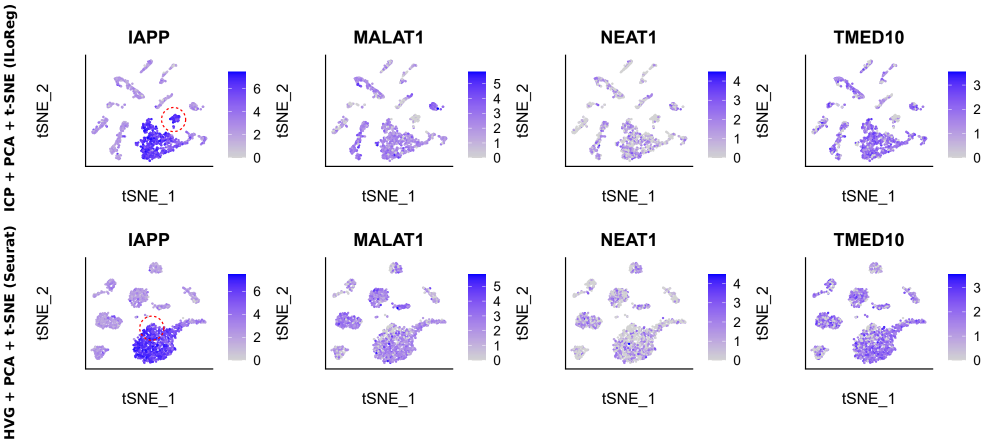

Figure 7. Comparing ILoReg and Seurat for identifying beta cell populations from *t*-SNE visualizations in **Publication I**. *IAPP* is a beta cell marker that encodes the islet amyloid polypeptide hormone *shepherd2004transcriptional*. *MALAT1*, *NEAT1* and *TMED10* are downregulated in the injured beta cells.

is the computational bottleneck of ILoReg because it requires training thousands of logistic regression models and projecting cells using the models. With the default parameters and the pbmc3k dataset that includes 3,000 cells, the run time of ILoReg was approximately one hour on a laptop with 8 GB of RAM and four logical processors (threads). When the number of cells was increased to 20,000, the run time was ten hours.

ILoReg includes six parameters (hyperparameters) that needed to be constrained to default values: 1) the number of clusters in ICP (k), 2) the proportion of cells in the training data of the LR model (d), 3) the cost of constraints (C) that regularizes the trade-off between training accuracy and feature selection in the ICP, 4) the number of reiterations (r) that affects how far the ICP algorithms converges, 5) the number of ICP runs (L), 6) the number of principal components in the PCA aggregation step (p).

For the number of clusters, $k = 15$ was set as the default value because this is, on average, close to the number of cell types that are identifiable in many tissues, such as pancreas and PBMC. Tuning the k parameter may be necessary if more complex tissues, such as mouse brain [157], are analyzed.

Figure 6 shows the same comparison as in **Figure 4c** of **Publication I**, except that the C parameter was increased to 0.3 from 1, increasing the number of selected features and decreasing the training error in the LR model. Changing the parameters did not impact the T cell populations identified by ILoReg, but the four separate B cell populations became merged into a single continuous B cell population. Based on the expression of the *TCL1A* gene, which is a marker of the naïve B cells, the upper part of the B cell population constitutes memory B cells that have differentiated from naïve B cells, which are in the lower half of the B cell population. The example shows how the C parameter can be used to control the resolution of the cell population identification, with a higher value decreasing the resolution. The difference in the resolution occurs because the algorithm becomes less sensitive to cell populations that are segregable by a small number of highly variable genes.

The d parameter controls the number of cells in the training data ($n = \lceil Nd/k \rceil$). Increasing d has a similar impact as increasing C, i.e., the resolution of the cell type identification decreases. The resolution decreases because the ICP runs become more dissimilar and the PCA aggregation has fewer correlated cluster probabilities as input.

The number of principal components (p) can be adjusted based on the elbow plot that visualizes the variance of the principal components, which is the most common approach for choosing its value. The default value of 50 was chosen because using too many principal components is generally considered to be less harmful than using too few.

For the number of ICP runs (L), we selected a high default value (200). The objective of the consensus approach is to obtain robust results, and we did not observe significant differences between the results generated using 100, 150, or 200 ICP runs. However, each dataset has its own point at which the consensus approach begins to produce results that are robust and reproducible, and hence a high default value was chosen to ensure robustness.

5.2 Benchmarking methods for detecting differential states between conditions from multi-subject scRNA-seq data – Publication III

Here we report the main results of **Publication II** in which we benchmarked methods for detecting differential states between conditions from multi-subject scRNA-seq data. The benchmark involved assessing the false positive control, sensitivity, area under the receiver operating characteristic curve (AUROC), and reproducibility. When discussing the results of **Publication II**, we use the term differential state instead of differential expression because differential expression is one of the six differential state types considered in the work. However, elsewhere in this thesis we primarily use the term differential expression because it has been widely adopted as the term for referring to changes in gene expression.

5.2.1 False positive control

Since the assignment of positive and negative findings was based on a single FDR threshold of 0.05, we can expect that 5% of the positive results are false. Comparison of the expected FDR with the estimated FDR level will enable to evaluate the FDR control [146]. No single rule exists to determine when the FDR control is accurate. However, the median of the estimated FDR should be close to the expected FDR, and the deviation from the median should be minimal. We considered the FDR accurate if the expected FDR was within the 0.25 and 0.75 quartiles of the estimated FDR and the variation was visually judging moderate.

For the simulation results, we obtained the estimated FDR by subtracting precision from one. The simulation results (**Figure 8**) showed that the pseudo-bulk methods achieved the most accurate FDR control of the different method types. In the reference-based simulation (**Figure 8b**), the estimated FDR levels of muscat_MM and NEBULA-LN indicated that their FDR control was too loose, whereas the FDR control of MAST_RE was accurate. However, in the reference-free simulation (**Figure 8a**), the FDR control was too loose for all three mixed models. In contrast, the pseudo-bulk methods achieved, in general, accurate FDR levels. The FDR control of the naive and latent methods was too loose in both simulations.

The mock analysis (**Figure 8c**) measures the false positive rate (FPR) by comparing the number of positive findings (false positive) with the number of negative findings (true negative). If we use the raw *p*-values that are uncorrected for multiple testing to define positive and negative findings, we can evaluate the FPR control (type I error control) for each method [146; 22]. The results of the mock analysis were, on average, in line with the simulation results, with the naïve and latent methods exhibiting overly high FPR levels and the pseudo-bulk methods showing accurate FPR control. However, unlike in the simulation results (**Figure 8a-b**), muscat_MM had the lowest FPR levels of the methods that reported positive findings, considerably below the expected FPR of 0.05. Furthermore, when the *p*-values were corrected for multiple testing using the Benjamini-Hockberg procedure, the FPR of all pseudo-bulk methods and mixed models was clearly below the FPR of 0.05 (**Fig. 4** of **Publication II**), with NEBULA-LN reporting more false positives than the other methods for some of the mock datasets.

5.2.2 Sensitivity and ROC analysis

Besides the false positive (type I error) control, another important aspect is the ability to detect true positives or avoid false negative findings (type II error). We measured sensitivity, i.e., what proportion of the positives in the truth set were predicted as positives (**Figure 9c,d**). The naïve methods achieved the highest overall sensitivity in both simulations. In the reference-based simulation, the naïve methods outperformed the latent methods in terms of sensitivity, but in the reference-free simulation the two method types achieved comparable levels of sensitivity. However, the pseudo-bulk methods and mixed models achieved considerably lower sensitivity than the naïve and latent methods. The sensitivity of the pseudo-bulk methods that use mean aggregation was lower than the sensitivity of the pseudo-bulk methods that use sum aggregation. ROTS achieved the lowest sensitivity of the four statistical tests. NEBULA-LN achieved the highest overall sensitivity of the three mixed models, whereas muscat_MM had the lowest sensitivity.

The receiver operating characteristic (ROC) curve is a popular method for evaluating performance in a way that is not restricted to a single cut-off that defines the positive and negative findings. It simultaneously considers both the sensitivity and the false positive ratio aspects of the performance. The area under ROC (AUROC) results for the simulation (**Figure 9a,b**) indicated that the pseudo-bulk methods outperformed the other method types. However, the pseudo-bulks that use the mean aggregation were slightly inferior to the sum aggregation methods, which we also observed from the sensitivity results (**Figure 9c,d**). Compared to sensitivity, the differences between the mixed models and pseudo-bulk methods were more pronounced in the AUROC results, with the pseudo-bulks generally outperforming the mixed models in terms of AUROC. The AUROC of the naïve methods was relatively high, suggesting that although they were more susceptible to false positives than the other method types (**Figure 9c,d**), they were still generally able to rank the genes with differential states higher than the genes without differential states. The latent models clearly had the lowest AUROC in the reference-based simulation, but in the reference-free simulation their AUROC levels were comparable with the naïve methods.

5.2.3 Reproducibility

Finally, we measured how reproducible the results of each method were between different subsets of the same dataset. We took 100 random subsets of the reference-simulated Liu dataset, which included ten replicates per group and 20,000 cells in total. We used Spearman's rank correlation coefficient to assess the correlation between the p-values. The results (**Figure 10**) show that all methods had relatively low average reproducibility, on average below 0.5. The moderate correlation is likely due to the high proportion of genes that do not have differential states, which makes the gene ranks sensitive to changes in the group composition. The pseudo-bulk method that uses ROTS for statistical testing and mean aggregation had the best average reproducibility. The latent methods achieved abnormally high reproducibility for a small proportion of the datasets due to constant p-values.

5.3 Linear trajectory inference using scShaper – Publication II

The results of **Publication III** consist of three parts. First, we benchmarked scShaper for linear trajectory inference from scRNA-seq data. Second, we compared scShaper and the principal curves algorithm [31], which is a method that is commonly used in trajectory inference to infer smooth paths through lineages (linear segments) of the trajectories with respect to a user-specified starting point [28; 30]. Finally, we investigated the impact of the parameters of scShaper on the performance and run time.

5.3.1 Benchmarking scShaper for inference of linear trajectories from scRNA-seq data

dynverse is a software environment for running and benchmarking trajectory inference methods [26]. The original benchmark data of dynverse included 69 linear trajectories, which we used to evaluate the performance scShaper and six other trajectory inference methods (Component 1 [26], Elpilinear [29], Embeddr [126], SCORPIUS [30], Slingshot [28] and TinGa [27]) in **Publication III**. dynverse includes 17 different performance evaluation metrics that are summarized in **Supplementary Table 1** of **Publication III**. Two of these metrics, the correlation of geodesic distances (accuracy of cell ordering) and the weighted correlation of feature importance (accuracy of differentially expressed genes), were averaged using the geometric mean to create the overall score.

The results suggested (**Figure 11a-c**) that scShaper achieved similar or better performance in terms of the accuracy of cell ordering but significantly better performance compared to the other methods in terms of the accuracy of differentially expressed features and the overall score (Wilcoxon signed-rank test; p-value ≤ 0.01). When grouping the overall scores by the data type (**Figure 11d**), the results indicated that scShaper outperformed the other methods for three of the four simulators (PROSSTT [136], dyngen [138], dyntoy [26]) and the real data. All methods achieved moderate performance for the Splatter-simulated data [137].

5.3.2 Comparing scShaper and the principal curves algorithm

scShaper can be used as a general-purpose method for inferring linear paths through data. To showcase this ability, we simulated two three-dimensional datasets and compared the performance of scShaper and the principal curves algorithm (**Figure 12**). The principal curves algorithm [31] is a commonly used method for inferring linear paths through data with arbitrary dimensions, and it is used by popular trajectory inference methods, such as Slingshot [28]. These examples are similar to the examples provided in **Supplementary File** of **Publication III**. Unlike the principal curves algorithm, scShaper managed to accurately infer the correct path through both datasets.

