Title
Recognizing Cell Identity: Classifying cell types in scRNAseq data

Author
Xu, Chenling

Publication Date
2021

Peer reviewed|Thesis/dissertation

Recognizing Cell Identity: Classifying cell types in scRNAseq data

by

Chenling Xu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computational Biology

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Nir Yosef, Chair
Professor Sandrine Dudoit
Professor Chun Jimmie Ye
Professor Hernan Garcia

Spring 2021

Recognizing Cell Identity: Classifying cell types in scRNAseq data

Abstract

Recognizing Cell Identity: Classifying cell types in scRNAseq data

by

Chenling Xu

Doctor of Philosophy in Computational Biology

University of California, Berkeley

Professor Nir Yosef, Chair

The classification of cell type is one of the first steps in scRNAseq analysis for translating observed transcriptional variation to biological insights. The same cell types can be sampled from different environment and using different technologies and their transcriptional profile can differ. Thus, defining cell types in scRNAseq data is much more than a matter of identifying clusters of cells that are similar to each other. In chapter 1, we developed a simulation method SymSim in order to understand the different facets of variability in scRNAseq. In Chapter 2, we applied a Bayesian Variational Inference method scVI for the harmonization scRNAseq datasets and propose a new method scANVI in the same frame work for the annotation of these datasets. We tested the performance of scVI and scANVI using both SymSim and experimental data. In Chapter 3 we applied our data harmonization method scVI to a Multiple Sclerosis (MS) case-control study using scRNAseq data to profile immune cells. We identified cellular changes associated with MS in tissue-specific cell type abundance and transcriptional changes after being able to identify shared cell types in both blood and CSF in multiple donors. In Chapter 4 we apply a number of scRNAseq harmonization and annotation including scVI and scANVI to a large consortium cell atlas project Tabula Sapiens. Tabula Sapiens aims to provide a comprehensive reference scRNAseq dataset for the scientific community. We developed an automatic annotation pipeline named PopularVote to facilitate the in-house data annotation process, and to be published for using as a public tool for other scientists to annotate their own data. This dissertation presents a set of tools that we developed or used in cell type annotation in a diverse set of scRNAseq applications (identifying rare cell types, comparing cell types across conditions, generating automatic data annotations). The potential of scRNAseq is best realized by generating a well-curated dataset that everyone in the research community can use and contribute to, and the ability to classify cells in an automatic manner will enable such efforts in the future.

To everyone who contributed to this work through the following:
mentoring
collaborating
data generating
technical and logistical support
funding
educating
keeping the sanity of its author.

Contents

List of Figures

List of Tables

Acknowledgments

I am grateful to have had the privilege of spend six years of my life learning and doing research on public and private funding and hope that I will one day be able to pay back what the society has invested in me.

I am grateful for the patience and guidance from my advisor Nir Yosef in how to think rigorously and be my own critic. Nir has always taught with example through his excitement about science, dedication to teaching and care for the well-being of everyone in the lab. I am especially grateful to Romain Lopez, Xiuwei Zhang and Galen Xing with whom I worked with closely. I am also grateful to Allon Wagner, Zoë Steier, Anat Kreimer, Matt Jones, Alex Khodavarian, Adam Gayoso, Shaked Afik, Jim Kaminski, Tal Ashuach, Alyssa Morrow and all my other extremely talented lab mates for all their help and discussion. I am grateful to my collaborators Gerd Meyer zu Hörste and Angela Pisco for working with me on Chapter 3 and 4, and everyone who worked on generating the data used in my thesis. I would also like to thank Carlos Buen Abad Najar for getting through the first year together, Henry Pinkard for all the statistics learning and creative ideas, Dat Mai for being Dat. I am thankful to Emilia Huerta Sanchez for her support in my second year, and to Kate Chase for her patience and guidance in navigating the PhD program.

I am grateful to all my friends for their mental and emotional support and the adventures we've been on together, especially Daniel Friedman, Linqing Chen, Matejovičová, Muireann Spain and Storm Weiner. I am grateful for the Berkeley Student Cooperative for providing me with affordable housing, amazing food and a community to rely on. My PhD life would have been so different without meeting my friends from different fields of study and backgrounds. I don't have enough space to name everyone, but you have all taught me the value of community and sharing, and have given me a new sense of purpose in life.

A lot of people have generously brought me up to the place I was to be able to come to graduate school at Berkeley. First and foremost my parents 陈甦 and 徐平 who had to send their only child across the oceans. I would not be who I am today without all the sacrifices they made for my education or their inspiration to be a upright, persevering, hard-working and caring human being. The same goes for my grand parents 陈其猛, 陈杏芝, 徐玉生 and 曹秋凤. I am grateful to my host parents Sharon Halperin and Jacques Peureux for making me part of my second family. My undergraduate mentors Michael Turelli and Graham Coop had introduced me to computational biology research. They had trusted in my ability to do research when I showed up in their office as a naïve undergraduate who didn't know what computational biology was, and introduced me to the world of research. I am equally grateful to my other teachers from earlier in life and outside of science. This work is dedicated to everyone who had contributed to it directly or indirectly.

Preface

The discovery of cell types has long attracted the curiosity of generations of scientist. Cells are the basic functional unit in most living organisms [1]. Through the collaboration of a variety of cell types, tissue, organs and whole organisms are formed. Although cells in any multi-cellular organism can have vastly diverse morphology, metabolism and function, they all developed from the same zygote, thus sharing the same genome. Cell type diversity exists because different genes are activated in different cells. By studying the regulatory patterns of gene expression through RNA sequencing, we are able to reconstruct the pattern of cellular diversity in multi-cellular organisms.

Single cell RNA sequencing (scRNAseq) has become an essential tool in molecular and cellular biology in the past decade since the first single cell RNA sequencing proof-of-principal study was published [2] with only 8 cells. Although other single-cell resolution data was possible before scRNAseq (single cell qPCR [3], single molecular FISH [4]) and single cell microarray [5]), no other technology was able to capture full transcriptomics information on one single cell. scRNAseq allows us to perform massively parallel assays of cells and define new cell types. Due to the small amount of starting material, scRNAseq suffers from low sensitivity, high batch effect and other technical issues [6, 7]. On the other hand, scRNAseq has the advantage of producing a large amount of data that enables the use of rapidly advancing machine learning algorithms for the analysis of this data. These massive assays (thousands to millions of cells, thousands of genes) provide unprecedented amount of information for understanding cellular biology in a data-driven fashion. Numerous computational methods have since come up with innovative ways to map the diversity of single-cell transcriptional profile to functional cellular diversity. Cell type annotation is at the heart of all scRNAseq methods because all downstream analysis using single cell data depend on cell type annotation. Once cells are accurately annotated, scRNAseq data can be used in differential expression analysis, differential abundance analysis, alternative splicing and more.

Cell Type Concept

It is not straightforward to come up with a rigorous, data-driven definition of cell type. There are multiple theoretical frameworks of cell type definition [8], but the most relevant

one in the context of transcriptional regulation is the self-stabilizing regulatory programs [9]. Living organisms are dynamic, yet the definition of cell type implies a certain degree of persistence [10]. In Waddington's influential treatise of development "The strategy of the genes" [11], cell types are thought of as a result of canalization. The developmental landscape are "grooved by valleys, each leading to one of the normal end states". Cell types can therefore be thought of as a stable attractor state in a dynamic system [9]. The observed cell states can vary in different environment but they return to the attractor state. A particular interesting example in the immune system is tissue macrophages, which has two distinct origins (embryogenesis and differentiation from monocytes later in life) but share similar transcriptional profiles and carry out similar functions [12]. The concept of cell type is further complicated by the continuous nature of transcriptional variation in some cell types. One example is the different types of T helper cells that span the continuum of inflammotory state [13, 14].

The definition of cell type also often vary according to the technology used for their definition [15, 8]. For example a number of immune cells are named after their histology properties such as "basophil", "eosinophil" and "erythrocyte". Subsequently as flow cytometry became widely applied in immunology, immune cell types are delineated by surface protein markers that can be tagged by fluorochrome-conjugated antibodies [16]. As an example, T cells are separated into helper and cytotoxic T cells based on their expression of the CD4 and CD8 surface markers. scRNAseq as a high-throughput, whole genome assay will again change how cell types are defined.

The reader might ask why it is important to define cell types based on scRNAseq data, if the concept of cell type itself is elusive. scRNAseq can act like a bridge between the morphological and functional studies on the cell level and genetic studies on the gene level. The field of molecular biology has accumulated a deep understanding of the function of individual genes. The emergence of comprehensive scRNAseq datasets will finally allow us to map the gene-specific knowledge onto the cell- and tissue-specific domains of biology. For example, Genome-Wide Association Studies (GWAS) have discovered many disease-related genetic variants, but the disease mechanism remains obscure. scRNAseq has been used to uncover the cell-type-specific and disease-specific expression of genes associated with these variants [17].

scRNAseq Technology

scRNAseq has rapidly progressed from a laborious and specialized experiment to standard lab technique in the past decade. Most of these methods rely on multiplexing by cDNA tagging to increase the throughput of the sequencing experiments to hundreds or thousands of cells per experiment. Briefly, the process of scRNAseq include five major steps: (1) Cell dissociation or nuclei purification (2) Single cell or nucleus isolation (3) Reverse Transcription

cDNA amplification (4) Library construction and (5) Next generation sequencing. The core scRNAseq platform usually refer to step 2-4. Cells can be enriched if needed before input into the scRNAseq platform. After the experimental data generation, the short sequencing reads are mapped back to the reference transcriptome and a count matrix of dimension *number of cells* by *number of genes* is generated for computational analysis [18, 6, 19].

Different sequencing platforms vary in the cell isolation strategy, barcode addition method and sequencing technology [6, 18]. Single cell or nucleus can be isolated using FACS-sorting into plates [20, 21], microfluidic devices (Fluidigm C1 HT [22]), droplet-based (10X [23], inDrop [24], Drop-seq [25]) or microwells(Cytoseq [26]). Cells that are not amenable to regular dissociation protocols such as neuron, adipocytes, or cells in preserved tissue can be sequenced using single nuclei sequencing [27, 28]. The RNA molecules are then reverse-transcribed into cDNA. In this step, RNA from the single cells is tagged with a cell-specific barcode. Some protocols also tag each transcript with a unique molecular identifier (UMI) [29]. The barcoded cDNA is then pooled for preparing a next-generation sequencing library. Different methods also differs by whether they sequence the 3'-ends [25], 5'-ends [29] or full length of the transcript [30, 31, 32]. An additional sample barcode can be added to multiplex libaries on the same sequencing lane. Each experimental protocol comes with its advantages and drawbacks, and depending on the objective of the studies the users can choose the most appropriate protocol [18].

Quality Control for scRNAseq

On the computational side, almost scRNAseq dataset typically go through the following analysis pipeline (1) Quality Control (2) Feature selection and dimensionality reduction (3) Clustering and cell type annotation. More specific biological insights can be uncovered based on the results of the previously mentioned analysis with additional investigation. Two major analysis software Seurat [33, 34] and Scanpy [35] have integrated a number of tools in the R and Python environment respectively for these analysis and are the most widely used in the scRNAseq community. New methods are continuously being developed and either extends or integrates into these two common frameworks [36].

Besides QC applicable to other next-generation sequencing data, scRNAseq comes with unique technical challenges that require specialized computational methods. First we discuss the three technical artifacts as a result of the cell isolation. (1) RNA molecules can be released into the cell suspension during the isolation step. This results in highly expressed, cell-type-specific genes to be observed at low level in other cell types [37]. Computation methods can infer the decontaminated RNA counts by modeling the ambient RNA level with empty droplets (defined in (2)) [38], or treating each observed cells as a mixture of native expression and contamination [37]. (2) Not all observed cell barcodes correspond to viable cells. If a well or a droplet only contains ambient RNA in the cell lysis solution, or cell

fragments, these RNA molecules will still be tagged by a unique cell-specific barcode. The number of expressed genes, the total number of UMIs and the fraction of counts from the mitochondrial genome are the most common metric used to filter out barcodes that do not correspond to viable cells [19]. (3) Another technical artifact is named doublets, corresponding to the scenario that more than a single cell is tagged with the same barcode. A number of computational methods identify doublets using expression signatures that correspond to distinct clusters or cell types in the data [39, 40]. Other methods try to recover the single cell data from doublets by deconvolution [41].

At the sequencing step, a common artifact is barcode swapping. This happens on multiple patterned flow-cell illumina sequencing machines, including HiSeq4000, HiSeq X and NovaSea [42]. In multiplexed scRNAseq data, barcode swapping will cause sequences from different samples to share the same cell barcode and UMI. We can compute the probability that this happens by chance based on the number of unique UMI and cell barcode. If the number of shared barcode and UMI across samples is much higher than by chance, we can remove those reads. A study from 2018 [42] estimate that approximately 2.5% of reads were mislabelled between samples on HiSeq4000. This filtering is not possible without the use of UMI.