Finally, we showed that scShaper outperformed the principal curves algorithm with scRNA-seq data (**Figure 13**). We performed the same dimensionality reduction steps (50 principal components and three t-SNE dimensions) for all datasets and compared the performance of the two methods. The rationale behind this comparison was to show that scShaper performs better because of the differences in the pseudotime estimation and not because of the differences in the pre-processing steps.

5.3.3 Robustness, run time and parameters of scShaper

Unlike Kruskal's algorithm for finding a minimum spanning tree (MST), the modified Kruskal's algorithm that scShaper uses for finding a solution to the shortest Hamiltonian path problem is sensitive to the input sequence of the vertices (clusters). Therefore, the algorithm is not guaranteed to find the minimum spanning path even when the graph weights are unique. To obtain robust results with scShaper, we developed an ensemble (consensus) method that aggregates multiple discrete pseudotimes by PCA. The ensemble method showed robust performance for a spiral trajectory when we shuffled the cells and ran the workflow thousand times with different random seeds (**Supplementary Figure 2** of **Publication III**).

The run time of scShaper is largely determined by the set of cluster numbers for which the discrete pseudotime is generated. The number of edges determines the time complexity of Kruskal's algorithm, $\mathcal{O}(E \log E)$, where E is the number of edges [159]. The number of edges is the main factor because Kruskal's algorithm sorts the edges into a descending order based on their weights, and the number of edges grows quadratically with the number of clusters (vertices). When the number of clusters ranges from 2 to 100, the run time of scShaper, with dimensionality reduction excluded, is $\sim$ 2 seconds, depending on the hardware (**Supplementary Figure 1** of **Publication III**). Increasing the upper limit of the number of clusters to 200 increased the run time to $\sim$ 1 minute.

scShaper includes a few parameters that may need to be adjusted by the user when applied to different applications. Most importantly, the number of clusters needs to be increased when the complexity of the trajectories increases. Such examples would include spiral trajectories (**Figure 12**) that have more rounds. However, linear paths through single-cell data are unlikely to have this level of complexity.

5.4 Cell-connectivity-guided trajectory inference using Totem – Publication IV

The results of **Publication IV** consist of three parts. In the first part, we used the dynverse environment to benchmark Totem with other trajectory inference methods. In the second part, we investigated the utility of the cell connectivity as a metric for choosing a clustering that is used to construct the trajectory network as an MST. In the third part, we provided practical examples of how the cell connectivity can aid trajectory optimization.

5.4.1 Benchmarking Totem for trajectory inference from scRNA-seq data

To benchmark our second trajectory inference method, Totem, in **Publication IV**, we used the dynverse framework, which was also used in **Publication III** to benchmark scShaper. dynverse comprises 216 benchmark datasets with a tree-shaped trajectory, i.e. linear, bifurcation, multifurcation, or some other more complex tree. dynverse also includes 17 performance evaluation metrics that assess different performance properties, which are summarized **Supplementary Table 2** of **Publication III**. The results of the four main metrics that determine the overall performance are visualized in **Figure 14**.

The results show that Slingshot performed well in datasets with a linear trajectory (overall score, Wilcoxon signed-rank test; p-value ≤ 0.01), but its average performance for the non-linear datasets was the lowest among the three methods. Totem had the best performance for the datasets with a non-linear trajectory (Wilcoxon signed-rank test; p-value ≤ 0.01), with TinGa providing the second-best overall performance. The performance difference is mainly attributable to the topology and branch assignment accuracy. Totem achieved better topology and branch assignment accuracy than TinGa for linear trajectories, but the accuracy of feature importance for linear trajectories was comparable with Slingshot. For the datasets with a non-linear trajectory, Totem had the best performance in terms of all four performance metrics. However, there were only negligible differences in the cell ordering accuracy between the three methods.

5.4.2 Comparison of clustering selection methods

We assessed how the clustering selection method of Totem that uses the cell connectivity and VRC performed compared to three other clustering selection methods. These methods were the random selection, which ranks clustering results into a random order, ASW, and VRC, all of which use the low-dimensional embedding as input. We generated 10,000 random clustering results using the CLARA clustering algorithm for each of the 216 dynverse benchmark datasets, selected the 100 highest-ranking clusterings with each selection method, performed trajectory inference using Slingshot for each clustering, and calculated the overall score for each trajectory. We varied the number of selected trajectories and calculated the average performance of the datasets in two ways: by considering only the best-performing trajectory of the selected trajectories (**Figure 15a**) and by calculating the average performance of the selected trajectories (**Figure 15b**).

As expected, the random selection method achieved the best performance when we considered only the best-performing trajectory (**Figure 15a**). However, it had the lowest average performance (**Figure 15b**) of the methods. The selection method of Totem achieved comparable average performance with VRC, which was the best method in terms of average performance. However, it had a more sharply increasing performance curve when we considered only the best-performing trajectory. ASW was the weakest-performing selection method due to its below-average performance in both comparisons.

5.4.3 Examples of trajectory inference using Totem

We showcased the benefits of cell-connectivity-guided trajectory inference with two examples. The first example involved a simulated dataset [26; 135] with a multifurcating trajectory, i.e., one starting point and more than two endpoints to which the cells diverge from the starting point. When we used the ground-truth trajectory and the clustering that can be derived from its milestone percentages as input to Slingshot, the resulting MST incorrectly implied that the trajectory was linear (**Figure 16a**). However, when we used Totem to select a trajectory in line with the cell connectivity, we obtained a trajectory with the correct topology. The cell connectivity correctly suggested that the trajectory had one branching point, the region with highest cell connectivity, and four start or endpoints, the regions with lowest cell connectivity.

The second example was a real, unsimulated dataset from mouse thymus [160] with a bifurcating topology, i.e., two endpoints and one starting point (**Figure 16b**). While this time Slingshot was able find the correct milestone network, the cell-connectivity-guided trajectory inference of Totem also provided accurate information about the topology of the trajectory, suggesting that the dataset was bifurcating because it had one branching point and three start or endpoints.

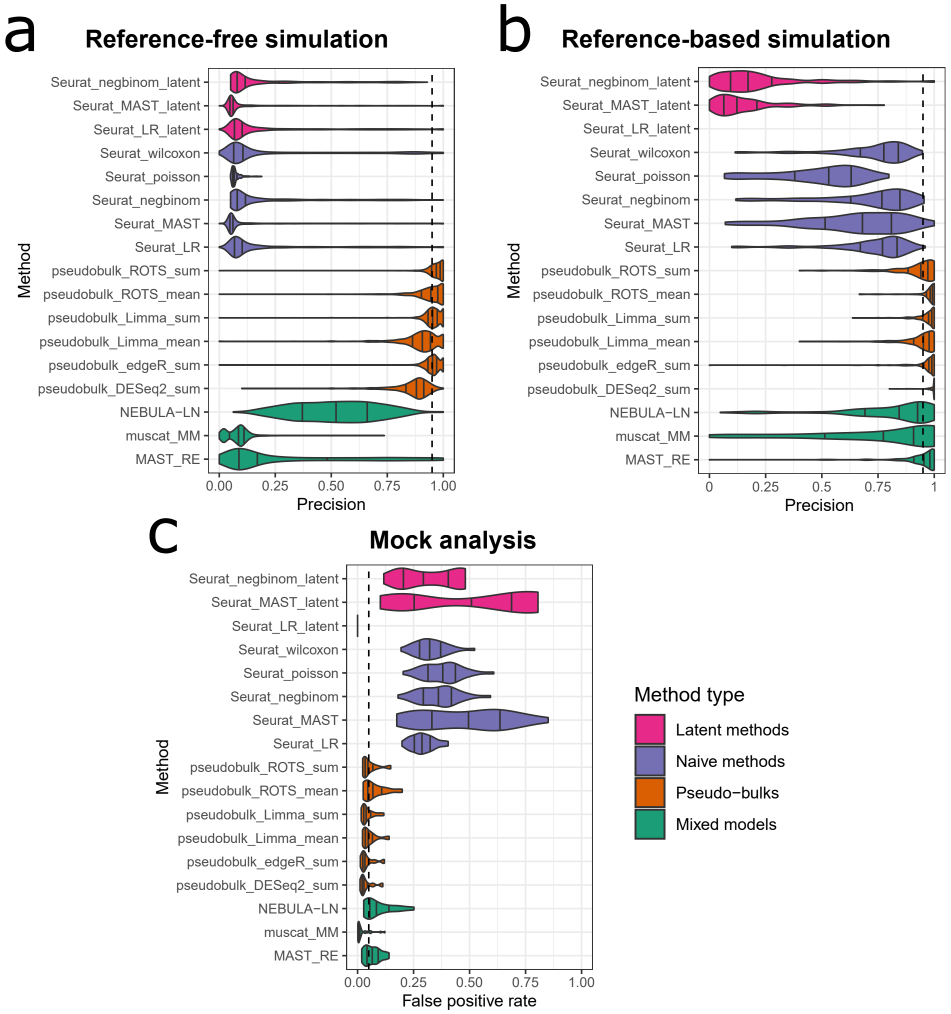

Figure 8. Precision and false positive rate (FPR) of differential detection methods in **Publication II**. (a-b) In both simulations, we used FDR=0.05 to assess FDR control by comparing the expected FDR levels (dashed, vertical line at 0.95 precision) with the estimated FDR levels. The FDR estimation was based on the Benjamini-Hochberg procedure. In the reference-based simulation, precision could not be calculated for Seurat_LR_latent because no positives were detected. (c) In mock analysis, the FPR levels of each method are compared with the expected FPR of 0.05 by considering uncorrected p-values [158; 145]. For Seurat_LR_latent method, FPR is zero for each mock comparison because no positives were detected. FDR = False Discovery Rate. FPR = False Positive Rate. Adapted from **Publication II**.

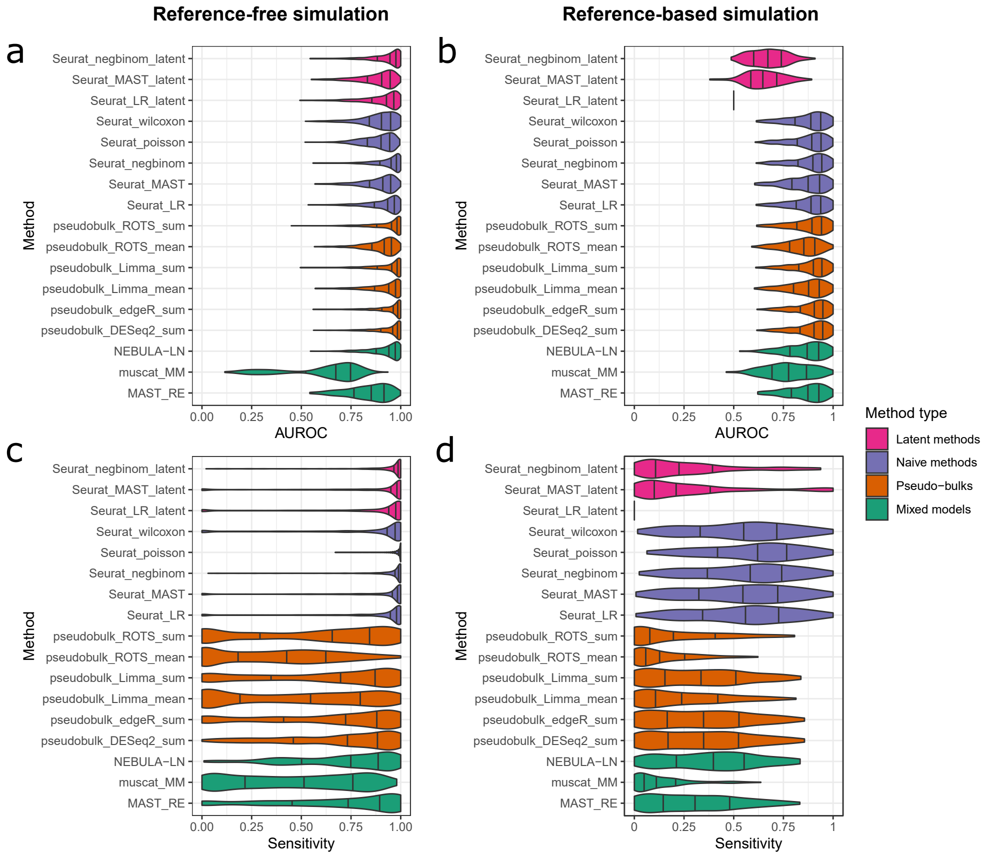

Figure 9. AUROC and sensitivity of differential state detection methods in **Publication II**. In both simulations, we used an FDR of 0.05 to define negative and positive findings. AUROC = Area Under Receiver Operating Characteristic. FDR = False Discovery Rate. Adapted from **Publication II**.