Normalization, Feature Selection and Dimensionality Reduction and Data Harmonzation

scRNAseq data are typically normalized by first correcting for cell size, and $log(x+1)$ transformed for visualization and methods that assumes normality [19]. Methods that explicitly models the stochasticity in count data such as scVI and scANVI [43, 14] does not require normalization.

Humans have around $20,000$ genes, which means that every scRNAseq dataset of human cells contains $20,000$ features. This means that any machine learning methods will suffer from the curse of dimensionality. In high dimensional space, similarity in Euclidean distance becomes meaningless because all samples are distant from their neighbors. The curse of dimensionality can be alleviated by both feature selection and dimensionality reduction. Feature selection aims to only keep genes that contribute to variation, often termed Highly Variable Genes (HVG) [44, 45, 46]. Methods used for dimensionality reduction and data harmonzation are discussed in detail in Chapter 2, thus will be omitted here.

Challenges and Methods in Cell Type Annotation

Cell type annotation is the step connecting the processed data to the downstream analysis. There are two main ways of annotating scRNAseq data: manual annotation and automatic

annotation. Manual annotation typically depend on unsupervised clustering of cells in a lower dimensional space. Most initial scRNAseq studies are annotated manually. This approach has led to many significant discoveries of new cell types and states in various study systems [45, 47, 48, 49, 32]. However this approach also has limitations. Many hyper-parameters used in dimensionality reduction and clustering affects the grouping of cells such as the number of dimension in the lower dimensional space and the number of neighbors used in construction k-nearest-neighbours similarity map [50]. Annotations generated this way might not be able to take into account the uncertainty in cluster assignment and lack reproducibility. In theory, manual annotation can also be derived directly from marker gene expression without clustering but due to the low sensitivity of scRNAseq [51, 52], many cells cannot be confidently assigned this way. Pooling cells [52, 53] or increasing the cluster resolution can ameliorate the problem of low sensitivity when annotating based on marker expression, but also increases the workload of manual annotation. The choice of granularity in manual annotation can vary from one study to another and make meta-analysis of multiple studies challenging.

Another challenge of cluster based annotation is the type of cell state variation. Although some cell types differ from each other in a discrete fashion, others vary more continuously. Developmental trajectory [54, 55, 56], certain T cell subsets [57, 58] are examples of continuously varying cells that are difficult to annotate using a clustering approach. Continuous variation is addressed by trajectory inference methods such as Monocle [55], Slingshot [59], RNA velocity [60] and many others [61]. In this thesis we only try to identify discrete clusters, and acknowledge that in some cases the boundaries we draw between two cell types can be arbitrary.

Recently, a plethora of automatic annotation methods have been developed and multiple evaluation efforts have been published [62, 63, 64]. The major advantages of automatic annotation are (1) it saves a lot of human time and (2) it preserves the naming conventions across datasets. There are two main classes of methods. The first class of method performs label transfer and annotates unannotated cells using annotated reference cells [14, 65, 66]. More general machine learning methods (SVM, LDA, RF, etc.) can also be used for label transfer by using the annotated dataset as training data and the unannotated dataset as testing data [64]. The second class of methods performs annotation based on curated marker lists [67] of known cell types [68, 69, 70]. Supervised annotation methods reduces the number of arbitrary choices of hyper-parameter associated with clustering, despite introducing its own hyper-parameters. The accuracy of the classifiers also depends on the accuracy of the reference data annotation. It is challenging to benchmark these annotation methods due to the lack of an agreed-upon metric and the fact that the best-performing method vary per dataset. However, with the use of a diverse evaluation panel and evaluation metric, researchers can make informed decisions about which methods to use in their own study [71].

Open-source Methods and Data

For scRNAseq to bring even more insights to the field of biology, we need to be able to integrate datasets from multiple replications, study systems, disease states etc. The sharing of scRNAseq data has been highly prioritized by the research community. scRNAseq is a resource intensive experimental procedure due to the high throughput of each experiment and the associated sequencing and computational costs. On the other hand, data generated from scRNAseq is extremely information rich. scRNAseq data can be explored in many ways when combined with other types of experimental data. For example, the transcriptional profile from scRNAseq data can be used to deconvolve bulk RNA sequencing data in order to infer cell type composition [72]. Most paper containing scRNAseq datasets do not exhaust the potential of the data. For example immune receptor reads can be used to reconstruct the B cell and T cell receptor sequences and further inform clonal relationships between immune cells [73, 74]. scRNAseq data can also reveal patterns of alternative splicing [75].

There have been many initiatives providing open-source, easily accessible datasets and analysis methods. One of the primary example is the Human Cell Atlas Project (HCA) [10]. HCA is committed to ensuring findable, accessible, interoperable and reusable (FAIR) data principles [76, 77]. This requires not only providing a repository of high quality data but also making sure that the analysis methods used are accurate, standardized and reproducible. HCA shares its data processing pipeline through a cloud computation pipeline [78] and supports a number of analysis tools. Another example is HuBMAP [79] and is also accompanied by its own cloud analysis pipeline Azimuth [80]. HCA, HuBMAP and another major data hosting consortium Single Cell Expression Atlas [81] are constructed with community-contributed data. Other databases such as Tabula Sapiens (Work in Progress, Chapter 4) and The adult human cell atlas [82] are produced by a coordinated sequencing effort. Many other data consortium focused on other model organisms and organs[47, 83, 84] or specific scientific questions such as disease mechanism [85, 86] are also providing important services to the scientific community. Chapter 4 discusses my contribution to open-source data and method development in the Tabula Sapiens project.

Outline of Dissertation Chapters

In this thesis we explore a few different aspects of cell type annotation. The first three chapters are co-authored publications and the fourth chapter is a manuscript in preparation. First we provide a simulation study of the noise structure of scRNAseq data in Chapter 1 [87]. This work is co-first-authored with Xiuwei Zhang. We separate the observed cell-to-cell variation in scRNAseq into intrinsic variation, extrinsic variation and technical noise. Simulation allow us to tune the amount of separation between different cell types, as well as the type of variation (continuous or discrete). Through the simulation we gained a better understanding of how the different sources of variation affects the data. We also gained

the ability to generate data with different levels of sensitivity and batch effect for testing algorithms' ability to recover biological variation, especially their ability to detect rare cell types. In Chapter 2, we developed a new method scANVI (Single-cell ANnotation using Variational Inference) that along with its predecessor scVI (Single-cell Variational Inference) can be used for data harmonization and label transfer between scRNAseq datasets [14, 43] that are sequenced by different labs, using different technologies or from different disease or environmental states. This work is co-first-authored with Romain Lopez. This study has confirmed the ability of Variational Inference methods to learn about shared biological variation in the presence of large batch effects without over-correction that can destroy biological signals. The label transfer abilities of scANVI provides a mechanism for automatically annotating cell types from a reference annotation, in effect transferring knowledge from existing studies to a new study in a data-driven way. In other words, we can discover which cells in one study a cell type in another study corresponds to. In Chapter 3[58], we apply the data harmonization ability of scVI to an immune dataset containing cells from two tissues (Cerebrospinal fluid, CSF; Peripheral Blood Mononuclear Cells, PBMC) from two disease states (Multiple sclerosis, MS; idiopathic intracranial hypertension, IIH). We discovered both tissue and disease specific changes in both cell type and transcript abundance. Without a reliable data harmonization method direct comparison of these datasets would not have been possible. The experimental work for this paper is led by David Schafflick (first author) and Gerd Meyer zu Hörste (corresponding author). In Chapter 4, we applied both the data harmonization and label transfer methods to the Tabula Sapiens project data and developed a new annotation pipeline named PopularVote. We aim to generate a reliable reference dataset as well as a easily accessible robust method for the scientific community to query their own data against the reference. Although the last two chapters are more data-focused, the code used for the analysis are all open source and versioned on Github.

Everything in this thesis would not have been possible without extensive collaboration both inside and outside of the lab, with collaborators who come from diverse backgrounds including statistics, computer science, molecular and cellular biology. scRNAseq provides a rich playground for scientists from many field to explore new ideas, develop new algorithms and discover new biological insights. The goal of this thesis is to show the discoveries we have made along the way, but more importantly it should be a road-map so that other scientists can use the tools that we have developed, as well as the data that we have generated in their own studies.

Future Directions of Single Cell Technology

Single cell technology continues to evolve. Recent advances in generating multi-modal single-cell resolution data include spatial transcriptomics that can capture both transcriptomic and spatial information [88, 89], CITE-seq that captures both protein and transcriptomic information [90], scATACseq [91, 92, 93] and scNMTseq [94, 95] that measures chromatin

accessibility data. As mentioned in the "Cell Type Concept", the technology available heavily influences the way cell types are thought about. These new assays will provide valuable insights and testable hypothesis for defining cell types. For example methylation happens at a slower time scale than transcription, and thus will allow us to better distinguish between transient cell states and stable cell types. New algorithms will also be needed for the integration of the multimodal data that will be generated.

Chapter 1

Simulating multiple faceted variability in single cell RNA sequencing: SymSim

1.1 Introduction

The advent of single cell RNA sequencing has led to a surge of computational and statistical methods for a range of analysis tasks. Some of the methods or the tasks that they perform have originated from bulk sequencing analysis, while others address opportunities (e.g., identification of new cell states [96, 97]) or technical limitations (e.g., limited sensitivity [98]) that are idiosyncratic to single cell genomics [99, 100]. While these computational methods are often based on reasonable assumptions it is difficult to compare them to each other and assess their performance without gold standards. One approach to address this is through simulations [52, 101, 102, 103, 104].

Existing simulation strategies (summarized by Zappia et al [105]) rely primarily on fitting distributional models to observed data and then drawing from these distributions. While the resulting models provide a good fit to observed data, their parameters are often abstract and do not directly correspond to the actual processes that gave rise to the observations. This leaves an important unaddressed problem in designing and using a simulator: the need to modulate and then study the effects of specific aspects of the underlying physical processes, such as the efficiency of mRNA capture, the extent of amplification bias (e.g., by changing the number of PCR cycles, or by using unique molecular identifiers [UMI]), and the extent of transcriptional bursting. To address this, we present SymSim (Synthetic model of multiple variability factors for Simulation), a software for simulation of single cell RNA-Seq data. SymSim explicitly models three of the main sources of variation that govern single cell expression patterns: allele intrinsic variation, extrinsic variation, and technical factors (**Figure1.1**). SymSim provides the users with knobs to control various parameters at these

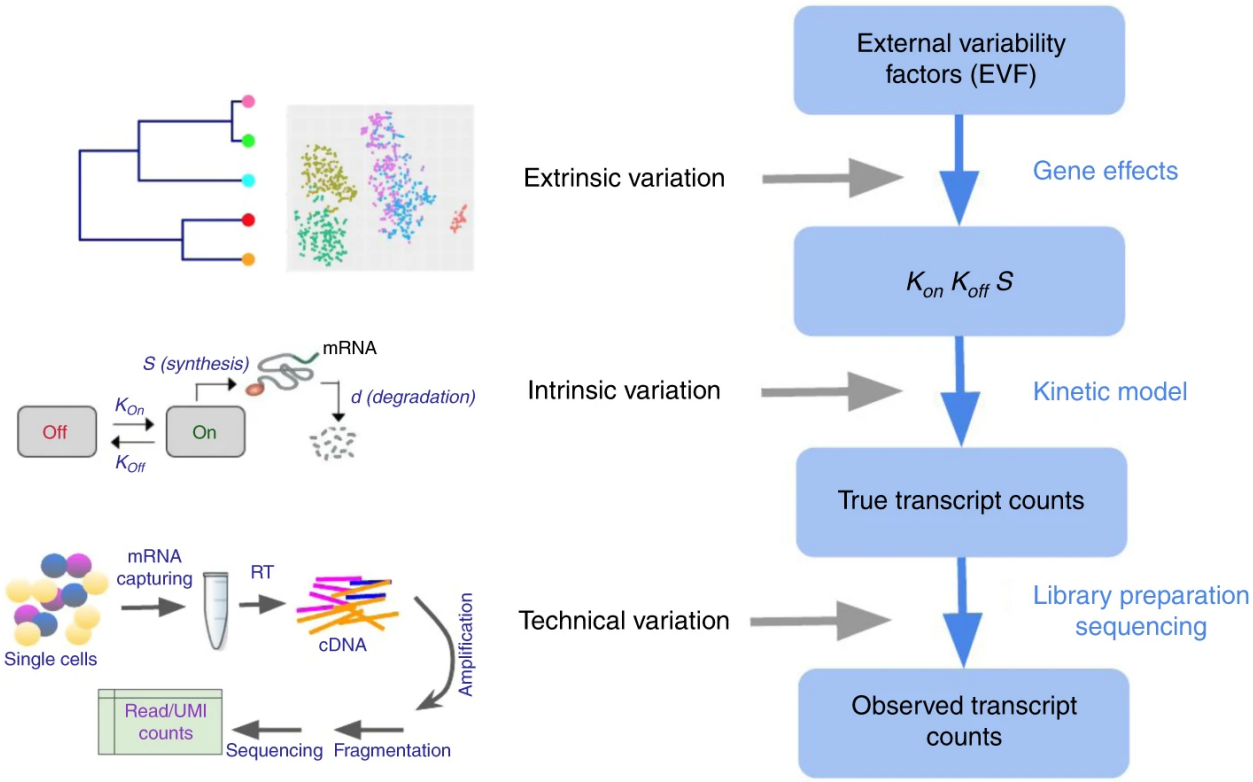
three levels. First, we generate true numbers of molecules using a kinetic model, which allows us to adjust allele intrinsic variation and the extent of burst effect; second, we provide an intuitive interface to simulate a sub-population structure, either discrete or along a continuum, through specification of cluster-trees, which define a low-dimensional manifold from which the transcriptional kinetics is determined for every gene and every cell; third, we simulate the main stages of the library preparation process and let users control the amount of variation stemming from these steps, such as capture efficiency, amplification bias, varying sequencing depth, and batch effect. Importantly, through this modeling scheme, SymSim recapitulates properties of the data (e.g., high abundance of zeros or increased noise in non-UMI protocols) without the need to explicitly force them as factors in a distributional model.