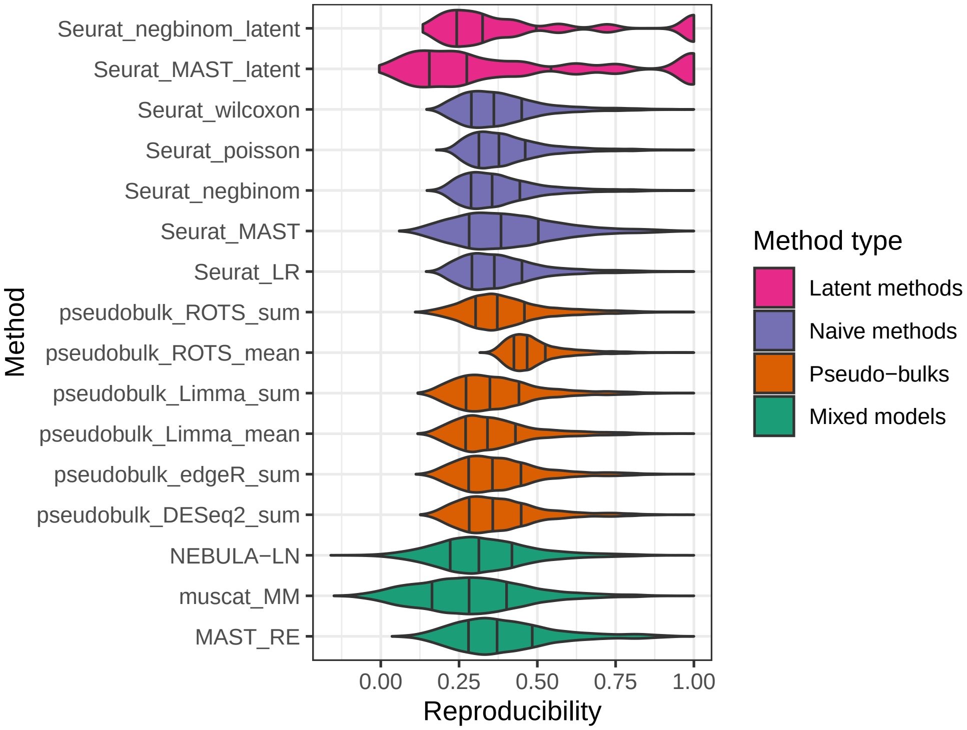

Figure 10. Reproducibility of differential state detection methods in **Publication II**. Reproducibility is Spearman's rank correlation coefficient between p-value lists obtained using 100 different subsets of a reference-simulated dataset (Liu). Adapted from **Publication II**.

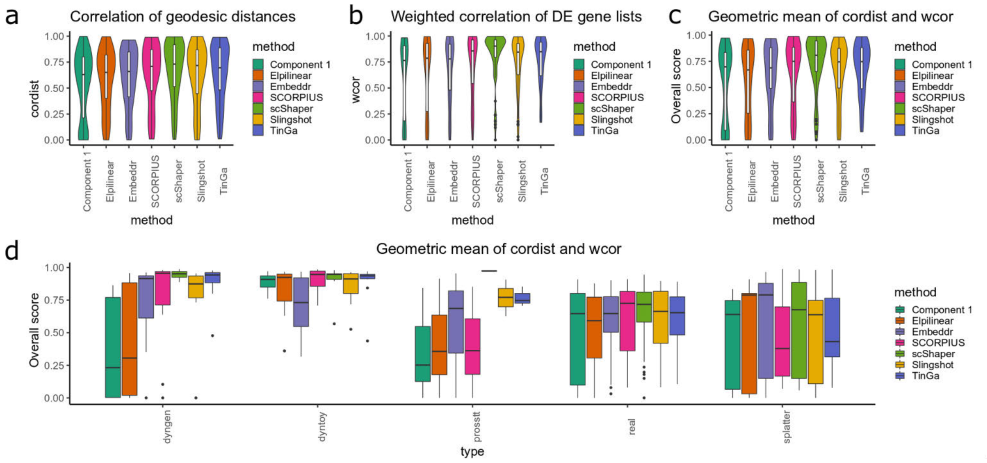

Figure 11. Benchmarking scShaper for linear trajectory inference from scRNA-seq data using dynverse framework in **Publication III**. (a) The correlation of geodesic distances (cordist) between the ground-truth and inferred trajectories, measuring accuracy of cell ordering. (b) The weighted correlation (wcor) between the feature importance lists of the ground-truth and inferred trajectories, obtained by DE analysis of the trajectories using random forest regression, measuring accuracy of DE genes. The overall score measuring the geometric mean of cordist and wcor. (d) The results grouped by data type. dyngen, dyntoy, prosstt and splatter are simulators, and real denotes real data. DE = Differential Expression. Adapted from **Publication III**.

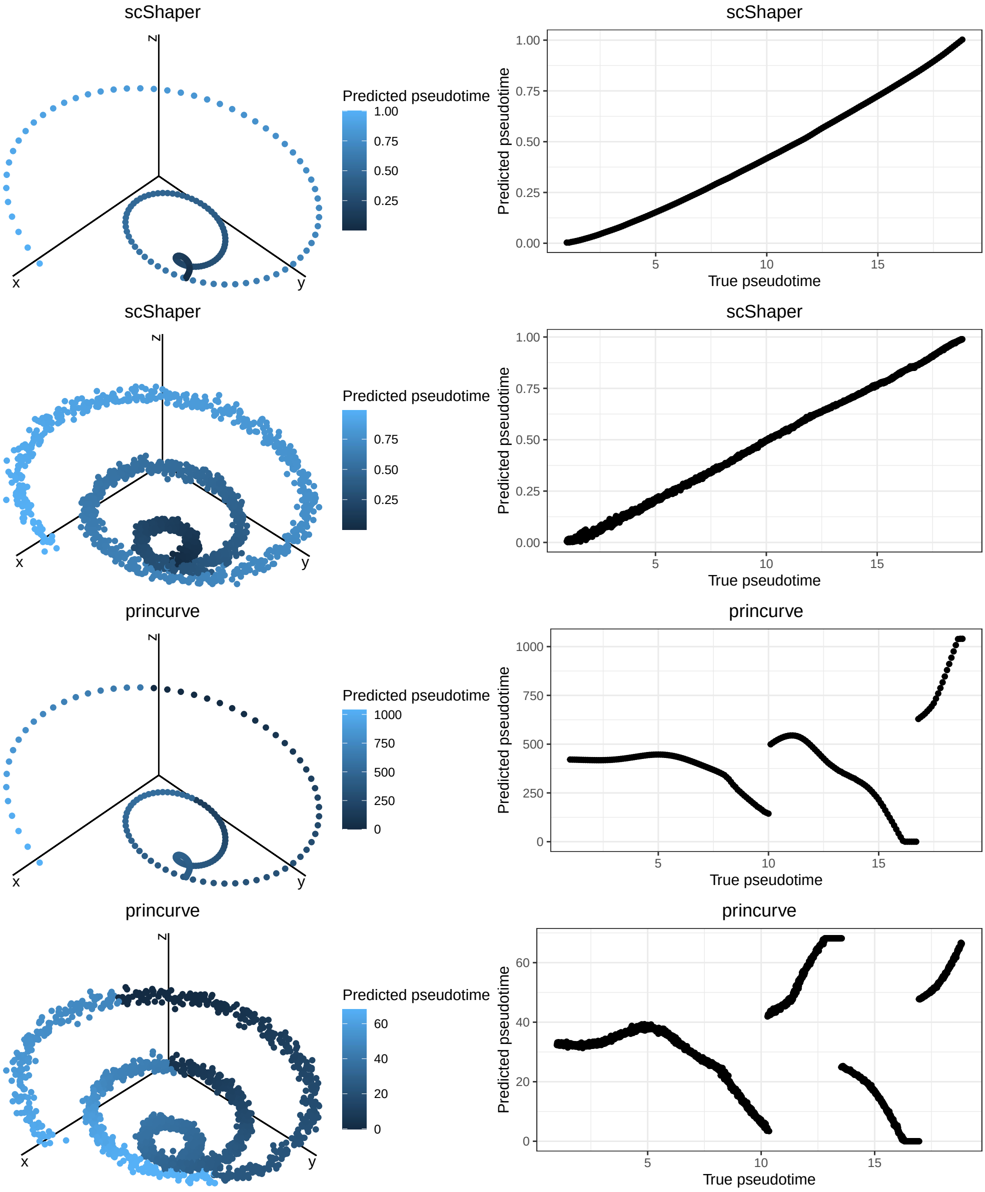

Figure 12. Inferring linear paths through spiral trajectories using scShaper and the principal curves algorithm in **Publication III**. The upper trajectory of which radius increases quadratically includes 179 data points, and the lower trajectory of which radius increases linearly includes 1785 data points. In addition, we added Gaussian noise with a standard deviation of 0.5 to the components of the linearly widening trajectory. Before scShaper analysis, the data points were randomly shuffled. Both methods were run with their default parameters.

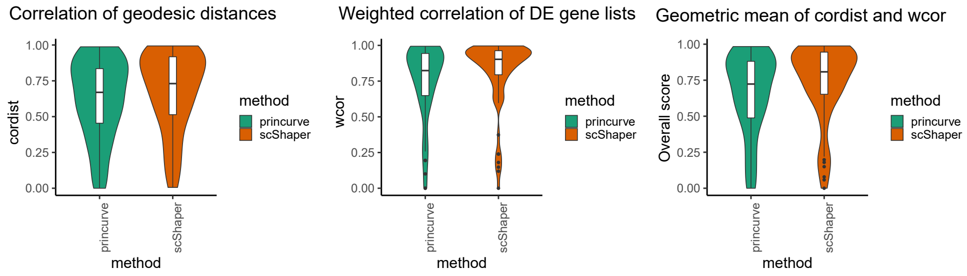

Figure 13. Benchmarking scShaper and the principal curves algorithm (princurve) for trajectory inference from scRNA-seq data in **Publication III**. DE = Differential Expression. Adapted from **Publication III**.

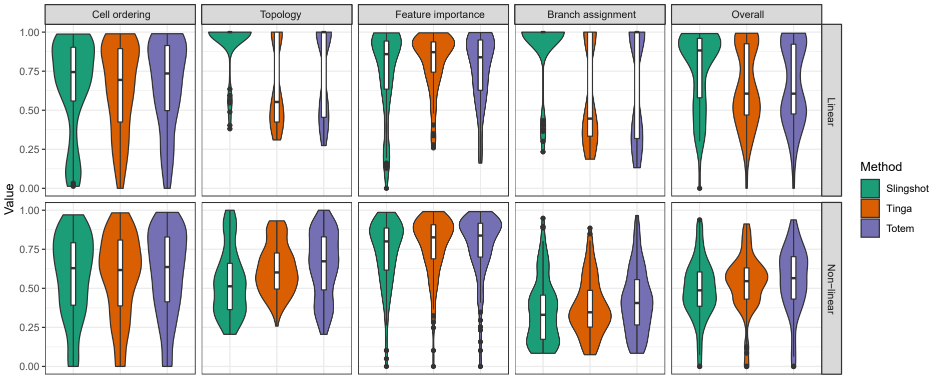

Figure 14. Results of dynverse benchmarking in **Publication IV**. The results are grouped by whether the dataset has a linear or non-linear trajectory. The benchmark data included a total of 69 linear and 147 non-linear datasets. The overall score denotes the geometric mean of the four other metrics, penalizing small values.