We demonstrate the utility of SymSim in two types of applications. In the first example, we use it to evaluate the performance of algorithms. We focus on the tasks of clustering, differential expression and trajectory inference, and test a number of methods under different simulation settings of biological separability and technical noise. In the second example, we use SymSim for the purpose of experimental design, focusing on the question of how many cells should one sequence to identify a certain sub-population.

Overview of SymSim

The true transcript counts, which are the number of molecules for each transcript in each cell at the time of analysis, are generated through the classical promoter kinetic model with parameters: promoter on rate (k_{on}), off rate (k_{off}) and RNA synthesis rate (s). The values of the kinetic parameters are determined by the product of gene-specific coefficients (termed gene effects) and cell-specific coefficients. The latter set of coefficients is termed extrinsic variability factors (EVF), and it is indicative of the cell state. The expected value of each EVF is determined in accordance to the position of the cell in a user-defined tree structure. The tree dictates the structure of the resulting cell–cell similarity map (which can be either discrete or continuous) since the distance between any two cells in the tree is proportional to the expected distance between their EVF values. For homogeneous populations (represented by a single location in the tree), the EVFs are drawn $i.i.d.$ from a distribution whose mean is the expected EVF value and variance is provided by the user. From the true transcript counts we explicitly simulate the key experimental steps of library preparation and sequencing, and obtain observed counts, which are read counts for full-length mRNA sequencing protocols, and UMI counts.

Figure 1.1: Overview of SymSim



Overview of SymSim.

The true transcript counts, which are the number of molecules for each transcript in each cell at the time of analysis, are generated through the classical promoter kinetic model with parameters: promoter on rate (k_{on}), off rate (k_{off}) and RNA synthesis rate (s). The values of the kinetic parameters are determined by the product of gene-specific coefficients (termed gene effects) and cell-specific coefficients. The latter set of coefficients is termed extrinsic variability factors (EVF), and it is indicative of the cell state. The expected value of each EVF is determined in accordance to the position of the cell in a user-defined tree structure. The tree dictates the structure of the resulting cell–cell similarity map (which can be either discrete or continuous) since the distance between any two cells in the tree is proportional to the expected distance between their EVF values. For homogenous populations (represented by a single location in the tree), the EVFs are drawn *iid* from a distribution whose mean is the expected EVF value and variance is provided by the user. From the true transcript counts we explicitly simulate the key experimental steps of library preparation and sequencing, and obtain observed counts, which are read counts for full-length mRNA sequencing protocols, and UMI count.

1.2 Results

Allele intrinsic variation

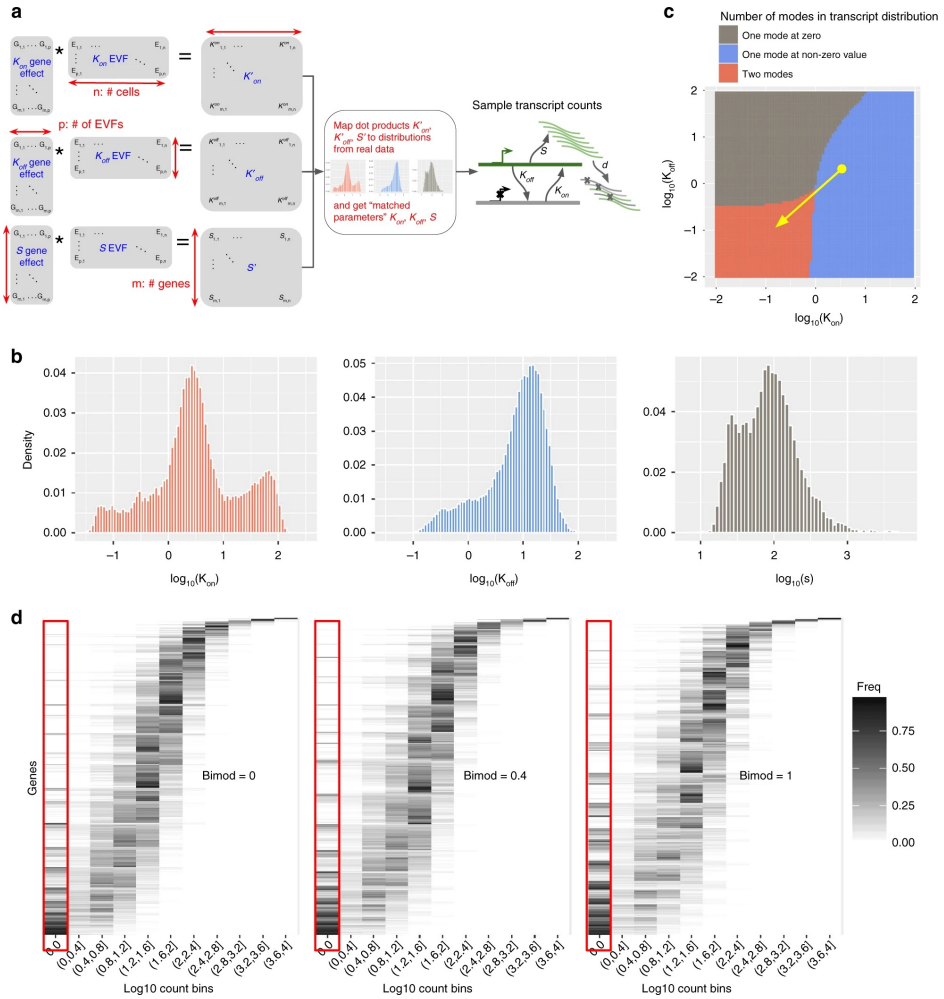
The first knob for controlling the simulation allows us to adjust the extent to which the infrequency of bursts of transcription adds variability to an otherwise homogeneous population of cells. We use the widely accepted two-state kinetic model, in which the promoter switches between an on and an off states with certain probabilities [106, 107]. We use the notation k_{on} to represent the rate at which a gene becomes active, k_{off} the rate of the gene becoming inactive, s the transcription rate, and d the mRNA degradation rate. For simplicity, and following previous work, we fix d to constant value of 1 [106, 108] and consider the other three parameters relative to d. Since RNA sequencing provides a single snapshot of the transcriptional process, we resort to assuming that the cells are at a steady state, and thus that the resulting single-cell measurements are drawn from the stationary distribution of the two-state kinetic model. Since d is fixed, we are able to express the stationary distribution for each gene analytically using a Beta-Poisson mixture [109] (**Methods**).

We show a diagram of how gene and cell-specific kinetic parameters are simulated from cell-specific EVF and gene-specific gene effect vectors, and how the kinetic parameters are used in a model of transcription in **Figure1.2**. The values of the kinetic parameters (k_{on}, k_{off} and s) are sampled for each gene in each cell using a product of cell-specific and gene-specific factors. Specifically, each cell is assigned with three low-dimensionality vectors (here, we use dimension 10), one for each kinetic parameter. The values inside the cell's vectors represent factors whose contribution is extrinsic to the noise generated intrinsically by the process of transcription (which we model by drawing from the stationary distribution above). These values, which we term extrinsic variability factors (EVF) represent a low dimension manifold that generates the data and can be interpreted as concentrations of key proteins, morphological properties, microenvironment and more. When simulating a homogeneous population, the EVFs of the cells are drawn from a normal distribution with a fixed mean of 1 and a standard deviation σ which is the within-populations variability parameter and can be set by the user (for the results in this section σ is set to 0.5). Each cell has a separate EVF vector for k_{on}, k_{off}, and s. Similarly, each gene is associated with three low-dimensionality vectors of a similar dimension. The values inside each gene's vectors can be interpreted as the dependence of its kinetics on the levels of EVFs. For instance, higher concentration of a certain EVF can give rise to a higher on rate of a certain promoter. We term these the gene effect vector. The gene effect values are first drawn independently from a standard normal distribution. We then replace each gene effect with a value of zero with probability η, thus ensuring that every gene is only affected by a small subset of EVFs. The sparseness parameter can be set by the user; in this paper we set η to a fixed value of 0.7. Each kinetic parameter is generated through two steps as shown in **Figure1.2(a)**: first, for each gene in each cell, we take the dot product of the corresponding EVF and gene effect vectors. Second, the dot product values are mapped to distributions of

parameters estimated from experimental data. The matched parameters are used to generate true transcript counts (see **Methods**). The distributions of k_{on}, k_{off} , and s that are used in SymSim for simulations are shown in **Figure1.2(b)**. These distributions are aggregated from inferred results of three subpopulations of the UMI cortex dataset (oligodendrocytes, pyramidal CA1 and pyramidal S1) after imputation by scVI and MAGIC. **Figure1.2(c)** is a heatmap showing the effect of parameter k_{on} and k_{off} on the number of modes in transcript counts. The value of s is fixed to 10 in this plot. The red area with low k_{on} and k_{off} have one zero mode and one non-zero mode. The gray area with low k_{on} and high k_{off} has only one zero mode, and the blue area with high k_{on} and low k_{off} have one non-zero mode. The yellow arrow shows how the parameter *bimod* can modify the amount of bimodality in the transcript count distribution.We show the effect of *bimod* on the transcript count distribution in **Figure1.2(d)**. Increasing *bimod* increases the zero-components of transcript counts and the number of bimodal genes. In these heatmaps, each row corresponds to a gene, each column corresponds to a level of expression, and the color intensity is proportional to the number of cells that express the respective gene at the respective expression level.

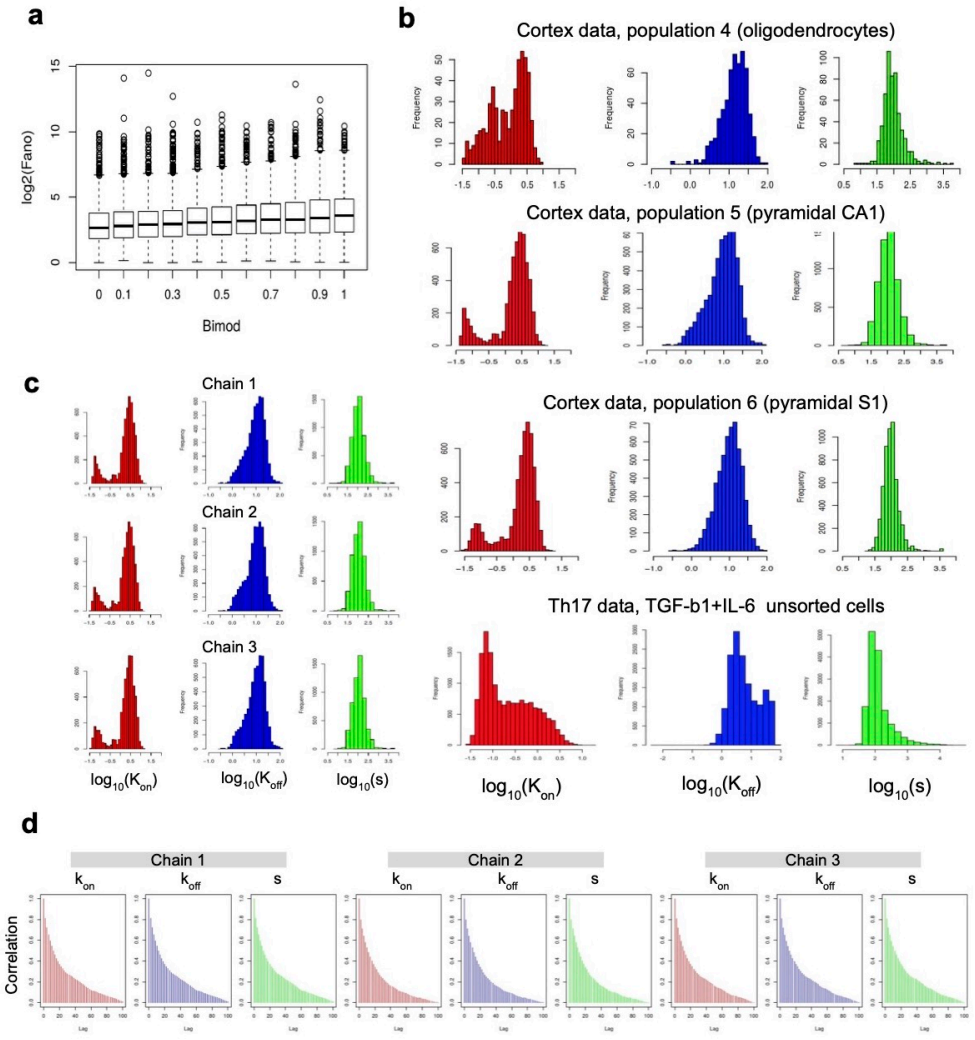
Finally, we account for the possibility of outlier genes with unusually high-expression level, commonly observed in real data. These outlier genes are hard to model with distributional methods, and require additional parametrization [105]. This phenomena is more pronounced in datasets from certain protocols (for example, 10x Chromium [45]) than others (for example, Smart-seq2 [110]), possibly due to selection bias which can be exacerbated by low capture rate. In SymSim, we model the high-expression outlier genes by designating a small subset of genes (whose proportion is determined by the parameter $prop_hge$) as constitutively transcribed, and adjusting their transcription rate s by a factor determined by the parameter $mean_hge$ (>1; **Methods**). To ensure that the parameters used for simulation fall into realistic ranges, we estimate the distribution of kinetic parameters of genes from real data (**Figure1.3**), and map the results of the dot product above to the distribution of the estimated parameters using a quantile approach (**Methods**). The estimation is done by fitting a Beta-Poisson distribution to imputed experimental counts. For this analysis, we used single cortex cells by Zeisel et al [111] as our data set and the software scVI [43] for imputation (**Methods**). The distributions of estimated parameters are shown in **Figure1.2(b)**. We performed the parameter estimation on sub-populations in the same dataset as well as another dataset of Th17 cells [112](these experimental datasets are described in **Methods**) and obtained similar distribution ranges (**Figure1.3(b)**). Importantly, the goal of this analysis is not to estimate the true parameter values (which may not be identifiable), but rather to identify the range of plausible parameter values, to be used for simulation. Furthermore, our estimations are in the same range with observations from experiments using smFISH [113, 114, 115, 116, 117, 118, 119] or transcription inhibition based [120] methods to measure kinetic parameters (**Methods**).

Figure 1.2: Effect of the kinetic parameters on transcript count distributions



(*a*) A diagram of how gene and cell-specific kinetic parameters are simulated from cell-specific EVF and gene-specific gene effect vectors, and how the kinetic parameters are used in a model of transcription. (*b*) The distributions of k_{on}, k_{off}, and s that are used in SymSim for simulations. These distributions are aggregated from inferred results of three subpopulations of the UMI cortex dataset (oligodendrocytes, pyramidal CA1 and pyramidal S1) after imputation by scVI and MAGIC. (*c*) A heatmap showing the effect of parameter k_{on} and k_{off} on the number of modes in transcript counts. The value of s is fixed to 10 in this plot. (*d*) Histogram heatmaps of transcript count distribution of the true simulated counts with varying values of bimod, showing that increasing bimod increases the zero-components of transcript counts and the number of bimodal genes. In these heatmaps, each row corresponds to a gene, each column corresponds to a level of expression, and the color intensity is proportional to the number of cells that express the respective gene at the respective expression level.

Figure 1.3: Kinetic model parameter estimation



(a) The Fano factor of genes across cells with different values for *bimod*. **(b)** Distribution of kinetic parameters estimated from different experimental data. **(c)** Distribution of kinetic parameters estimated for the same data (UMI cortex dataset, population pyramidal CA1, imputed with scVI) using different starting values to obtain multiple MCMC chains. **(d)** Auto-correlation diagnose plots for the three chains in (c). The correlation values are average over all genes. For each lag value k, correlation is calculated between x_t and x_{t+k}, where x_t is a sample from MCMC at step t.

An intriguing question in the analysis of single cell RNA-seq is the extent to which the conclusion drawn from the data (e.g., clustering) may be confounded by transcriptional bursting and transcriptional noise. SymSim provides a way to explore this. We first note that modality [107, 109] and extent of the intrinsic noise [107] in the expression of a gene in a homogeneous population of cells (i.e., cells with similar EVFs) can vary for the different ranges of k_{on}, k_{off} and s. Specifically, one can distinguish the following three types of gene-expression distributions by the number of inflection points in the smoothed density function: unimodal with highest frequency at 0 (no inflection point), unimodal with highest frequency at non-zero value (one inflection point), and bimodal (two inflection points). **Figure1.2(c)** shows the number of inflection points for different configurations of k_{on} and k_{off} with given $s = 10$. The red area with low k_{on} and k_{off} have one zero mode and one non-zero mode. The gray area with low k_{on} and high k_{off} has only one zero mode, and the blue area with high k_{on} and low k_{off} have one non-zero mode. The yellow arrow shows how the parameter bimod can modify the amount of bimodality in the transcript count distribution. This gives a clear correspondence between kinetic parameter configurations and types of gene-expression distributions. For example, when s is relatively large, we obtain bimodal distributions when k_{on} and k_{off} are smaller than 1 (< 0 on log scale).

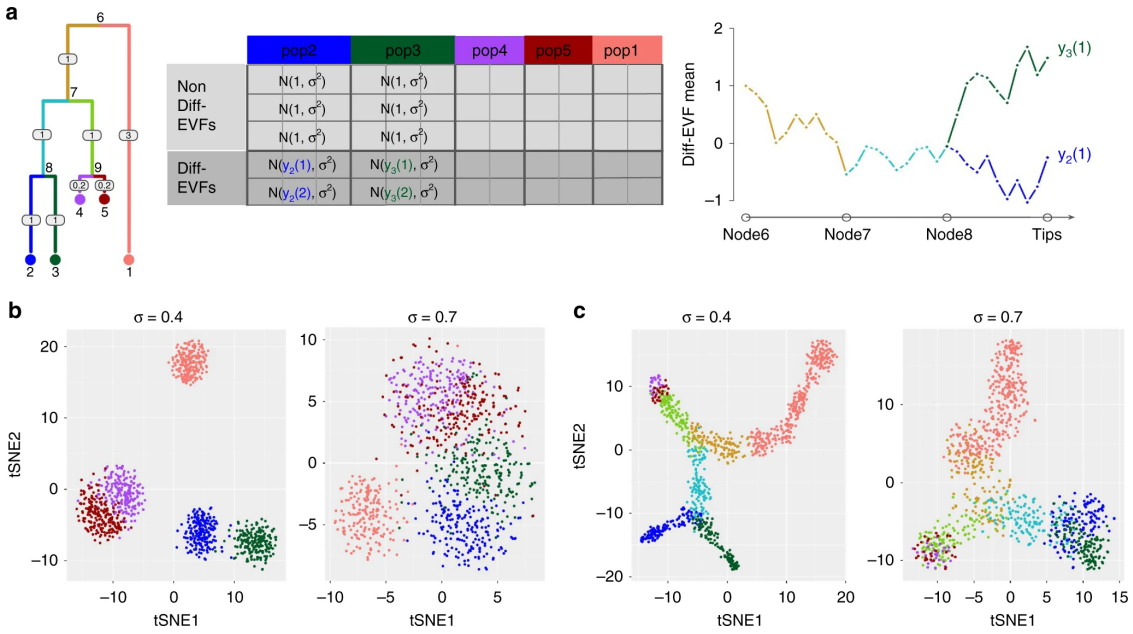
These results thus guide us in tuning kinetic parameters to obtain desired gene-expression distributions to simulate. Specifically, we focus on adjustment of the bimodality of the distribution, which can lead to large, yet transient fluctuations in gene expression at the same cell over time, thus potentially misleading methods for cell state annotation and differential expression. To increase the overall extent of bimodality in the data, we divide (decrease) all k_{on} and k_{off} values by 10^{bimod}, where the parameter *bimod* can take value from 0 to 1. This way, other properties such as burst frequency ($k_{on}/(k_{on} + k_{off})$) and synthesis rate ($s$) remain the same (**Figure1.2**). In **Figure1.2(d)**, we show a series of histogram heatmaps of the gene-expression distribution of all genes in a simulated homogeneous population while we increase *bimod*. When $bimod = 1$,we have a clearly increased number of bimodal genes. The increase of bimodality often gives rise to an increase in the fano factor which can be a measure of intrinsic variation (**Figure1.3(a)**).

Extrinsic variation via extrinsic variability factors

While the first knob focuses on variation within a homogeneous set of cells, the second knob allows the user to simulate multiple, different cell states. This added complexity is achieved by setting different EVF values for different cells, in a way that allows users to control cellular heterogeneity and generate discrete subpopulations or continuous trajectories.

To this end, SymSim represents the desired structure of cell states using a tree (which can be specified by the user), where every subpopulation (in the discrete mode) or every cell (in the continuous mode) is assigned with a position along the tree. The tree represents the relationship between cells. The numbers on the edges are branch lengths; the node numbers indicate the ID of the respective subpopulation (each subpopulation is represented by

Figure 1.4: Illustration of generating a diverse set of cell states with SymSim with extrinsic variation



(a) Illustration of generating a diverse set of cell states with SymSim. The tree represents the relationship between cells. The numbers on the edges are branch lengths; the node numbers indicate the ID of the respective subpopulation (each subpopulation is represented by a single position [leaf] in the tree). The matrix to the right depicts the derivation of EVF values. Each row corresponds to an EVF (only two are Diff-EVF), each column corresponds to a position in the tree, and the content specifies the distribution from which the EVF values are drawn. We use the notation ya(b) to represent the expected value of EVF b in position a in the tree. The rightmost plot depicts the derivation of these expected values with Brownian motion. We use subpopulations 2 and 3 as examples for both discrete cases (sampling only cells within the subpopulations) or continuous (sampling cells along the trajectories from the root progenitor state [node 6] to the two target subpopulations [nodes 2 and 3]).(b) tSNE plots of five discrete populations generated from the tree structure shown in a. Different values of σ give rise to different heterogeneity of each population. (c) tSNE plots of continuous populations generated from the same tree. The colors correspond to the colors on branches in the tree shown in a. When increasing σ, cells are more scattered around the main paths which follow the tree structure.

a single position (leaf) in the tree). Different positions in the tree correspond to different expected EVF values, and the expected absolute difference between the value of an EVF of any two cells is linearly proportional to the square root of their distance in the tree. When SymSim is applied in a discrete mode, the cells are sampled from the leaves of the tree. The set of cells that are assigned to the same leaf in the tree form a subpopulation, and their EVF values are drawn from the same distribution. As above, we draw these EVF from

a normal distribution, where the mean is determined by the position in the tree and the standard deviation is defined by the parameter σ. When SymSim is applied in a continuous mode, the cells are positioned along the edges of the tree with a small step size (which is determined by branch lengths and number of cells; Methods). The EVF values are then drawn from a normal distribution where the mean is determined by the position in the tree, and the standard deviation is defined by σ.

The matrix to the in the middle of **Figure1.4(a)** depicts the derivation of EVF values. Each row corresponds to an EVF (only two are Diff-EVF), each column corresponds to a position in the tree, and the content specifies the distribution from which the EVF values are drawn. We use the notation $y_a(b)$ to represent the expected value of EVF b in position a in the tree. Notably, SymSim only generates a subset of EVFs from the tree, while the remaining ones are drawn from the same distribution for all subpopulations, as shown in the matrix in **Figure1.4(a)**. The tree-sampled subset, which we term Diff-EVFs (Differential EVFs) represents the conditions or factors which are different between subpopulations, and they usually account for a small proportion of all the EVFs. The number of Diff-EVFs can be set by the user. The results in this section were produced with 60 EVFs, 20% of them are Diff-EVFs. With this formulation, users can control the extent of between-population variation by setting the branch lengths of the input tree, and combine it with a desired level of within-population variation by setting the parameter σ.

Take the $Diff-EVF1$ of populations 2 and 3 in **Figure1.4(a)** as an example: we can show that

$$E(|y_2(1) - y_3(1)|) = \sqrt{d_{23}} \cdot \sqrt{2/\pi}$$

where d_{23} is the distance in the tree between Populations 2 and 3. As the EVF values of $Diff-EVF1$ for cells in Populations 2 and 3 are sampled respectively from distributions $N(y_1(1), \sigma^2)$ and $N(y_1(2), \sigma^2)$ **Figure1.4**, the ratio $H = \frac{E(|y_2(1)-y_3(1)|)}{\sigma} = \frac{\sqrt{d_{23}} \cdot \sqrt{2/\pi}}{\sigma}$ correlates with the separability between cells from Population 2 and cells from Population 3.