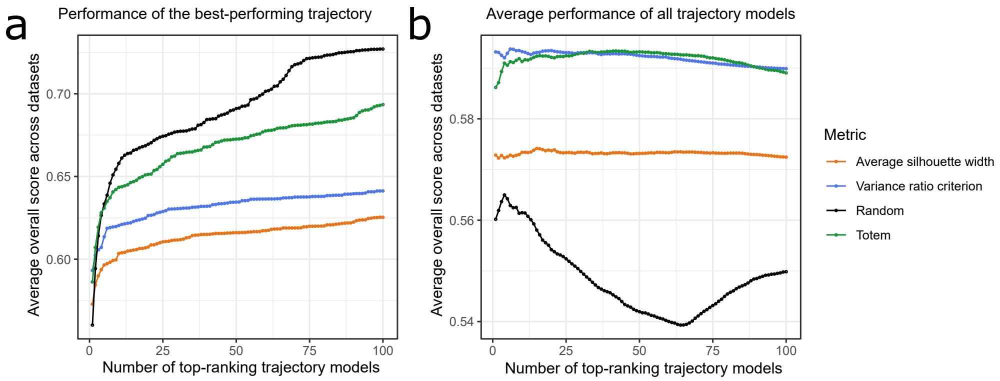

Figure 15. Comparison of clustering selection methods for trajectory inference in **Publication IV**. For each of the 216 dynverse benchmark datasets, we generated 10,000 dissimilar clustering results and ranked the clustering results using four different methods: Random, which ranks the clusterings randomly; VRC; ASW; Totem, which uses the cell connectivity and the VRC. We altered the number of selected trajectory models (x-axis) and calculated the average overall score across the datasets (y-axis) in two different ways. (a) The average overall score was calculated by considering only the best-performing trajectory in each dataset. (b) Overall scores were averaged across all selected trajectories. VRC = Variance Ratio Criterion. ASW = Average Silhouette Width.

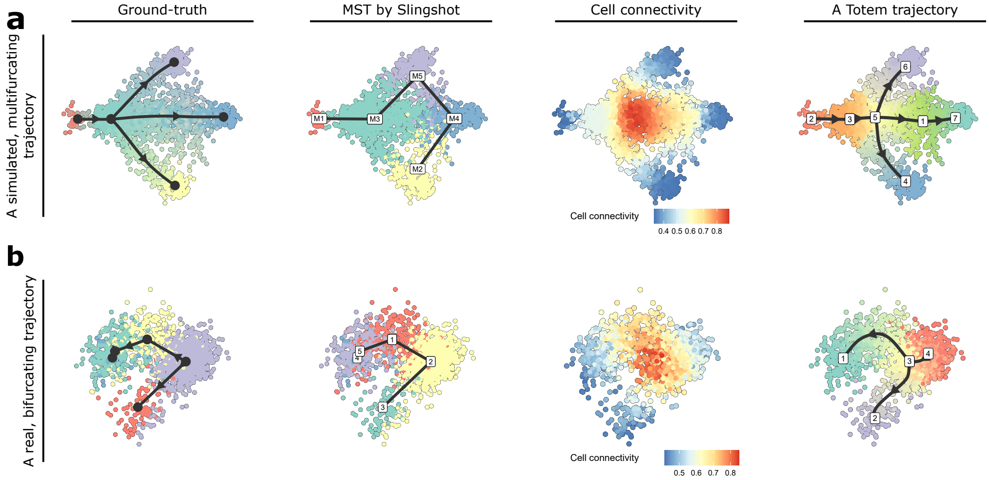

Figure 16. Examples of trajectory inference using Slingshot and Totem in **Publication IV**. (a) A simulated dataset with a multifurcating trajectory. (b) A real dataset from mouse thymus, with a bifurcating trajectory. Ground truth in the first column denotes the trajectory specified in the dynverse benchmark dataset. The second column shows the MST inferred by Slingshot when using the ground-truth clustering as input, i.e., the clustering that was inferred from the milestone percentages and milestone network.

6 Discussion

In this chapter, we discuss the findings of this thesis. We begin by discussing the scientific importance and novelty. We continue with discussions of the challenges faced during conducting the studies, the limitations in the benchmarks, and the novel computational models introduced. Furthermore, we discuss the future prospects of the works, that is, what could be potentially done to improve the novel computational methods and the related benchmarking.

6.1 Scientific importance and novelty

Our studies introduced three new computational methods for scRNA-seq data analysis. The machine learning algorithms that form the basis of the methods are, to our knowledge, novel. While there exist comprehensive reviews that aim to summarize the existing methods, authors are sometimes forced to limit the number of reviewed methods if the number of algorithms is high to those that are in active use, which is especially true for many intensively studied machine learning fields, such as clustering [161]. Although a literature review is part of every scientific work to ensure novelty and give credit to related, existing works, its success largely depends on the accuracy of the scientific literature search engines, and performing it can be very time-consuming.

The algorithms proposed in this thesis open new avenues for machine learning research. Since cross-field use of algorithms is a routine practice in computational science, data analysts outside the single-cell field can adopt the algorithms for other applications. After all, none of the machine learning algorithms that are now routinely used in scRNA-seq analysis, such as PCA, *t*-SNE, or graph-based clustering, were originally developed for this application.

It is not enough to invent new algorithms that solve computational problems. The tools that implement the algorithms need to be computationally efficient, easy to use, and clearly documented so that new users can use them with a moderate effort to extract knowledge from data. In trajectory inference, the main issue slowing down the process is the lack of one-size-fits-all methods, and the users are often forced to test different methods and adjust their parameters to obtain biologically meaningful or otherwise satisfactory trajectories [26]. Many new users are unaware of this issue because tool manuals rarely address it adequately, which results in extra work and suboptimal trajectories. This issue motivated us to develop Totem (**Publication IV**), of which central idea is to facilitate the trajectory optimization process by providing a a metric called cell connectivity as a reference, which helps locate transitions states and give an overview of the trajectory topology. Totem generates a large catalogue of trajectories from which the user can select those for further analysis that are in line with the cell connectivity, providing more flexibility compared to current state-of-the-art trajectory inference methods. The cell-connectivity-guided trajectory inference of Totem represents a novel approach to trajectory inference.

Besides method development, it is equally important to produce benchmark studies that systematically compare computational methods to provide guidelines and recommendations that method users can effortlessly follow. Without benchmark studies, an average user will have difficulties in deciding which tools to choose for certain analyses. In particular, independent studies that do not consider methods developed by the authors of the benchmark studies are valuable [158] because it is a common issue that researchers favor their own methods if they are included in their own studies. **Publication II** provided new information about DE analysis of multi-subject, multi-condition scRNA-seq data that helps users to select appropriate methods for their analyses. Prior to **Publication II**, it was unknown whether the latent variable models of the Seurat toolkit could be applied to DE analysis of multi-subject scRNA-seq data. More importantly, the recently introduced mixed models that model the subjects as a random effect (muscat_MM, MAST_RE, and NEBULA-LN) had not been benchmarked before in an non-partisan manner.

6.2 Challenges and limitations in benchmarking

Benchmarking computational methods was pivotal in all four publications of this thesis. In benchmarking, the key challenges are deciding what we want to show with the benchmarking and how to perform it in a way that accurately answers the objectives. In retrospect, it is easy to conclude that the benchmarking and especially its interpretation could have been improved in some of the works.

In **Publication I**, we compared ILoReg and four other multi-step cell type identification methods in terms of their clustering accuracy and the ability to identify rare cell types based on visualization. We measured the clustering accuracy using ARI [107], which is a widely used metric for evaluating the performance of scRNA-seq clustering algorithms [12; 14; 11]. However, while clustering is the central aim of ILoReg in order to generate clusters of cells that represent cell types, and the ICP algorithm that is at the core of ILoReg is essentially a clustering algorithm, we did not design ILoReg to compete with other clustering algorithms, such as the Louvain algorithm for graph-based clustering. Instead, it was designed to address the issues in clustering of high-dimensional data, which requires prior dimensionality reduction, and the ICP algorithm mitigates these issues with model-based feature selection. To cluster the cells into cell populations based on the PCA-aggregated ICP probabilities, ILoReg uses hierarchical clustering with Ward's agglomeration. However, the clustering algorithm used downstream of ICP and PCA could practically be any clustering algorithm. Therefore, the clustering performance evaluation in which the clustering algorithms varied between the cell type identification methods likely reflected to a significant extent the performance of the clustering algorithms and not the pre-processing steps. To demonstrate the utility of the ICP algorithm, it would have been more relevant to use a single clustering algorithm, or several algorithms, and instead change the dimensionality reduction steps prior to the PCA aggregation and compare the dimensionality reduction methods.

Another thing that could have been improved in the clustering comparison of **Publication I** would have been using the same number of clusters for each benchmarked method. Our original idea was to compare the cell type identification methods with the default settings, which include different methods for selecting the optimal number of clusters. However, the main weakness of our comparison is that we do not know which parts of the cell type identification workflow actually explain the differences in the clustering performance, and it remains unknown whether the ICP algorithm actually has a positive impact on the clustering accuracy.

Especially in **Publication I**, we can also ask whether the number of benchmark datasets was large enough to make reliable conclusions about the average performance. 11 datasets in the clustering performance evaluation are undoubtedly not persuasive enough to assess the overall performance, considering the scRNA-seq technology comprises tens of protocols that can be used to generate data and tissues with varying levels of heterogeneity. In addition, scRNA-seq datasets can include batch effects that aggravate clustering, and the data quality can vary depending on the quality of the biological samples and the lab preparation steps. Considering all these different factors in the benchmarking was not feasible back when the available benchmark data was scarce and would still require considerable effort. A major challenge would be finding benchmark data that would not bias the comparison in favor of Seurat and Scanpy, which are the most widely used scRNA-seq analysis toolkits, because a significant proportion of the publicly available datasets have been analyzed using these tools, and we would need to use the cell type annotation inferred by authors of the studies as ground truth.

In addition to evaluating the clustering performance in **Publication I**, we provided two examples of how the high-resolution visualization of ILoReg can identify cell types that are difficult to detect with other cell type identification methods. In the first example, ILoReg identified a cell population from a PBMC dataset (pbmc3k), of which gene markers indicated it to encompass naïve CD8+ T cells. The pbmc3k dataset is used as a primary example in the tutorials of Seurat and Scanpy toolkits. Around the same time as ILoReg was introduced, the developers of Seurat introduced a new normalization method, sctransform [67], and their analysis of the pbmc3k dataset showed similar results, with an improvement in the identification of the CD8+ naïve T cells, which were previously not identifiable with Seurat when the LogNormalize method was used to normalize the data. Despite this strong example showing ILoReg can identify naïve CD8+ T cells from the pbmc3k dataset, even with the older normalization method, the true cell type annotation was not available for this dataset. The CITE-seq technology [45] that measures cell surface protein (epitope) levels besides transcriptomics would provide a more reliable validation because cells are not always expressing all marker genes, but the surface proteins are generally more stable.

Overall, a significant limitation in this thesis' works was the lack of interpretation of central findings. In **Publication II**, we had the same limitation as in **Publication I** of not providing answers as to why some methods performed better than others. For example, it would have been interesting to know why the pseudo-bulk methods that use mean aggregation were generally inferior to the pseudo-bulk methods that use sum aggregation. In **Publication III**, we did not investigate or speculate why the metric that measures feature importance was higher for scShaper, even though it was the main reason why scShaper outperformed the other trajectory inference methods.