Notably, both σ and the square root of branch lengths in the tree are in units of EVF values. For any given $Diff-EVF$ and any two given populations, the ratio of square root tree distance to σ determines the overlap between the distributions of the two $Diff-EVF$s. Thus this ratio determines the separability between the two populations. In the function of generating multiple discrete populations, users can control the extent of between-population variation by setting the branch lengths of the input tree, and control the within-population variation by parameter σ.

To facilitate the correspondence between EVF values and distances in the tree we use a Brownian motion procedure as described in [121]. Specifically, for each EVF we set the mean value at the root of the tree to a fixed number (default set root node to 1) and then perform Brownian motion along the branch. The rightmost plot in **Figure1.4(a)** illustrates this process using populations 2 and 3 in the tree as an example. Notably, in the continuous mode, this formulation can give rise to a rich set of patterns of changes in gene expression

from root (progenitor cells) to leaves (target cells). We use subpopulations 2 and 3 as examples for both discrete cases (sampling only cells within the subpopulations) or continuous (sampling cells along the trajectories from the root progenitor state [node 6] to the two target subpopulations [nodes 2 and 3]). As an alternative, we also implemented a mode for simulating continuous data by which gene expression from root to leaves is determined explicitly by an impulse function. This might be preferable if the user would like to generate smoother changes in gene expression, or specific temporal patterns. In the following analyses we use the Brownian motion model. As illustration, **Figure1.4bc** depicts the tSNE plots of cells from the same input tree with different σ in either a discrete (**Figure1.4(b)**) or continuous (**Figure1.4(c)**) mode. The colors correspond to the colors on branches in the tree shown in **Figure1.4**a. When increasing *sigma*, cells are more scattered around the main paths which follow the tree structure. Notably, both panels show that the tSNE plots reflect the structure of the input tree well.

The third knob: technical variation

A large part of the variation observed in scRNA-seq data sets stems from technical sources [122, 123, 124]. The technical confounders reflect noise, reduced sensitivity and bias that are introduced during sample processing steps such as mRNA capture, reverse transcription, PCR amplification, RNA fragmentation, and sequencing. In order to introduce realistic technical variation into our model, we explicitly simulate the major steps in the experimental procedures. We implemented two library preparation protocols: (1) full length mRNAs profiling without the use of UMIs (e.g., SmartSeq2 [110]); and (2) mRNA 3' end profiling with UMIs (e.g., 10x Chromium [23]). The former protocol is usually applied for a small number of cells and with a large number of reads per cell, providing full information on transcript structure [125]. The latter is normally applied for many cells with shallower sequencing, and it is affected less by amplification and gene length biases [122].

The workflow of these steps are shown in **Figure1.4(a)** (**Methods**). Starting from the simulated true mRNA content of a given cell (namely, number of transcripts per gene, sampled from the stationary distribution of the promoter kinetic model), the first step is mRNA capture, where every molecule is retained with probability . The value of the capture efficiency associated with each cell is drawn from a normal distribution with a mean and standard deviation , which can be set by the user. The second step is amplification, where in every cycle SymSim selects each available molecule with a certain probability and duplicates it. The expected amplification efficiency and the number of PCR cycles can be set by the user in a manner similar to that of capture efficiency. As an optional step, SymSim provides the option of linear amplification for the pre-amplification step (e.g., as in CEL-Seq [31]). We do not consider this option in this manuscript. In the third step each amplified molecule is broken down into fragments, in preparation for size selection and sequencing. Here, we assume that the average fragment length is 400bp (fragments are sequenced double-ended with read length 100bp) and use transcript lengths from the human reference genome. Fragments

that are within the acceptable size range (100 to 1000 bp). The number of reads per cell (namely, the number of sequenced fragments) is drawn from a Normal distribution whose mean is determined by the parameter *Depth*, which, along with the respective standard deviation can be provided by the user. To derive the final "observed" expression values we do not account for alignment errors, and assume that every sequenced fragment is assigned to the correct gene of interest. For the non-UMI option, we define the raw measurement of expression as the number of reads per gene. If UMIs are used, SymSim counts every original mRNA molecule only once by collapsing all reads that originated from the same molecule. Notably, the resulting distribution of number of reads per UMI is similar to the one observed in a dataset of murine cortex cells [111].

It has been previously shown that estimation of gene-expression levels from full length mRNA sequencing protocols has amplification biases related to sequence-specific properties like gene length and GC-content [122, 126], whereas the use of UMIs can correct these biases [7, 126] In particular, we have observed a negative correlation between gene length and length-normalized gene-expression in our reference SmartSeq2 dataset (murine Th17 cells from Gaublomme et al [112]; **Figure1.5(b)**), and the same trend is reported by Phipson et al [126]. To account for that, we parametrize the efficiency of the PCR amplification step using a linear model that represents gene length bias, and other sequence-specific factors (Methods). The effect of this addition can be controlled by a user defined $MaxAmpBias$ parameter (Methods). As a result, our simulated data with a non-UMI protocol shows a similar dependence of gene-expression on gene length and as in experimental data (**Figure1.5(b)**, real data is from [112]). In cases where UMIs are used gene length effects are not modeled and thus there is no observable gene length bias in the simulated data, similarly to the experimental data (Figure S2c, real data is from [111]). In **Figure1.5(c)**, we show the effect of simulated technical variation obtained through: (1) non-UMI, good parameters ($\alpha = 0.2, MaxAmpBias = 0.1, Depth = 1e6$) for high quality data; (2) UMI, good parameters ($\alpha = 0.2, MaxAmpBias = 0.1, Depth = 5e5$) for high quality data; (3) non-UMI, bad parameters ($\alpha = 0.05, MaxAmpBias = 0.2, Depth = 1e6$) for low quality data; (4) UMI, bad parameters ($\alpha = 0.04, MaxAmpBias = 0.2, Depth = 5e5$) for low quality data. We then compare our simulation with real data in **Figure1.5(c,d)**. We generated with parameters which best match the input experimental counts for three different datasets (non-UMI Th17, UMI cortex and UMI 10x t4k datasets). First we compared 2D transcript counts histogram heatmaps (**Figure1.5(c)**) then

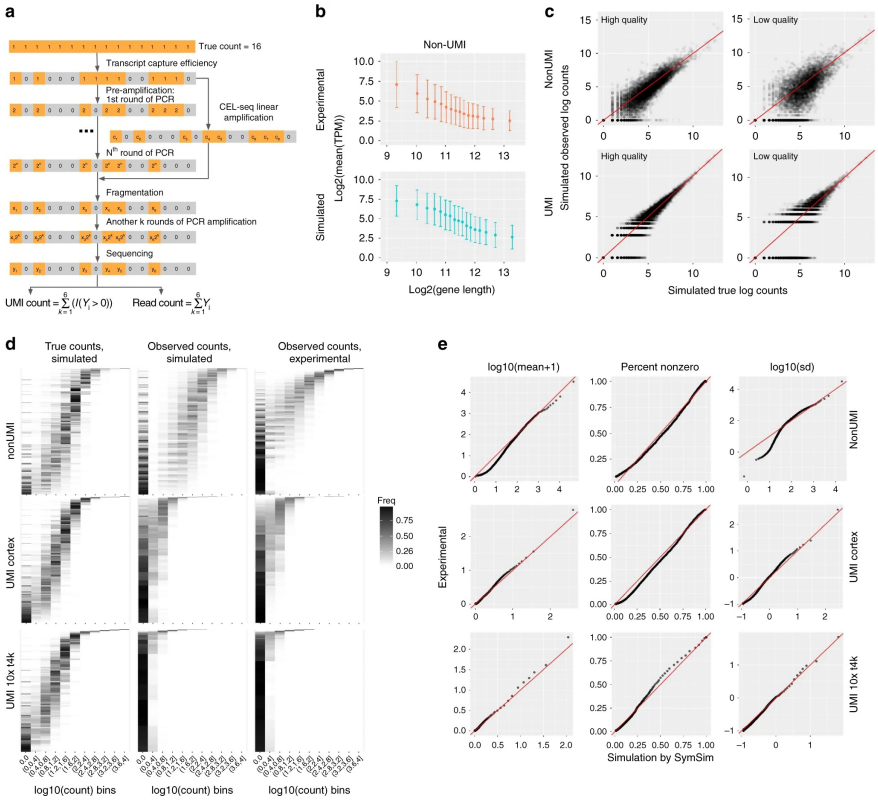
In **Figure1.4(c)**, we show the comparison between the simulated true mRNA content of one cell and the observed counts obtained with or without UMI. We consider two scenarios - the first scenario represents a study with a low technical confounding and the second one represents a highly confounded dataset. Parameters which differ between these "good" and "bad" cases in this example include capture efficiency (α), extent of amplification bias ($MaxAmpBias$) and sequencing depth ($Depth$). Using "bad" technical parameters introduce more noise to true counts, and compared to the nonUMI simulation the UMIs reduces tech-

nical noise. The histogram heatmaps of true counts and four versions of simulated counts are shown in **Figure1.3(a)**. The quantile-quantile plots (Q-Q plots) in **Figure1.3(b)** show that the UMIs help in maintaining a better representation of the true counts in the observed data.

Fitting parameters to real data

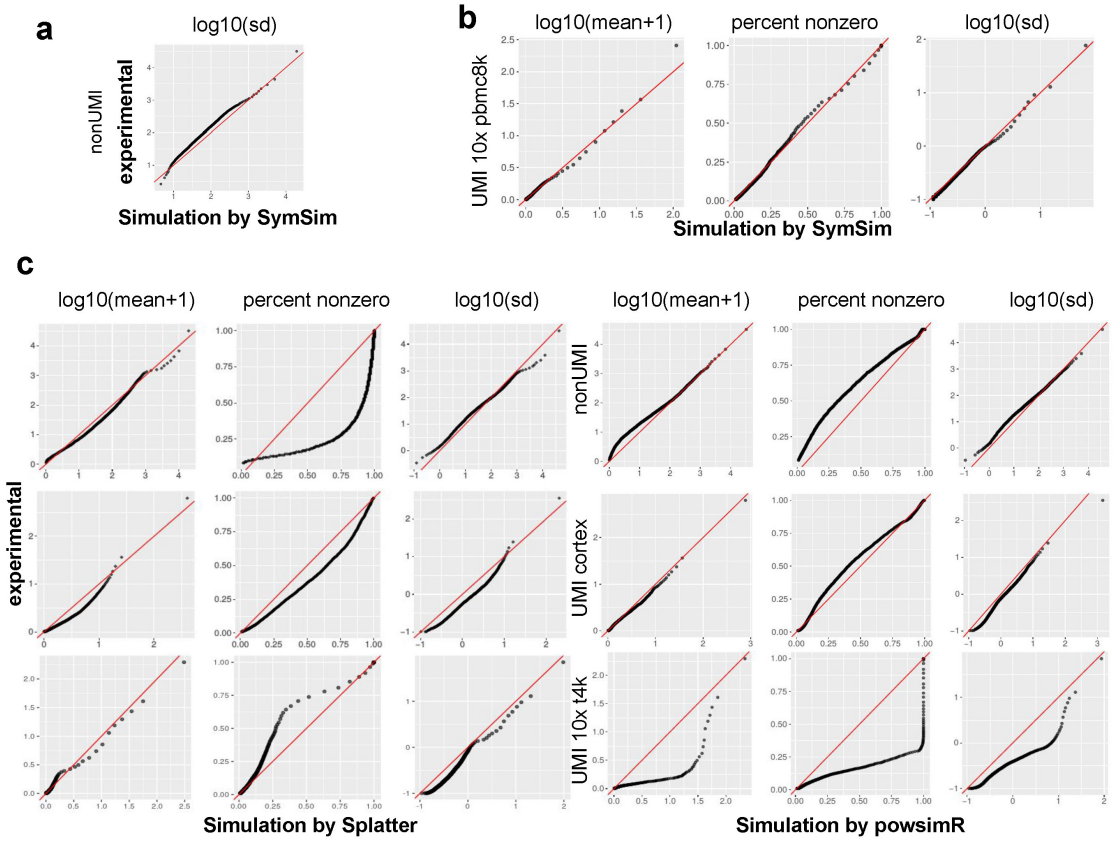
For a given real data set, SymSim can produce observed (read or UMI) counts which have similar statistical properties to the real data (**Figure1.4de**), by searching in a database of simulations obtained from a range of parameter configurations. This procedure focuses on within-population variability (Similarly to Splatter [105]) and sets the values of eight parameters from both the first and third knobs. We test this function with the non-UMI Th17 dataset [112] (using a subpopulation of 130 TGF$^{-}\beta$1^{-}IL-6 unsorted cells) and the cortex dataset [111] (using a subpopulation of 948 CA1 pyramidal neuron cells).

Figure 1.5: Simulating Technical Variation



(*a*) A diagram showing the workflow of adding technical variation to true simulated counts. Each gray or orange square represents a molecule of the same transcript in one cell. We implement the following steps: mRNA capturing, pre-amplification (PCR or linear amplification of the cDNAs), fragmentation, amplification after fragmentation, sequencing, and calculation of UMI counts or read counts. Details of these steps can be found in Methods. (*b*) Gene length bias in both simulated and experimental data for the non-UMI protocol. Error bars represent the ranges of (mean-SD, mean+SD), where SD means standard deviation. (*c*) Scatter plots comparing true counts and observed counts. (*d*) 2D transcript counts histogram heatmaps for three experimental datasets comparing simulated true counts, simulated observed counts and experimental observed counts. (*e*) Q–Q plots comparing the mean, percent non-zero and standard deviation in experimental counts and SymSim simulated observed counts. A good match is indicated by most of the dots falling close to the red line.

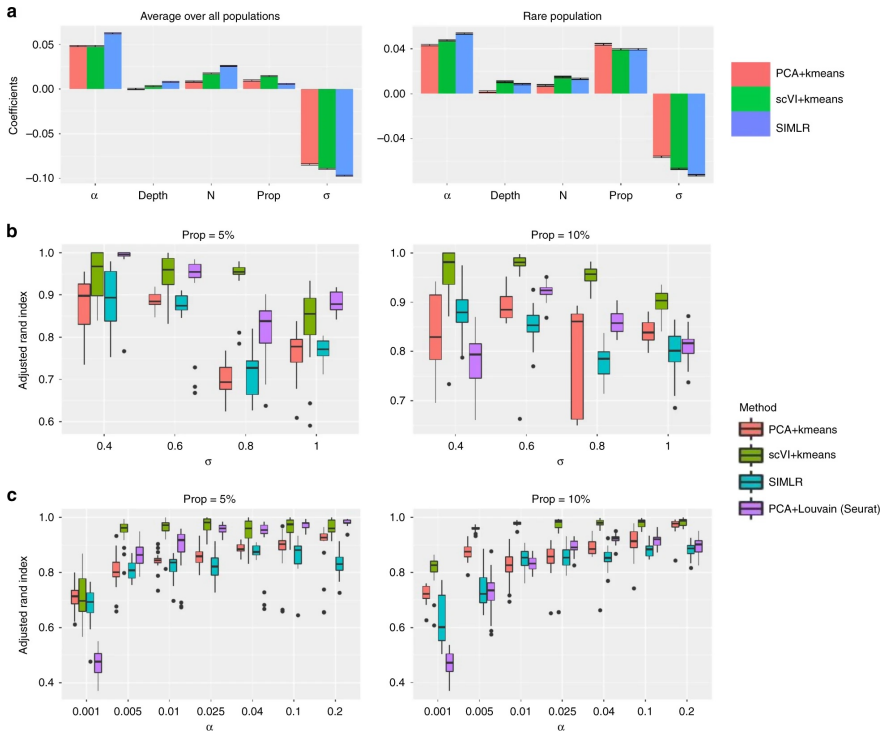
Figure 1.6: Summary Statstic Matching Q-Q Plot



Summary Statstic Matching Q-Q Plot

(a) The distribution of number of reads per UMI sequenced in the cortex dataset. This plot comes from the supplementary material of the original paper [111]. (b) The distribution of number of reads per UMI sequenced in our simulated data, when using the UMI protocol, with 100k reads per cell. (c) The gene length bias in observed counts with UMI protocol, respectively from experimental and simulated data. (d) TSNE plot of cells simulated for one homogeneous population in two batches. (e) The histogram heatmap of gene expression of true simulated counts, and observed simulated counts under "good" and "bad" parameter settings. The parameters are the same as described in Figure 4c. In these heatmaps, each row corresponds to a gene, each column corresponds to a level of expression, and the color intensity is proportional to the number of cells that express the respective gene at the respective expression level. (f) Q-Q plots of gene expression of true simulated counts and observed simulated counts under "good" and "bad" parameter settings.

Figure 1.7: Benchmarking of clustering methods using SymSim



Benchmarking of clustering methods

(a) Coefficients of various parameters from multiple linear regression between parameters and the adjusted Rand index (ARI). In the left plot the ARI are averaged over all populations, and in the right plot the ARI is only for the rare population (population 2). (b) ARI of the rare populations using the four clustering methods when changing σ ($\alpha = 0.04$). Left plot: the rare population accounts for 5% of all the cells; right plot: the rare population accounts for 10% of all the cells. (c) ARI of the rare populations using the four clustering methods when changing α ($\sigma = 0.6$). Left plot: the rare population accounts for 5% of all the cells; right plot: the rare population accounts for 10% of all the cells.

Figure1.3(d) shows the histogram heatmaps of true mRNA levels (simulated) and observed counts (simulated and experimental) for the non-UMI and UMI datasets. For a more quantitative comparison, in **Figure1.3(e)**, we show Q-Q plots of the distributions of mean, percent-non-zero and standard deviation of genes between simulated and experimental data. In general, **Figure1.3de** show that SymSim can output simulated observed counts which match the real data well for both UMI and non-UMI protocols. It is challenging to match the standard deviation (*sd*) between simulated and real data at the lower end, as they correspond to some lowly expressed genes (although they account for only a very small proportion of genes) whose variation are usually noisy. For example, for the non-UMI protocol, the lower end unmatched part is likely caused by a set of lowly expressed genes which have lower variation than the majority of genes at the same expression level (Figure S3c). Notably, we generated the same Q-Q plots after training Splatter [105] with the same experimental datasets as input, and found that SymSim matches this data significantly better (**Figure1.6**). We further inspected the relationship between mean (across all cells) and detection rate (fraction of cells in which the gene is detected) from the SymSim simulations, and get the same relationship as in experimental.

Using SymSim to evaluate the performance of computational methods

SymSim can be used to benchmark methods for single cell RNA-Seq data analysis as it provides both observed counts and a reference ground truth. In this section we demonstrate the utility of SymSim as tool for benchmarking methods for clustering and differential expression in a heterogeneous sample, consisting of multiple subpopulations (using the structure depicted by the tree in Figure 3a). The design of SymSim allows us to evaluate the effect of various biological and technical confounders on the accuracy of downstream analysis. Here, we investigate the effect of total number of cells (N), within population variability (σ), mRNA capture rate (α) and sequencing depth ($Depth$). We also test the effect of the proportion of cells associated with the smallest sub- population of cells (Prop), using population number 2 in the tree as our designated "rare" sub- population.

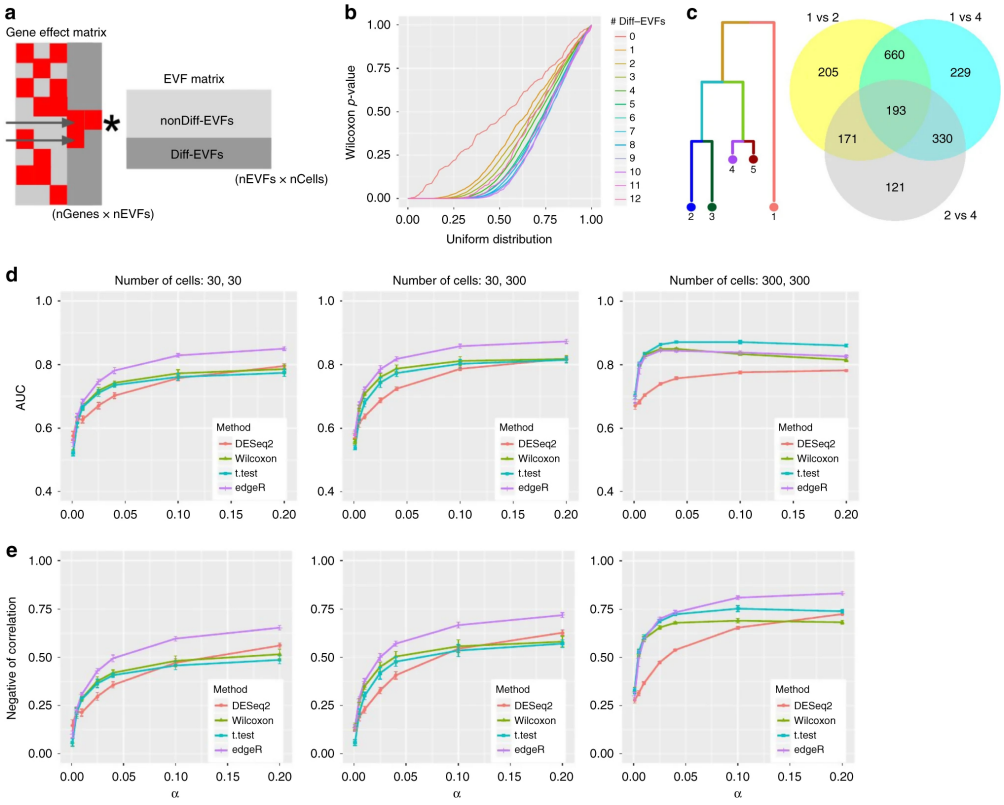
We begin by inspecting the impact of each parameter on the performance of clustering methods. To this end, we simulated observed counts using the UMI option, and traversed a grid of values for the five parameters with 18 simulation runs per configuration. The values of the remaining parameters are determined according to the cortex dataset [111] except for that we have kept the standard deviation (SD) of α and $Depth$ small so that the SD do not dominate the mean in cases of low α and $Depth$ values. We tested three clustering methods: k-means based on euclidean distance of the first 10-principle components, k-means based on euclidean distance in a nonlinear latent space learned by scVI [43] and SIMLR [127]. In all cases we set the number of clusters to the ground truth value (n=5). The accuracy of

the methods is evaluated using the adjusted Rand index (ARI; higher values indicate better performance). To inspect the effects of the various parameters on clustering performance, we performed multiple linear regression between the parameters and the ARI. The regression coefficients are shown in Figure 5a. Overall, σ appears to be the most dominant factor, and the proportion of the rare population (Prop) is clearly positively associated with better performance. Among the technical parameters, while α plays a role on the performance especially for the rare population, the impact of *Depth* is minor.

Focusing on the dominant factors (except N, which we discuss in the next section), provides the expected results, with better accuracy as the quality of the data or the differences between sub-populations increase (**Figure1.7bc**). Interestingly, comparing $\sigma = 0.1$ and $\sigma = 0.8$, we can tell that when σ is high enough to make the clustering challenging, further increasing σ does not yield obvious changes (data not shown). We observe a similar trend of saturation, inspecting increasing levels of capture efficiency (α), especially with scVI. Comparing the methods to each other, we see that scVI has the highest ARI in most cases and that PCA and SIMLR are comparable with SIMLR being slightly better when the rare population accounts for 5% of all cells and the other way around when the size of rare population increases to 10% of all cells.

Our mechanism of simulating multiple populations automatically generates differentially expressed (DE) genes between populations (in the discrete setting; **Figure1.4(b)**) or along pseudotime (in the continuous setting; **Figure1.4(c)**). In the following, we use SymSim to benchmark methods for detecting DE genes, focusing on the discrete setting. We use two criteria to define the ground truth set of DE genes. The first criterion is that the number of Diff-EVFs that are associated with a non-zero gene effect value (which we denote as $nDiff - EVFgene$; **Figure1.8(a)**) should be larger than zero. This criterion is motivated by our model of transcription regulation: the kinetic parameters of a gene are affected by extrinsic factors, and changes to extrinsic factors might therefore lead to changes in the number of transcripts. In **Figure1.8(a)** how DE genes are generated through the Diff-EVFs. Red squares in the gene effect matrix correspond to non-zero values. The two genes indicated by the arrows are DE genes by number of Diff-EVFs they have (respectively, 2 and 1). Indeed, when we compare the true simulated gene expression values between subpopulations (i.e., before introducing technical confounders), we get a uniform (random) distribution of p-values for genes with no Diff-EVFs, and an increasing skew as $nDiff - EVFgene$ increases (**Figure1.8(b)**, using Wilcoxon test). The numbers of Diff-EVFs used by genes can be thought of as the degree of DE-ness. Genes with more Diff-EVFs have p-values further diverged from uniform distribution. As expected, the log fold change of gene-expression between subpopulations increases with $nDiff - EVFgene$ (data not shown). An additional constraint for a gene being differentially expressed is that it must have a sufficiently large fold change in their simulated true simulated expression levels (threshold of absolute $log2$ fold change ranges from 0.6 to 1). The DE genes are determined by log2 fold change (LFC) of true counts with criterion $|LFC| > 0.8$ (Methods).