Publication II would have benefited from a better summarization of the results to reach a consensus on which DS detection methods were superior and to improve the result interpretation for readers. It would have been especially helpful in ranking the methods within each method type, of which performance differences were more subtle than the differences between the method types. The results of the two simulations, mock comparison, and reproducibility comparison could have been summarized in the same way as was done in several pioneering benchmark studies [26; 55; 146; 162]: by ranking the methods based on the average of all performance scores and visualizing the relative performance scores as a table. However, one could ask whether the metrics should be weighted equally or if, for example, precision or specificity is more important than sensitivity. Moreover, a perfect precision of one would mean that the method is too conservative because it exceeds the expected precision of 0.95 (expected FDR of 0.05). Therefore, metrics that assess the false positive control, such as the precision, would need to be adjusted to measure the difference between the expected and estimated false positive control levels.

In general, the performance evaluation metrics were used correctly in all works of this thesis, and they are the gold-standard metrics that researchers use in similar situations. However, AUROC, which was used in **Publication II**, is not optimal in situations where the distribution of positive (differentially expressed) and negative (not differentially expressed) observations is highly imbalanced [163], which is often the case in gene expression data in which only a small proportion of the genes are differentially expressed. In imbalanced data, AUROC values are more likely to be inflated and can hence misleadingly suggest that the performance is better than what it actually is. It is widely recommended to use the precision-recall curve instead of the receiver operating characteristic (ROC) curve when the distribution is imbalanced [142; 143; 144], which would have hence made it a worthwhile addition to the comparison.

Moreover, as was noted at the beginning of this section when discussing **Publication I**, it would have been better if the number of clusters was equal for the two clustersets that were compared using ARI. We observed during benchmarking that it was possible to achieve ARI values as high as 0.80 for the Baron pancreatic datasets [15] when the predicted number of clusters was only 6 and the true number of cell types in the annotation was 13. The contradiction occurred because the distribution of the cell types was imbalanced, and ARI weights the impact of clusters based on their size; and while the large cell types were clustered accurately, the small, rare cell types were not. However, other publications have conducted the clustering comparison similarly, using deviant numbers of clusters [12; 164]. Performing the comparison in both ways would have given a more comprehensive picture of the clustering performance.

In all works except **Publication I**, simulation provided the means to acquire benchmark data that included accurate ground truth. The general issue of simulation, however, is that it can never truly model the complex processes that constitute the transcription of mRNA in cells [165] because the gene expression signal that is measured by sequencing is biased by a large number of technical and biological factors [166; 167], and it is questionable whether count data without biasing factors would still follow any mathematical models, such as the negative binomial distribution. The simulation of scRNA-seq count data is mostly based on negative binomial generative models [19; 22], which can lead to poor-fitting models [168].

6.3 Limitations in novel computational methods

In **Publication IV**, we introduced Totem, a tool that facilitates the search for a clustering that generates an MST that accurately models the true cell development network by utilizing the cell connectivity metric. The main drawback of this method is its inability to handle more complex trajectories with cycles or disconnected parts. In addition, Totem does not support converging trajectories, i.e., trajectories that converge to a single cell from multiple lineages and trajectories that have both diverging and converging parts, diverging meaning trajectories in which the cells diverge from a single cell to multiple lineages. While the directions in the milestone network are easy to adjust by flipping, creating the correct pseudotime for the more complex trajectories is not as straightforward when the pseudotime estimation is performed using the principal curves algorithm.

scShaper and Totem (**Publications III** and **IV**) have the same limitation of being unable to determine the trajectory direction automatically. In this aspect, they are inferior to RNA velocity methods [97; 98], which can estimate the direction based on

RNA splicing information. However, scShaper and Totem allow more freedom in the dimensionality reduction and pre-processing steps. They can be easily used with any dimensionality reduction method, which can be useful in joint analysis of multiple datasets that requires data integration to remove batch effects. RNA velocity methods are currently not optimal for datasets that have batch effects because the samples need to be modeled independently [99], which raises further issues regarding how to accurately and analytically compare multiple trajectories. Suppose the experimental design involves multi-subject, multi-condition data as in **Publication II**. In that case, a sensible approach is to perform separate data integration for each condition, infer a trajectory for each condition, and investigate the differences between the trajectories, which are significantly smaller in number than the number of subjects. Alternatively, a single trajectory can be generated for the whole dataset, and the differential expression analysis can be effortlessly performed between the conditions along each lineage.

6.4 Future

The works of this thesis open intriguing avenues for future method development and studies. In this section, we discuss some ideas for the future development of the works presented in this thesis.

In **Publication I**, we showed that ILoReg was a promising tool for identifying cell populations with subtle transcriptomic differences from scRNA-seq data. However, we have only introduced the first version of the algorithm, which can certainly be improved to enable even more accurate cell type identification and better computational efficiency. After all, the first algorithm is often flawed, especially when the underlying principle is unique, and developers continue to improve their algorithms over time.

A modification that would likely improve the cell type identification accuracy would be implementing a mechanism that controls the learning rate in the ICP algorithm, i.e., how fast the clustering similarity (ARI) is allowed to change during the iteration process. In machine learning, we know from objective function optimization, such as stochastic gradient descent (SGD), that a lower learning rate generally yields better optimization results [169]. In SGD optimization, it is common to use only a subset of the training data, referred to as the batch. Implementing a mechanism that would allow using a specific batch size during the ICP learning would probably not significantly decrease the performance but would decrease the run time. Moreover, we have only tested one supervised classification algorithm, logistic regression, but the range of available classifier algorithms is broad. Methods such as random forest and support vector machines, which work well with high-dimensional data, are worth considering. It would also be interesting to investigate how the ICP algorithm would work with dimensionally reduced data, which would allow testing classification algorithms such as *k*-nearest neighbor algorithms that perform better when the number of features is small. It may also be possible to improve both the run time and accuracy by a better initialization of the ICP clusters. For example, instead of initializing the ICP clusters randomly, the clusters could be initially defined as clusters estimated using a different clustering algorithm, such as *k*-means or *k*-medoids.

The ICP algorithm of ILoReg holds potential as a beneficial pre-processing method in trajectory inference. Methods such as Palantir [170] use diffusion maps [171] to improve the capture of differentiation trajectories in scRNA-seq data because the cell types are often clustered too tightly to infer the correct trajectory topology. MARGARET [172] uses refined embeddings to adjust the distances between neighboring cell types to provide a more accurate topological representation. Since we observed from the *t*-SNE visualizations that ILoReg could segregate cell types into more distinct subsets, this could also facilitate trajectory inference when the cell types are otherwise too tightly clustered.

We hinted in **Publication III** that scShaper could potentially be used for generating smooth, linear paths through the lineages of tree-shaped trajectories, which is how Slingshot [28] operates. However, a few issues need to be solved before scShaper can be applied to this purpose. Pseudotime measures the progress of cell differentiation at the single-cell level, and it would need to correlate with the cell distances in the input data. However, pseudotime in scShaper is currently calculated by averaging the discrete pseudotimes derived from the numeric cluster labels that increase in even intervals from 1 to *k*, from 0 to 1 when min-max-scaled, without correlating with the cell distances in the input embedding. Discrete pseudotime that would correlate better with the cell distances in the input data could be obtained using cluster centroid distances, which would be used to adjust the numeric cluster labels. The discrepancy in the cluster distances forms an issue in lineage smoothing because the length of the lineages can vary considerably, and the pseudotimes of the lineages need to be synchronous when they are averaged in trajectory parts that have several smoothed curves passing through simultaneously.

Furthermore, while we were developing Totem (**Publication IV**), we observed that the *k*-means algorithm used in scShaper produced more similar clustering results when the number of clusters was small, and *k*-medoids (CLARA) generated more dissimilar clustering results than *k*-means. Using *k*-medoids to generate more dissimilar clustering would likely benefit scShaper by creating more varying discrete pseudotime results, which would, in turn, generate more precise pseudotime in the aggregation step because it would provide more information about the relative cell positions in the trajectory.

Although Kruskal's algorithm is efficient for graphs with many vertices (clusters), it becomes slower when the graph is dense and includes many edges. Therefore, it would be interesting to investigate whether other MST algorithms, such as Prim's algorithm, would be more efficient for the purpose of path optimization in scShaper. However, like Kruskal's algorithm, these algorithms are not directly applicable to pathfinding and would need to be modified. The run time of scShaper can also be decreased by changing the clustering algorithm. For example, the mini-batch *k*-means [173] algorithm is a faster implementation of the basic *k*-means algorithm.

Each computational method has a multi-step workflow, and each step should be provided with a reasonable argument and actionable proof of its utility. It is possible that some of the steps are not beneficial. For example, in scShaper the average discrete pseudotime is transformed into rank values (integers) before LOESS smoothing. The original idea was to make small pseudotime changes between cells appear more pronounced, which would help capture small transition states in trajectories. While LOESS can be justified as a step to correct local unevenness, it is not entirely clear how suitable it is for rank-transformed data for which monotonic smoothing has already been performed. Totem includes a step that scales the cluster-level connectivity values so that the maximum connectivity of each clustering is one. The scaling was intended as a step to bring the different cluster-level connectivity vectors to a uniform scale before averaging. However, no proof was provided for that this step actually is beneficial.

Publication II and other similar studies [18; 19] have reached the same conclusion that the pseudo-bulk methods can sometimes be underpowered, that is, their ability to identify true positive findings is weak due to a small sample size. In addition, they can only compare the mean shift between subject groups and not the variance or other attributes in the expression distributions of the subjects. Novel statistical methods that can compare the distributions of subjects or other biological replicates between conditions have been developed [174; 175], and they reportedly provide better sensitivity than the pseudo-bulk methods. However, the performance of these methods need to be still validated in an independent manner.

7 Summary of publications

I: ILoReg: a tool for high-resolution cell population identification from single-cell RNA-seq data

In this paper, we introduced a novel machine learning algorithm, ICP, as a solution to address the ”curse of dimensionality” phenomenon in unsupervised cell type identification from scRNA-seq data. In ICP, the high-dimensional scRNA-seq input data are clustered iteratively by training a logistic regression model with a subset of the data, projecting the whole dataset with the trained model, and optimizing the clustering similarity between the projected and training cluster labels. The logistic regression model learns to select the most important genes using the L1-regularization, making the clustering outcome depend on a subset of informative genes that segregate cell types. In the next phase, the cluster probabilities from an ensemble of ICP models are used as features, which are processed further to identify the cell types by visualization, clustering, and gene marker discovery. Our examples showed how ILoReg, the R package that implements the pipeline, could identify biologically relevant cell populations with subtle transcriptomic differences that the other methods could not find.

II: Benchmarking methods for detecting differential states between conditions from multi-subject single-cell RNA-seq data

DS analysis of scRNA-seq data involves comparing expression levels between cell populations. scRNA-seq experiments are increasingly designed to study gene expression changes between two or multiple conditions, such as healthy and sick subjects. Suppose the cells within the conditions are from different subjects. In that case, this can result in a hierarchical data structure in which cells within subjects are at the transcriptomic level more similar than cells between the subjects. Therefore, the expression levels of cells within a condition are not statistically independent, resulting in a pseudoreplication bias that induces false positive findings. Two approaches have been developed to mitigate the pseudo-replicate bias in multi-subject DS analysis: pseudo-bulk methods that aggregate counts at the subject level and use bulk RNA-seq tools for DS analysis and mixed models that model the subjects as a random effect. Our comprehensive comparison showed that the pseudo-bulk methods and mixed models were superior to the methods that do not model subjects in any way. Generally, our findings indicated that pseudo-bulk methods outperformed mixed models.