Figure 1.8: Benchmarking of DE detection methods



(*a*) Illustration of how DE genes are generated through the Diff-EVFs. (*b*) Q–Q plot comparing the p-value obtained from differential expression analysis between subpopulations 2 and 4 (using Wilcoxon test on the true simulated counts) to a uniform distribution. Genes are grouped by the number of Diff-EVFs and plotted in different color lines. (*c*) Venn diagram showing that closely related populations have less DE genes between them compared to distantly related populations. (*d*) The AUROC (area under receiver operating characteristic curve) of detecting DE genes using four different methods from observed counts with changing capture efficiency α (σ=0.6). The populations under comparison are 2 and 4. Three sets of criteria were used to define the true DE genes and the final performance was the average performance from the three sets: (1) $nDiff - EVFgene > 0$ and $|LFC| > 0.6$; (2) $nDiff - EVFgene$>0 and $|LFC| > 0.8$; (3) $nDiff - EVFgene > 0$ and $|LFC| > 1$. LFC was calculated with theoretical means from the kinetic parameters. (*e*) The negative of correlation between log2 fold change on theoretical mean of gene-expression and p-values obtained by a DE detection method, with changing capture efficiency α ($\sigma = 0.6$). The populations under comparison are 2 and 4.

An important distinguishing feature of SymSim is that it is capable of generating case studies for differential expression analysis that consist of multiple sub-populations, with a predefined structure of similarity. To illustrate this, consider populations 1, 2, and 4 (**Figure1.8(c)**), which are form a small hierarchy (2 and 4 are closer to each other and similarly distant from 1). This is reflected in the sizes of the sets of DE genes, obtained respectively from populations 1 vs 2 (1212 genes), 1 vs 4 (1204 genes) and 2 vs 4 (680 genes). The first two sets are larger than the third one, and there is a big overlap between the first two sets.

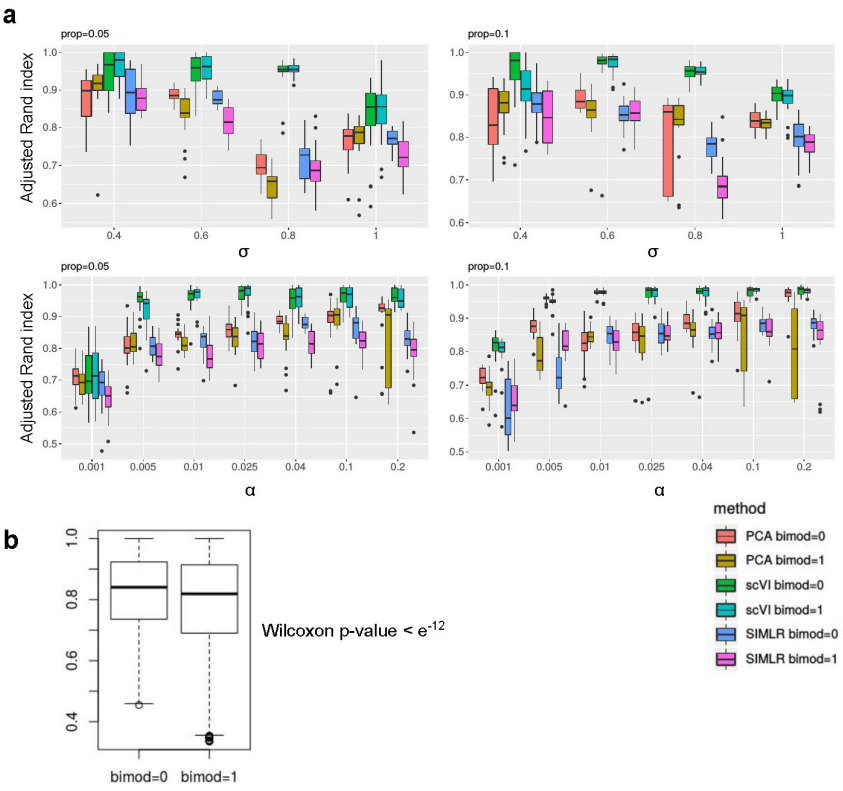
As an example for a benchmark study, we used four methods to detect DE genes: edgeR [128], DESeq2 [129], Wilcoxon rank-sum test and t-test on observed counts generated by various parameter settings (Methods). As above, we tested the effect of the total number of cells (N), within population variability (σ), and mRNA capture rate (α) with 10 simulation runs per parameter configuration. We use two accuracy measures: a) AUC (area under curve) of the ROC (receiver operating characteristic) curves obtained by treating the p-values output from each method as a predictor (Figure 6d); b) negative of Spearman correlation between the p-values of each detection method and the LFC-Theo (**Figure1.8(e)**).

From **Figure1.8de**, one can observe that when the numbers of cells are small (30 in each population), edgeR has the best performance while the other three methods are comparable to each other. When the numbers of cells increase to 300, the two naive methods Wilcoxon test and t-test tend to improve in their relative performance, compared to edgeR and DESeq2. When increasing capture efficiency, all methods gain performance except for the case of AUC with 300 cells. In that case, the drop of AUC for some methods is caused by inflation in p-values as α increases, which results in lower specificity. Finally, we noticed that the adjusted p-values from DESeq2 can have a lot of NAs especially when α is low (and thus counts are low), so we use its p-values in **Figure1.8de**. On the other hand, the assignment of NAs filters out genes which do not pass a certain threshold of absolute magnitude (explained in DESeq2 vignette [130]). To make use of this filtering, we conducted an additional analyses where we used the adjusted p-values for DESeq2 and compare it to to all other methods using only the non-filtered (non NA) genes. As expected, the performance of all methods (and specifically DESeq2) is improves when considering only this set of genes, and converges to high values already at lower capture efficiency rates.

To summarize, we find that edgeR has the best overall performance, with the t-test rank second followed by Wilcoxon test. This ranking is consistent with results from a recent paper which evaluated 36 methods for DE analysis with single cell RNA-Seq data (e.g., as in [131]).

We also investigate the effects of bimodality (controlled by parameter *bimod*) on the performance of clustering and differential expression algorithms. We increased the parameter bimod from 0 to 1 for the same datasets used in the analysis of Section "Using SymSim to evaluate clustering methods" and performed clustering. The comparison of clustering results between different methods and different values of *bimod* is shown in **Figure1.9**. We see that

Figure 1.9: Effects of the *bimod* parameter on the performance of clustering method

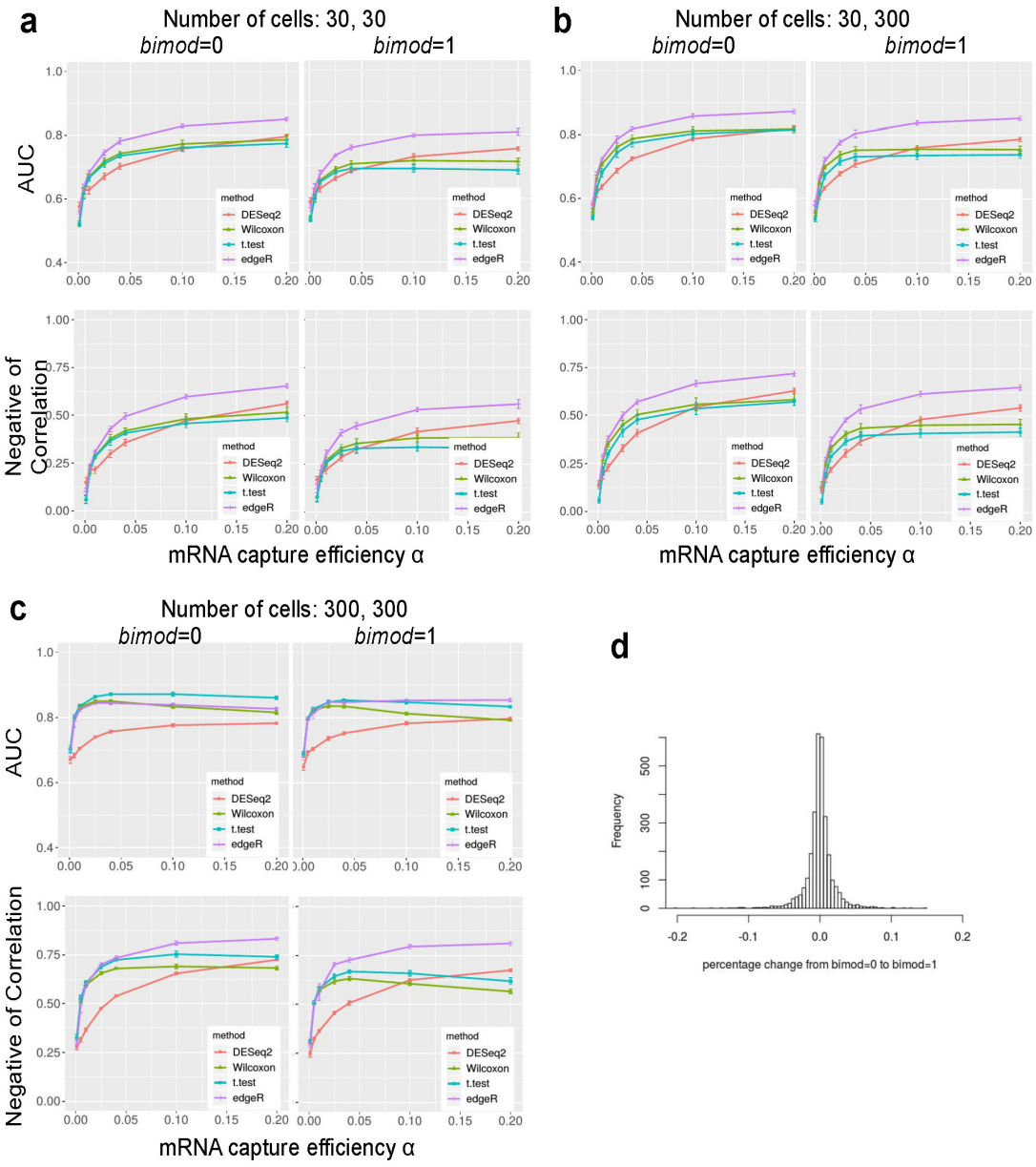


bimod's effect on clustering algorithms

(a) Top row: ARI of the rare populations using the three clustering methods (PAC, scVI and SIMLR) with different values of *bimod*. Left: the rare population accounts for 5% of all the cells; right: the rare population accounts for 10% of all the cells. (b) Boxplots of aggregated ARI values, separated only by different values of *bimod*.

in most cases there is a decrease of performance for the same method with increase of bimod. We then aggregate all the values of adjusted Rand index for all methods and all the parameters of α and σ, but only group them by the bimod value, and performed Wilcoxon test between the two groups of values. **Figure1.9(b)** shows that the difference between these two groups is small but significant (p-value $< e^{-12}$).

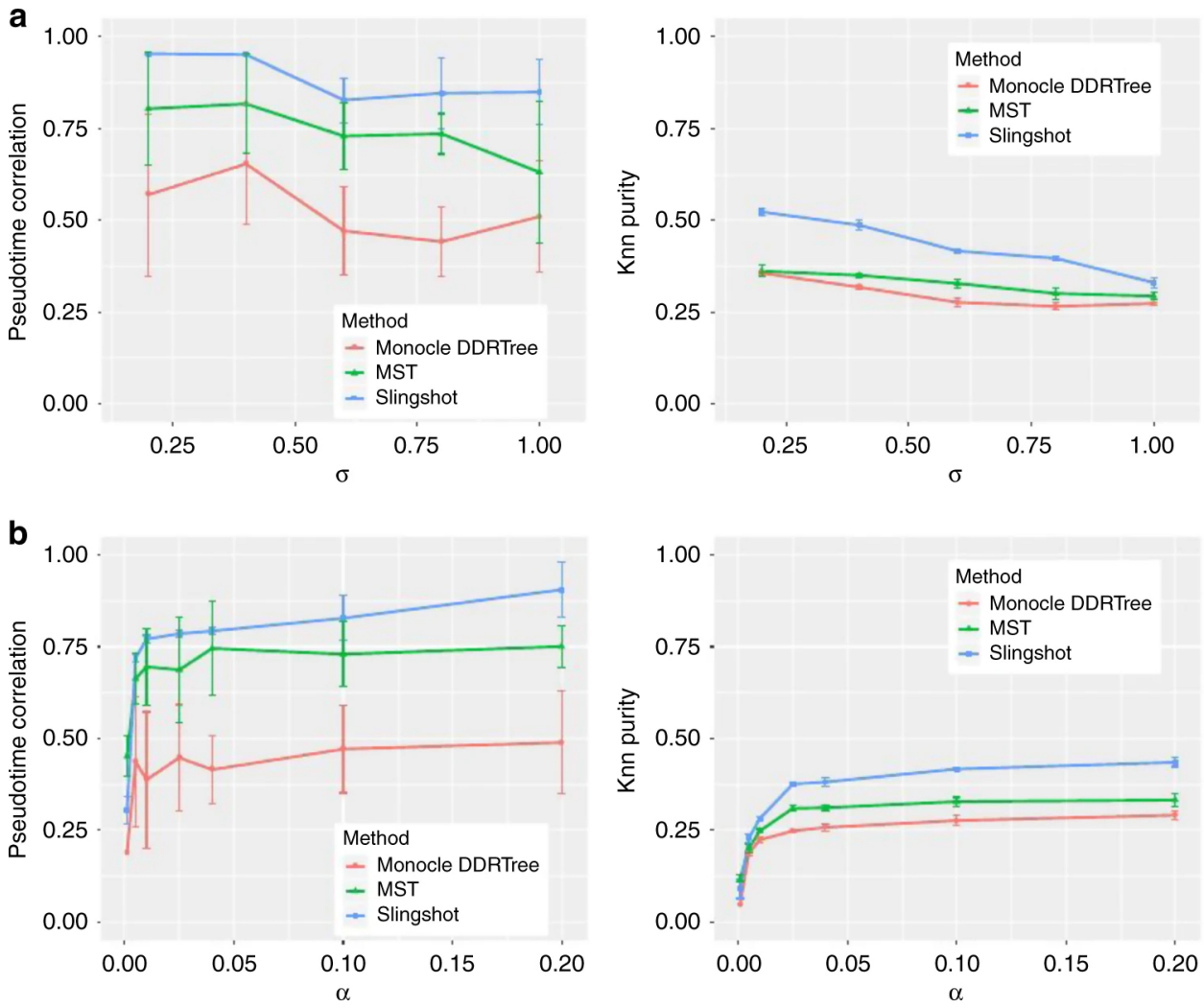
Figure 1.10: Effects of the *bimod* parameter on the performance of clustering method