III: scShaper: an ensemble method for fast and accurate linear trajectory inference from single-cell RNA-seq data

In this work, we developed a new general-purpose algorithm, scShaper, for inferring linear paths through high-dimensional data. scShaper is based on graph theory and uses a modified Kruskal's algorithm to optimize linear paths for graphs. scShaper runs the *k*-means clustering algorithm a large number of times with different *k* values to generate dissimilar clustering results, calculates the centroids of the clusters in each clustering, estimates the optimal path through the centroids of each clustering using a modified Kruskal's algorithm, and uses the paths from all the clustersets to obtain discrete pseudotimes that measure the cell differentiation at the cluster level. The discrete pseudotimes are aggregated with PCA and smoothed with LOESS, generating continuous pseudotime that measures cell differentiation at the cell level. Comprehensive benchmarking using scRNA-seq datasets suggested that scShaper was superior to state-of-the-art trajectory inference methods. Moreover, the results indicated that scShaper outperformed the principal curves algorithm, a popular method for inferring linear paths through single-cell data.

IV: Totem: a user-friendly tool for clustering-based inference of tree-shaped trajectories from single-cell data

In clustering-based inference of tree-shaped trajectories, a clusterset is used as the basis to infer the milestone network as an MST, which models how the cell types (milestones) are connected as a network. The MST is then smoothed to generate a directed trajectory, along with pseudotime that measures the cell differentiation at the single-cell level. A key challenge in this process is finding an optimal clustering that will generate the correct milestone network. Even if accurate cell type labels are available, the resulting MST will not necessarily correlate accurately with the true milestone network. To address this challenge, we developed a user-friendly tool, Totem, that enables fast and effortless search for an optimal clustering used to generate the MST. To facilitate the clustering selection, we introduced a new metric called cell connectivity, which enables to locate milestones relevant to the trajectory and give a general overview of the trajectory. With the cell connectivity as a reference, the user can compare different MSTs generated from different clustersets and select the ones for downstream analysis that are in line with the cell connectivity profile.

List of References

[1] Zhong Wang, Mark Gerstein, and Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63, 2009.

[2] Shanrong Zhao, Wai-Ping Fung-Leung, Anton Bittner, Karen Ngo, and Xuejun Liu. Comparison of rna-seq and microarray in transcriptome profiling of activated t cells. *PloS one*, 9(1):e78644, 2014.

[3] Ashraful Haque, Jessica Engel, Sarah A Teichmann, and Tapio Lönnberg. A practical guide to single-cell rna-sequencing for biomedical research and clinical applications. *Genome medicine*, 9(1):1–12, 2017.

[4] Simone Picelli, Åsa K Björklund, Omid R Faridani, Sven Sagasser, Gösta Winberg, and Rickard Sandberg. Smart-seq2 for sensitive full-length transcriptome profiling in single cells. *Nature methods*, 10(11):1096–1098, 2013.

[5] Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):1–12, 2017.

[6] Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5): 1202–1214, 2015.

[7] Vasilis Ntranos, Lynn Yi, Páll Melsted, and Lior Pachter. A discriminative learning approach to differential expression analysis for single-cell rna-seq. *Nature methods*, 16(2):163–166, 2019.

[8] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):1–5, 2018.

[9] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411–420, 2018.

[10] Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

[11] Vladimir Yu Kiselev, Kristina Kirschner, Michael T Schaub, Tallulah Andrews, Andrew Yiu, Tamir Chandra, Kedar N Natarajan, Wolf Reik, Mauricio Barahona, Anthony R Green, et al. Sc3: consensus clustering of single-cell rna-seq data. *Nature methods*, 14(5):483–486, 2017.

[12] Peijie Lin, Michael Troup, and Joshua WK Ho. Cidr: Ultrafast and accurate clustering through imputation for single-cell rna-seq data. *Genome biology*, 18(1):1–11, 2017.

[13] Christopher Yau et al. pcareduce: hierarchical clustering of single cell transcriptional profiles. *BMC bioinformatics*, 17(1):1–11, 2016.

[14] Josip S Herman, Dominic Grün, et al. Fateid infers cell fate bias in multipotent progenitors from single-cell rna-seq data. *Nature methods*, 15(5):379–386, 2018.

[15] 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, et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell systems*, 3(4):346–360, 2016.

[16] Jordan W Squair, Matthieu Gautier, Claudia Kathe, Mark A Anderson, Nicholas D James, Thomas H Hutson, Rémi Hudelle, Taha Qaiser, Kaya JE Matson, Quentin Barraud, et al. Con-

fronting false discoveries in single-cell differential expression. *Nature communications*, 12(1): 1–15, 2021.

[17] Andrew L Thurman, Jason A Ratcliff, Michael S Chimenti, and Alejandro A Pezzulo. Differential gene expression analysis for multi-subject single-cell rna-sequencing studies with aggregatebiovar. *Bioinformatics*, 37(19):3243–3251, 2021.

[18] Kip D Zimmerman, Mark A Espeland, and Carl D Langefeld. A practical solution to pseudoreplication bias in single-cell studies. *Nature communications*, 12(1):1–9, 2021.

[19] Helena L Crowell, Charlotte Soneson, Pierre-Luc Germain, Daniela Calini, Ludovic Collin, Catarina Raposo, Dheeraj Malhotra, and Mark D Robinson. Muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nature communications*, 11(1):1–12, 2020.

[20] Aaron T L Lun, Karsten Bach, and John C Marioni. Pooling across cells to normalize single-cell rna sequencing data with many zero counts. *Genome biology*, 17(1):1–14, 2016.

[21] Greg Finak, Andrew McDavid, Masanao Yajima, Jingyuan Deng, Vivian Gersuk, Alex K Shalek, Chloe K Slichter, Hannah W Miller, M Juliana McElrath, Martin Prlic, et al. Mast: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell rna sequencing data. *Genome biology*, 16(1):1–13, 2015.

[22] Liang He, Jose Davila-Velderrain, Tomokazu S Sumida, David A Hafler, Manolis Kellis, and Alexander M Kulminski. Nebula is a fast negative binomial mixed model for differential or co-expression analysis of large-scale multi-subject single-cell data. *Communications biology*, 4(1): 1–17, 2021.

[23] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck III, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

[24] Yanglan Gan, Cheng Guo, Wenjing Guo, Guangwei Xu, and Guobing Zou. Entropy-based inference of transition states and cellular trajectory for single-cell transcriptomics. *Briefings in Bioinformatics*, 23(4):bbac225, 2022.

[25] David Schafflick, Chenling A Xu, Maike Hartlehnert, Michael Cole, Andreas Schulte-Mecklenbeck, Tobias Lautwein, Jolien Wolbert, Michael Heming, Sven G Meuth, Tanja Kuhlmann, et al. Integrated single cell analysis of blood and cerebrospinal fluid leukocytes in multiple sclerosis. *Nature communications*, 11(1):1–14, 2020.

[26] Wouter Saelens, Robrecht Cannoodt, Helena Todorov, and Yvan Saeys. A comparison of single-cell trajectory inference methods. *Nature biotechnology*, 37(5):547–554, 2019.

[27] Helena Todorov, Robrecht Cannoodt, Wouter Saelens, and Yvan Saeys. Tinga: fast and flexible trajectory inference with growing neural gas. *Bioinformatics*, 36(Supplement_1):i66–i74, 2020.

[28] 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):1–16, 2018.

[29] Luca Albergante, Evgeny Mirkes, Jonathan Bac, Huidong Chen, Alexis Martin, Louis Faure, Emmanuel Barillot, Luca Pinello, Alexander Gorban, and Andrei Zinovyev. Robust and scalable learning of complex intrinsic dataset geometry via elpigraph. *Entropy*, 22(3):296, 2020.

[30] Robrecht Cannoodt, Wouter Saelens, Dorine Sichien, Simon Tavernier, Sophie Janssens, Martin Guilliams, Bart Lambrecht, Katleen De Preter, and Yvan Saeys. Scorpius improves trajectory inference and identifies novel modules in dendritic cell development. *Biorxiv*, page 079509, 2016.

[31] Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

[32] David Bick and David Dimmock. Whole exome and whole genome sequencing. *Current opinion in pediatrics*, 23(6):594–600, 2011.

[33] Jamie K Teer and James C Mullikin. Exome sequencing: the sweet spot before whole genomes. *Human molecular genetics*, 19(R2):R145–R151, 2010.

[34] Peter J Park. Chip–seq: advantages and challenges of a maturing technology. *Nature reviews genetics*, 10(10):669–680, 2009.

[35] Arfa Mehmood, Asta Laiho, Mikko S Venäläinen, Aidan J McGlinchey, Ning Wang, and Laura L Elo. Systematic evaluation of differential splicing tools for rna-seq studies. *Briefings in bioinformatics*, 21(6):2052–2065, 2020.

[36] Shihao Shen, Juw Won Park, Zhi-xiang Lu, Lan Lin, Michael D Henry, Ying Nian Wu, Qing Zhou, and Yi Xing. rmats: robust and flexible detection of differential alternative splicing from replicate rna-seq data. *Proceedings of the National Academy of Sciences*, 111(51):E5593–E5601, 2014.

[37] Robert Piskol, Gokul Ramaswami, and Jin Billy Li. Reliable identification of genomic variants from rna-seq data. *The American Journal of Human Genetics*, 93(4):641–651, 2013.

[38] Jean-Simon Brouard, Flavio Schenkel, Andrew Marete, and Nathalie Bissonnette. The gatk joint genotyping workflow is appropriate for calling variants in rna-seq experiments. *Journal of animal science and biotechnology*, 10(1):1–6, 2019.

[39] AT Vivek and Shailesh Kumar. Computational methods for annotation of plant regulatory non-coding rnas using rna-seq. *Briefings in Bioinformatics*, 22(4):bbaa322, 2021.

[40] Nicholas E Ilott and Chris P Ponting. Predicting long non-coding rnas using rna sequencing. *Methods*, 63(1):50–59, 2013.

[41] Rory Stark, Marta Grzelak, and James Hadfield. Rna sequencing: the teenage years. *Nature Reviews Genetics*, 20(11):631–656, 2019.

[42] Samuel Marguerat and Jürg Bähler. Rna-seq: from technology to biology. *Cellular and molecular life sciences*, 67(4):569–579, 2010.

[43] Xiliang Wang, Yao He, Qiming Zhang, Xianwen Ren, and Zemin Zhang. Direct comparative analyses of 10x genomics chromium and smart-seq2. *Genomics, proteomics & bioinformatics*, 19(2):253–266, 2021.

[44] Geng Chen, Baitang Ning, and Tieliu Shi. Single-cell rna-seq technologies and related computational data analysis. *Frontiers in genetics*, page 317, 2019.

[45] 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.

[46] Zhi-Jie Cao and Ge Gao. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. *Nature Biotechnology*, pages 1–9, 2022.

[47] Yingxin Lin, Tung-Yu Wu, Sheng Wan, Jean YH Yang, Wing H Wong, and YX Wang. scjoint integrates atlas-scale single-cell rna-seq and atac-seq data with transfer learning. *Nature Biotechnology*, 40(5):703–710, 2022.

[48] Mika Sarkin Jain, Krzysztof Polanski, Cecilia Dominguez Conde, Xi Chen, Jongeun Park, Lira Mamanova, Andrew Knights, Rachel A Botting, Emily Stephenson, Muzlifah Haniffa, et al. Multimap: dimensionality reduction and integration of multimodal data. *Genome biology*, 22 (1):1–26, 2021.

[49] Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcześniak, Daniel J Gaffney, Laura L Elo, Xuegong Zhang, et al. A survey of best practices for rna-seq data analysis. *Genome biology*, 17(1):1–19, 2016.

[50] Ying Ma, Shiquan Sun, Xuequn Shang, Evan T Keller, Mengjie Chen, and Xiang Zhou. Integrative differential expression and gene set enrichment analysis using summary statistics for scrna-seq studies. *Nature communications*, 11(1):1–13, 2020.

[51] 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, et al. Scenic: single-cell regulatory network inference and clustering. *Nature methods*, 14(11): 1083–1086, 2017.

[52] Emma Dann, Neil C Henderson, Sarah A Teichmann, Michael D Morgan, and John C Marioni. Differential abundance testing on single-cell data using k-nearest neighbor graphs. *Nature Biotechnology*, 40(2):245–253, 2022.

[53] Jun Zhao, Ariel Jaffe, Henry Li, Ofir Lindenbaum, Esen Sefik, Ruaidhrí Jackson, Xiuyuan Cheng, Richard A Flavell, and Yuval Kluger. Detection of differentially abundant cell subpopulations in scrna-seq data. *Proceedings of the National Academy of Sciences*, 118(22):e2100293118, 2021.

[54] Axel A Almet, Zixuan Cang, Suoqin Jin, and Qing Nie. The landscape of cell–cell communication through single-cell transcriptomics. *Current opinion in systems biology*, 26:12–23, 2021.

[55] 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.

[56] Manuel Holtgrewe, Clemens Messerschmidt, Mikko Nieminen, and Dieter Beule. Digestiflow: from bcl to fastq with ease, 2020.

[57] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997*, 2013.

[58] Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. Star: ultrafast universal rna-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.

[59] Cole Trapnell, Lior Pachter, and Steven L Salzberg. Tophat: discovering splice junctions with rna-seq. *Bioinformatics*, 25(9):1105–1111, 2009.

[60] Teemu Kivioja, Anna Vähärautio, Kasper Karlsson, Martin Bonke, Martin Enge, Sten Linnarsson, and Jussi Taipale. Counting absolute numbers of molecules using unique molecular identifiers. *Nature methods*, 9(1):72–74, 2012.

[61] Yingying Cao, Simo Kitanovski, Ralf Küppers, and Daniel Hoffmann. Umi or not umi, that is the question for scrna-seq zero-inflation. *Nature Biotechnology*, 39(2):158–159, 2021.

[62] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[63] Tom Smith, Andreas Heger, and Ian Sudbery. Umi-tools: modeling sequencing errors in unique molecular identifiers to improve quantification accuracy. *Genome research*, 27(3):491–499, 2017.

[64] Tomislav Ilicic, Jong Kyoung Kim, Aleksandra A Kolodziejczyk, Frederik Otzen Bagger, Davis James McCarthy, John C Marioni, and Sarah A Teichmann. Classification of low quality cells from single-cell rna-seq data. *Genome biology*, 17(1):1–15, 2016.

[65] Pierre-Luc Germain, Aaron Lun, Carlos Garcia Meixide, Will Macnair, and Mark D Robinson. Doublet identification in single-cell sequencing data using scdblfinder. *F1000Research*, 10, 2021.

[66] Jani Huuhtanen, Dipabarna Bhattacharya, Tapio Lönnberg, Matti Kankainen, Cassandra Kerr, Jason Theodoropoulos, Hanna Rajala, Carmelo Gurnari, Tiina Kasanen, Till Braun, et al. Single-cell characterization of leukemic and non-leukemic immune repertoires in cd8+ t-cell large granular lymphocytic leukemia. *Nature Communications*, 13(1):1981, 2022.

[67] Christoph Hafemeister and Rahul Satija. Normalization and variance stabilization of single-cell rna-seq data using regularized negative binomial regression. *Genome biology*, 20(1):1–15, 2019.

[68] Mark D Robinson and Alicia Oshlack. A scaling normalization method for differential expression analysis of rna-seq data. *Genome biology*, 11(3):1–9, 2010.

[69] Nicholas Lytal, Di Ran, and Lingling An. Normalization methods on single-cell rna-seq data: an empirical survey. *Frontiers in genetics*, 11:41, 2020.

[70] Rhonda Bacher, Li-Fang Chu, Ning Leng, Audrey P Gasch, James A Thomson, Ron M Stewart, Michael Newton, and Christina Kendziorski. Scnorm: robust normalization of single-cell rna-seq data. *Nature methods*, 14(6):584–586, 2017.

[71] Beate Vieth, Swati Parekh, Christoph Ziegenhain, Wolfgang Enard, and Ines Hellmann. A systematic evaluation of single cell rna-seq analysis pipelines. *Nature communications*, 10(1):1–11, 2019.

[72] Alex A Pollen, Tomasz J Nowakowski, Joe Shuga, Xiaohui Wang, Anne A Leyrat, Jan H Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, et al. Low-coverage single-cell

mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature biotechnology*, 32(10):1053–1058, 2014.

[73] Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. scmap: projection of single-cell rna-seq data across data sets. *Nature methods*, 15(5):359–362, 2018.

[74] Dvir Aran, Agnieszka P Looney, Leqian Liu, Esther Wu, Valerie Fong, Austin Hsu, Suzanna Chak, Ram P Naikawadi, Paul J Wolters, Adam R Abate, et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nature immunology*, 20(2): 163–172, 2019.

[75] Aleksandr Ianevski, Anil K Giri, and Tero Aittokallio. Fully-automated and ultra-fast cell-type identification using specific marker combinations from single-cell transcriptomic data. *Nature communications*, 13(1):1–10, 2022.

[76] Jose Alquicira-Hernandez, Anuja Sathe, Hanlee P Ji, Quan Nguyen, and Joseph E Powell. scpred: accurate supervised method for cell-type classification from single-cell rna-seq data. *Genome biology*, 20(1):1–17, 2019.

[77] 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.

[78] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019.

[79] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[80] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[81] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[82] Dongfang Wang and Jin Gu. Vasc: dimension reduction and visualization of single-cell rna-seq data by deep variational autoencoder. *Genomics, proteomics & bioinformatics*, 16(5):320–331, 2018.

[83] Trung Ngo Trong, Juha Mehtonen, Gerardo González, Roger Kramer, Ville Hautamäki, and Merja Heinäniemi. Semisupervised generative autoencoder for single-cell data. *Journal of Computational Biology*, 27(8):1190–1203, 2020.

[84] Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Žiga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, et al. Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology*, 40(1):121–130, 2022.

[85] Carlo Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62, 1936.

[86] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

[87] Laleh Haghverdi, Aaron TL Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421–427, 2018.

[88] Malte D Luecken, Maren Büttner, Kridsadakorn Chaichoompu, Anna Danese, Marta Interlandi, Michaela F Müller, Daniel C Strobl, Luke Zappia, Martin Dugas, Maria Colomé-Tatché, et al. Benchmarking atlas-level data integration in single-cell genomics. *Nature methods*, 19(1):41–50, 2022.

[89] Kip D Zimmerman and Carl D Langefeld. Hierarchicell: an r-package for estimating power for tests of differential expression with single-cell data. *BMC genomics*, 22(1):1–8, 2021.

[90] 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 dynam-

ics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology*, 32(4):381–386, 2014.

[91] 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.

[92] Qi Mao, Li Wang, Ivor W Tsang, and Yijun Sun. Principal graph and structure learning based on reversed graph embedding. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2227–2241, 2016.

[93] Junyue Cao, Malte Spielmann, Xiaojie Qiu, Xingfan Huang, Daniel M Ibrahim, Andrew J Hill, Fan Zhang, Stefan Mundlos, Lena Christiansen, Frank J Steemers, et al. The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566(7745):496–502, 2019.

[94] F Alexander Wolf, Fiona K Hamey, Mireya Plass, Jordi Solana, Joakim S Dahlin, Berthold Göttgens, Nikolaus Rajewsky, Lukas Simon, and Fabian J Theis. Paga: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome biology*, 20(1):1–9, 2019.

[95] 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.

[96] Philipp Weiler, Koen Van den Berge, Kelly Street, and Simone Tiberi. A guide to trajectory inference and rna velocity. In *Single Cell Transcriptomics: Methods and Protocols*, pages 269–292. Springer, 2022.

[97] Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E Kastriti, Peter Lönnerberg, Alessandro Furlan, et al. Rna velocity of single cells. *Nature*, 560(7719):494–498, 2018.

[98] 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.

[99] Volker Bergen, Ruslan A Soldatov, Peter V Kharchenko, and Fabian J Theis. Rna velocity—current challenges and future perspectives. *Molecular systems biology*, 17(8):e10282, 2021.

[100] Dongyuan Song and Jingyi Jessica Li. Pseudotimede: inference of differential gene expression along cell pseudotime with well-calibrated p-values from single-cell rna sequencing data. *Genome biology*, 22(1):1–25, 2021.

[101] Kieran R Campbell and Christopher Yau. switchde: inference of switch-like differential expression along single-cell trajectories. *Bioinformatics*, 33(8):1241–1242, 2017.

[102] David S Fischer, Fabian J Theis, and Nir Yosef. Impulse model-based differential expression analysis of time course sequencing data. *Nucleic acids research*, 46(20):e119–e119, 2018.

[103] Elvis Han Cui, Dongyuan Song, Weng Kee Wong, and Jingyi Jessica Li. Single-cell generalized trend model (scgtm): a flexible and interpretable model of gene expression trend along cell pseudotime. *Bioinformatics*, 38(16):3927–3934, 2022.

[104] Yuval Lieberman, Lior Rokach, and Tal Shay. Castle–classification of single cells by transfer learning: harnessing the power of publicly available single cell rna sequencing experiments to annotate new experiments. *PloS one*, 13(10):e0205499, 2018.

[105] Tallulah S Andrews and Martin Hemberg. M3drop: dropout-based feature selection for scrnaseq. *Bioinformatics*, 35(16):2865–2867, 2019.

[106] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *the Journal of machine Learning research*, 9:1871–1874, 2008.

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

[108] Gökcen Eraslan, Lukas M Simon, Maria Mircea, Nikola S Mueller, and Fabian J Theis. Single-cell rna-seq denoising using a deep count autoencoder. *Nature communications*, 10(1):1–14, 2019.

[109] Thomas A Geddes, Taiyun Kim, Lihao Nan, James G Burchfield, Jean YH Yang, Dacheng Tao, and Pengyi Yang. Autoencoder-based cluster ensembles for single-cell rna-seq data analysis. *BMC bioinformatics*, 20(19):1–11, 2019.

[110] Yulun Wu, Yanming Guo, Yandong Xiao, and Songyang Lao. Aae-sc: A scrna-seq clustering framework based on adversarial autoencoder. *IEEE Access*, 8:178962–178975, 2020.

[111] Dai-Jun Zhang, Ying-Lian Gao, Jing-Xiu Zhao, Chun-Hou Zheng, and Jin-Xing Liu. A new graph autoencoder-based consensus-guided model for scrna-seq cell type detection. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[112] Bin Yu, Chen Chen, Ren Qi, Ruiqing Zheng, Patrick J Skillman-Lawrence, Xiaolin Wang, Anjun Ma, and Haiming Gu. scgmai: a gaussian mixture model for clustering single-cell rna-seq data based on deep autoencoder. *Briefings in Bioinformatics*, 22(4):bbaa316, 2021.

[113] Eugene Lin, Sudipto Mukherjee, and Sreeram Kannan. A deep adversarial variational autoencoder model for dimensionality reduction in single-cell rna sequencing analysis. *BMC bioinformatics*, 21(1):1–11, 2020.

[114] Leonard Kaufman and Peter J Rousseeuw. Clustering large applications (program clara). *Finding groups in data: an introduction to cluster analysis*, pages 126–146, 2008.

[115] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936.

[116] Raphael Petegrosso, Zhuliu Li, and Rui Kuang. Machine learning and statistical methods for clustering single-cell rna-sequencing data. *Briefings in bioinformatics*, 21(4):1209–1223, 2020.

[117] Ren Qi, Anjun Ma, Qin Ma, and Quan Zou. Clustering and classification methods for single-cell rna-sequencing data. *Briefings in bioinformatics*, 21(4):1196–1208, 2020.

[118] Wenpin Hou, Zhicheng Ji, Hongkai Ji, and Stephanie C Hicks. A systematic evaluation of single-cell rna-sequencing imputation methods. *Genome biology*, 21(1):1–30, 2020.

[119] Davis J McCarthy, Kieran R Campbell, Aaron TL Lun, and Quin F Wills. Scater: pre-processing, quality control, normalization and visualization of single-cell rna-seq data in r. *Bioinformatics*, 33(8):1179–1186, 2017.

[120] Johannes Smolander, Sini Junttila, Mikko S Venäläinen, and Laura L Elo. Iloreg: a tool for high-resolution cell population identification from single-cell rna-seq data. *Bioinformatics*, 37 (8):1107–1114, 2021.

[121] Aaron TL Lun and John C Marioni. Overcoming confounding plate effects in differential expression analyses of single-cell rna-seq data. *Biostatistics*, 18(3):451–464, 2017.

[122] Matthew E Ritchie, Belinda Phipson, DI Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic acids research*, 43(7):e47–e47, 2015.

[123] Tomi Suomi, Fatemeh Seyednasrollah, Maria K Jaakkola, Thomas Faux, and Laura L Elo. Rots: An r package for reproducibility-optimized statistical testing. *PLoS computational biology*, 13 (5):e1005562, 2017.

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

[125] Sini Junttila, Johannes Smolander, and Laura L Elo. Benchmarking methods for detecting differential states between conditions from multi-subject single-cell rna-seq data. *bioRxiv*, 2022.

[126] Kieran Campbell, Chris P Ponting, and Caleb Webber. Laplacian eigenmaps and principal curves for high resolution pseudotemporal ordering of single-cell rna-seq profiles. *bioRxiv*, page 027219, 2015.

[127] Alexander Gorban and Andrey Zinovyev. Elastic principal graphs and manifolds and their practical applications. *Computing*, 75(4):359–379, 2005.

[128] Bernd Fritzke. A growing neural gas network learns topologies. *Advances in neural information processing systems*, 7, 1994.

[129] Johannes Smolander, Sini Junttila, Mikko S Venäläinen, and Laura L Elo. scshaper: an ensemble method for fast and accurate linear trajectory inference from single-cell rna-seq data. *Bioinformatics*, 38(5):1328–1335, 2022.

[130] Johannes Smolander, Sini Junttila, and Laura L Elo. Totem: a user-friendly tool for clustering-based inference of tree-shaped trajectories from single-cell data. *bioRxiv*, 2022.

[131] Peter van Galen, Volker Hovestadt, Marc H Wadsworth II, Travis K Hughes, Gabriel K Griffin, Sofia Battaglia, Julia A Verga, Jason Stephansky, Timothy J Pastika, Jennifer Lombardi Story, et al. Single-cell rna-seq reveals aml hierarchies relevant to disease progression and immunity. *Cell*, 176(6):1265–1281, 2019.

[132] Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M Lanata, et al. Multiplexed droplet single-cell rna-sequencing using natural genetic variation. *Nature biotechnology*, 36(1): 89–94, 2018.

[133] Henna Kallionpää, Juhi Somani, Soile Tuomela, Ubaid Ullah, Rafael De Albuquerque, Tapio Lönnberg, Elina Komsi, Heli Siljander, Jarno Honkanen, Taina Härkönen, et al. Early detection of peripheral blood cell signature in children developing β-cell autoimmunity at a young age. *Diabetes*, 68(10):2024–2034, 2019.

[134] Can Liu, Andrew J Martins, William W Lau, Nicholas Rachmaninoff, Jinguo Chen, Luisa Imberti, Darius Mostaghimi, Danielle L Fink, Peter D Burbelo, Kerry Dobbs, et al. Time-resolved systems immunology reveals a late juncture linked to fatal covid-19. *Cell*, 184(7):1836–1857, 2021.

[135] Robrecht Cannoodt, W Saelens, H Todorov, and Y Saeys. Single-cell-omics datasets containing a trajectory. *Zenodo (Oct. 2018). DOI*, 10, 2018.

[136] Nikolaos Papadopoulos, Parra R Gonzalo, and Johannes Söding. Prosstt: probabilistic simulation of single-cell rna-seq data for complex differentiation processes. *Bioinformatics*, 35(18):3517–3519, 2019.

[137] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: simulation of single-cell rna sequencing data. *Genome biology*, 18(1):1–15, 2017.

[138] Robrecht Cannoodt, Wouter Saelens, Louise Deconinck, and Yvan Saeys. Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nature Communications*, 12(1):1–9, 2021.

[139] Alaa Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 2020.

[140] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21(1):1–13, 2020.

[141] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC bioinformatics*, 12(1):1–8, 2011.

[142] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[143] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3):e0118432, 2015.

[144] Max Schubach, Matteo Re, Peter N Robinson, and Giorgio Valentini. Imbalance-aware machine learning for predicting rare and common disease-associated non-coding variants. *Scientific reports*, 7(1):1–12, 2017.

[145] Maria K Jaakkola, Fatemeh Seyednasrollah, Arfa Mehmood, and Laura L Elo. Comparison of methods to detect differentially expressed genes between single-cell populations. *Briefings in bioinformatics*, 18(5):735–743, 2017.

[146] Charlotte Soneson and Mark D Robinson. Bias, robustness and scalability in single-cell differential expression analysis. *Nature methods*, 15(4):255–261, 2018.

[147] Marvin N Wright and Andreas Ziegler. ranger: A fast implementation of random forests for high dimensional data in c++ and r. *arXiv preprint arXiv:1508.04409*, 2015.

[148] Giuseppe Jurman, Roberto Visintainer, Michele Filosi, Samantha Riccadonna, and Cesare Furlanello. The him glocal metric and kernel for network comparison and classification. In

2015 IEEE international conference on data science and advanced analytics (DSAA), pages 1–10. IEEE, 2015.

[149] Paul Jaccard. Nouvelles recherches sur la distribution florale. *Bull. Soc. Vaud. Sci. Nat.*, 44: 223–270, 1908.

[150] Datasets-Single Cell Gene Expression-Official. 10x genomics support, 2021.

[151] Kazunaga Agematsu, Sho Hokibara, Haruo Nagumo, and Atsushi Komiyama. Cd27: a memory b-cell marker. *Immunology today*, 21(5):204–206, 2000.

[152] François Brinas, Richard Danger, and Sophie Brouard. Tcl1a, b cell regulation and tolerance in renal transplantation. *Cells*, 10(6):1367, 2021.

[153] Tobias Roider, Julian Seufert, Alexey Uvarovskii, Felix Frauhammer, Marie Bordas, Nima Abedpour, Marta Stolarczyk, Jan-Philipp Mallm, Sophie A Herbst, Peter-Martin Bruch, et al. Dissecting intratumour heterogeneity of nodal b-cell lymphomas at the transcriptional, genetic and drug-response levels. *Nature Cell Biology*, 22(7):896–906, 2020.

[154] Wilson KM Wong, Guozhi Jiang, Anja E Sørensen, Yi Vee Chew, Cody Lee-Maynard, David Liuwantara, Lindy Williams, Philip J O'Connell, Louise T Dalgaard, Ronald C Ma, et al. The long noncoding rna malat1 predicts human islet isolation quality. *JCI insight*, 4(16), 2019.

[155] Yingyao Zhou, Bin Zhou, Lars Pache, Max Chang, Alireza Hadj Khodabakhshi, Olga Tanaseichuk, Christopher Benner, and Sumit K Chanda. Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. *Nature communications*, 10(1):1–10, 2019.

[156] Décio L Eizirik, Alessandra K Cardozo, and Miriam Cnop. The role for endoplasmic reticulum stress in diabetes mellitus. *Endocrine reviews*, 29(1):42–61, 2008.

[157] Rachel C Bandler, Ilaria Vitali, Ryan N Delgado, May C Ho, Elena Dvoretskova, Josue S Ibarra Molinas, Paul W Frazel, Maesoumeh Mohammadkhani, Robert Machold, Sophia Maedler, et al. Single-cell delineation of lineage and genetic identity in the mouse brain. *Nature*, 601(7893):404–409, 2022.

[158] Lukas M Weber, Wouter Saelens, Robrecht Cannoodt, Charlotte Soneson, Alexander Hapfelmeier, Paul P Gardner, Anne-Laure Boulesteix, Yvan Saeys, and Mark D Robinson. Essential guidelines for computational method benchmarking. *Genome biology*, 20(1):1–12, 2019.

[159] Prasanta K Jana and Azad Naik. An efficient minimum spanning tree based clustering algorithm. In *2009 Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS)*, pages 1–5. IEEE, 2009.

[160] Xiaoping Han, Renying Wang, Yincong Zhou, Lijiang Fei, Huiyu Sun, Shujing Lai, Assieh Saadatpour, Ziming Zhou, Haide Chen, Fang Ye, et al. Mapping the mouse cell atlas by microwell-seq. *Cell*, 172(5):1091–1107, 2018.

[161] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017.

[162] Lijia Yu, Yue Cao, Jean YH Yang, and Pengyi Yang. Benchmarking clustering algorithms on estimating the number of cell types from single-cell rna-sequencing data. *Genome biology*, 23 (1):1–21, 2022.

[163] Edieal Pinker. Reporting accuracy of rare event classifiers. *NPJ digital medicine*, 1(1):1–2, 2018.

[164] Yuchen Yang, Ruth Huh, Houston W Culpepper, Yuan Lin, Michael I Love, and Yun Li. Safe-clustering: single-cell aggregated (from ensemble) clustering for single-cell rna-seq data. *Bioinformatics*, 35(8):1269–1277, 2019.

[165] Yue Cao, Pengyi Yang, and Jean Yee Hwa Yang. A benchmark study of simulation methods for single-cell rna sequencing data. *Nature Communications*, 12(1):1–12, 2021.

[166] Wei Zheng, Lisa M Chung, and Hongyu Zhao. Bias detection and correction in rna-sequencing data. *BMC bioinformatics*, 12(1):1–14, 2011.

[167] Michael I Love, John B Hogenesch, and Rafael A Irizarry. Modeling of rna-seq fragment sequence bias reduces systematic errors in transcript abundance estimation. *Nature biotechnology*, 34(12):1287–1291, 2016.

[168] Stijn Hawinkel, JCW Rayner, Luc Bijnens, and Olivier Thas. Sequence count data are poorly fit by the negative binomial distribution. *PloS one*, 15(4):e0224909, 2020.

[169] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[170] Manu Setty, Vaidotas Kiseliovas, Jacob Levine, Adam Gayoso, Linas Mazutis, and Dana Pe'Er. Characterization of cell fate probabilities in single-cell data with palantir. *Nature biotechnology*, 37(4):451–460, 2019.

[171] Ronald R Coifman, Stephane Lafon, Ann B Lee, Mauro Maggioni, Boaz Nadler, Frederick Warner, and Steven W Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the national academy of sciences*, 102(21): 7426–7431, 2005.

[172] Kushagra Pandey and Hamim Zafar. Inference of cell state transitions and cell fate plasticity from single-cell with margaret. *Nucleic Acids Research*, 50(15):e86–e86, 2022.

[173] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010.

[174] Mengqi Zhang, Si Liu, Zhen Miao, Fang Han, Raphael Gottardo, and Wei Sun. Ideas: individual level differential expression analysis for single-cell rna-seq data. *Genome biology*, 23(1):1–17, 2022.

[175] Simone Tiberi, Helena L Crowell, Lukas M Weber, Pantelis Samartsidis, and Mark D Robinson. distinct: a novel approach to differential distribution analyses. *bioRxiv*, pages 2020–11, 2020.

TURUN
YLIOPISTO
UNIVERSITY
OF TURKU

Painosalama, Turku, Finland 2023

ISBN 978-951-29-9392-5 (Print)
ISBN 978-951-29-9393-2 (PDF)
ISSN 2736-9390 (Painettu/Print)
ISSN 2736-9684 (Sähköinen/Online)