In each panel we show the AUROC and Negative of Correlation of detecting DE genes using four different methods from observed counts with changing capture efficiency α ($\sigma = 0.6$) (a) 30 cell to 30 cell comparison (b) 300 cell to 300 cell comparison (c) Percentage of change of total transcript for each gene from $bimod = 0$ to $bimod = 1$

We then perform all DE methods on the datasets with $bimod = 1$ and compare the results with the original dataset with $bimod = 0$, used in Section "Using SymSim to evaluate differential expression methods". From Supplementary **Figure1.10a-c** we can see a drop in performance when increasing *bimod*, especially when the number of cells is small (**Figure1.10ab**). The drop is less prominent when the number of cells in the two populations are respectively 300 and 300. Notably, the drop in the performance of clustering and DE performance cannot be simply attributed to a global decrease in gene expression levels, since increasing *bimod* does not change this statistic (**Figure1.10(d)**).

Figure 1.11: Benchmark trajectory inference methods



Benchmark trajectory inference methods

(*a*)Pseudotime correlation and knn purity of all methods when varying σ ($\alpha = 0.1$). (*b*) Pseudotime correlation and knn purity of all methods when varying α ($\sigma = 0.6$)

Using SymSim to evaluate trajectory inference methods

The ability of SymSim to generate a continuum of cell states makes it a convenient choice to benchmark trajectory inference methods. We compare three methods including Monocle [55, 132], Slingshot [59], and a minimum spanning tree (MST) algorithm implemented in the package dynverse [56] (Methods). We generate datasets with different values of σ and α with the input tree shown in **Figure1.4(c)**. For each parameter configuration, we repeat the simulation 10 times. To evaluate the trajectory inference methods, we use two measures: (1) Spearman correlation between true cell order and inferred cell order. We consider cells on each lineage (a path from root to a leaf) separately and take the average of correlation on all five lineages. (2) k-nearest neighbor purity (kNN purity) of cells, that is, for each cell, we calculate the Jaccard Index between its k-nearest neighbors in the true trajectory and that in the inferred trajectory. Results are shown in **Figure1.11**. In these plots, k is set to 100. In **Figure1.11(a)**, we vary σ and fix α as 0.1. Both the correlation and kNN purity decrease when σ increases. In **Figure1.11(b)**, we vary α and fix σ as 0.6. All methods show an overall increasing trend along with α with both measures. Consistent with a recent benchmark study [56], we observe that overall Slingshot clearly outperforms the other two methods.

Experimental Design

Deciding how many cells to sequence is a decision many researchers face when designing an experiment, and the optimal number of cells to sequence depends on both the biological system and the goal of the experiment. A previous approach to this problem [133] only considered the aspect of counting cells (namely, having enough cells in the pool from each subpopulation), but did not account for the identifiability of each subpopulation, which may be hampered by both technical and biological factors as well as the performance of clustering algorithms.

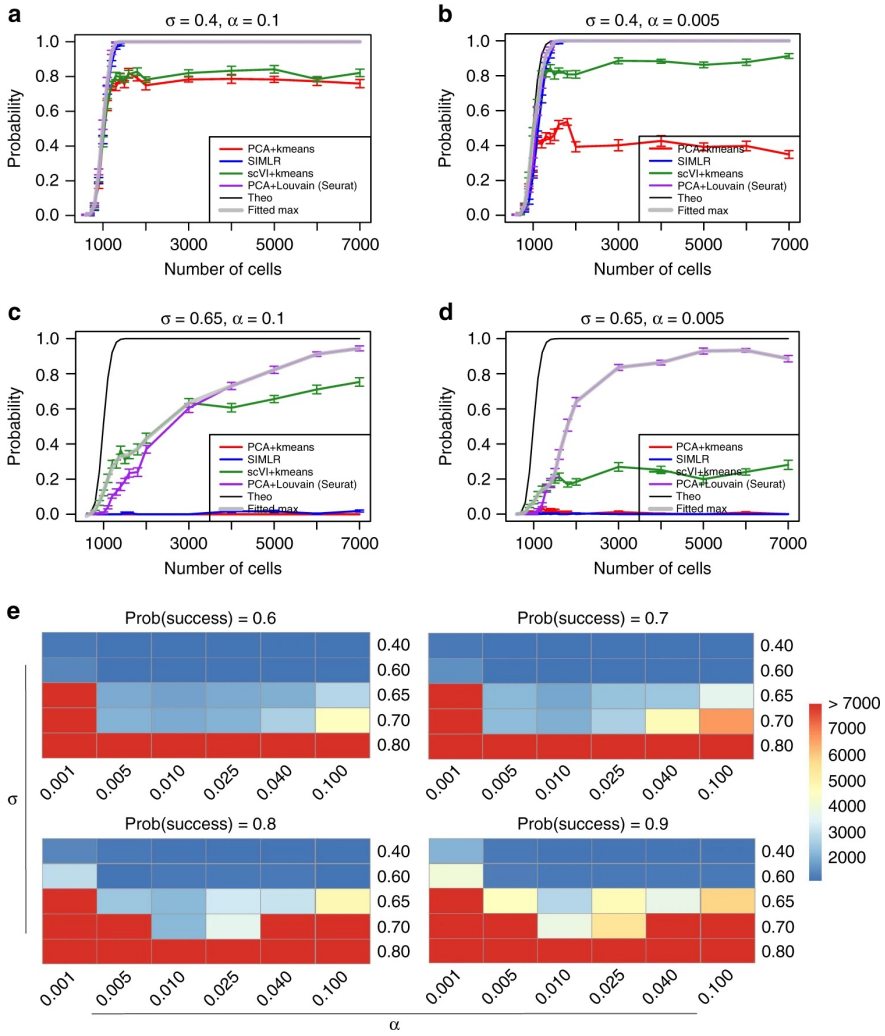
In the following we demonstrate how SymSim can be used to shed more light on this important problem. Importantly, in its current form SymSim does not use real data to model between-population variability. We therefore interpret the results in a relative manner - how do different variability factors shift the required number of cells, compared to each other and to the theoretical lower bound (i.e., sampling a prespecified minimal number of cells, Methods). Our example focuses on a case of one rare subset, represented by cells from population 2 (using the same tree in **Figure1.8(c)**; note that one can easily generalize this procedure to multiple rare subpopulations). We simulate observed counts with numbers of cells (N) ranging from 600 to 7000. These simulations were based on the parameters fit to the cortex dataset [111] with varying levels of σ and α (100 simulations per parameter configuration).

We applied the same three clustering methods as described in the previous section (k-

means with scVI or PCA and SIMLR). We say that a given algorithm was successful in "detecting the rare population" if at least 50 cells from this set are assigned to the same cluster, and form at least 70% of the cells in that cluster. We use these labels to compute an empirical success probability P for each algorithm and each parameter configuration. To get an upper bound on performance that better reflects the data (rather than the algorithm), we take the maximum P at each configuration, and apply cubic spline smoothing (gray curves, Figure 7a-d). In each plot we also include the theoretical limit which only requires the presence of at least 50 cells from the rare subpopulation (Methods). The theoretical curve (which is independent of all parameters except N) reaches almost 1 at N=1400. Conversely, the empirical curves vary dramatically, based on parameter values. For an easy case of low within- population variability (σ=0.6) and high capture efficiency (α=0.1) the empirical upper bound curve is close to the theoretical one (**Figure1.11(c)**). This curve clearly decreases when increasing the effect of either nuisance factor (**Figure1.11bc**). The reduction is substantially more dramatic when both nuisance factors increase (**Figure1.11(d)**).

To understand the implications on the number of cells required in a given setting, we calculated how many cells are required, in each configuration, to achieve a success rate of 0.75 (**Figure1.11(e)**). As expected, the resulting numbers can be much higher than the theoretical lower bound. For example, even when we have good capture efficiency ($\alpha = 0.1$), when the within- population variability increases ($\sigma = 0.8$), we need 6225 cells, while with the theoretical curve, we need only less than 1100 cells (for $P = 0.75$). Considering only the binomial sampling of cells may therefore underestimate the number of cells needed for a realistic scenario, and considerations of biological and technical variations with simulators like SymSim is merited.

Figure 1.12: The number of cells needed to detect a rare population.



The number of cells needed to detect a rare population.

We generate five populations according to the tree structure shown in Figure 1.4 and set population 2 as the rare population which accounts for 5% of the cells. Other populations share 95% of the cells evenly. The criteria of detecting the rare population are that at least 50 cells from this population are correctly detected and the precision (positive predicted value) is at least 70%. **a-d**The probability of detecting the rare population when sequencing N (x-axis) cells under different σ and α configurations, with different clustering methods. The black curve represents the theoretical probability from the binomial model, assuming that all cells sequenced are assigned correctly to the original population. The gray curve with transparency takes the maximum value at each data point from all four clustering methods with smoothing. Error bars are standard deviation over 20 randomizations. (*b*) The heatmaps show the number of cells needed to sequence under different configurations of σ and α to detect the rare population with success rates $0.6, 0.7, 0.8, 0.9$, always using the best clustering method.

1.3 Methods

Simulating gene expression with the kinetic model

As shown in **Figure1.1(a)**, the kinetic model of gene expression considers that a gene can be either on or off and the probabilities to transit between the two states are k_{on} and k_{off}. When the gene is on it is transcribed with transcription rate s. The transcripts degrade with rate d. For a given gene, based on these parameters one can simulate the number of its transcript molecules over time. The theoretical probability distribution can be calculated via the Master Equation, which is the steady state solution for the kinetic model. Therefore, the gene expression values for a gene can be sampled from the Master Equation. Alternatively, the kinetic model can be represented by a Beta-Poisson model, which we use in our implementation. Calculating parameters for the kinetic model in SymSim simulation For a gene in a cell, the parameters for the kinetic model k_{on}, k_{off}, and s are calculated from the EVF vectors of this cell and the gene effect vectors for the gene (**Figure1.1(a)**). To allow independent control of the three parameters, we use one EVF vector and one gene effect vector for each parameter. Take kon as an example: denoting the EVF vector as $(e_1^{kon}, e_2^{kon}, ..., e_p^{kon})$,and the gene effect vector fork_{on} as $(g_1^{kon}, g_2^{kon}, ..., g_p^{kon})$, the cell-gene specific value for k_{on} is the dot product of these two vectors. Then we map these kon values to the distribution of kinetic parameters estimated from experimental data, to obtain the matched parameters. We sort the k_{on} values for all genes in all cells, sample the same number of values from the experimental k_{on} distribution (the number of values would be $m * n$, where m is the number of genes and n is the number of cells), and update the k_{on} values to the ones sampled from the experimental distribution with the same rank. The parameters of k_{off} and s are calculated in the same way.

Estimating kinetic parameters from real data

We estimated kinetic parameters from experimental data using an MCMC approach. For each gene, its expression X depends on p, the proportion of time it is on, and the mRNA synthesis rate s.The parameter p itself is also a random variable determined by the kinetic parameters k_{on} and k_{off}. We model p as a Beta distributed variable with shape parameters k_{on} and k_{off}. We model X as a Poisson distributed variable with parameter $p * s$. The distribution of X is then identical to the distribution calculated using the Master Equation [134]. The downsampling effect is modeled as a Binomial sampling with X being the number of trials, and f being the probability that a transcript is sampled for sequencing.

We fit this model to the experimental data using the Gibbs sampler implemented in RJAGS. At every iteration, we sample each parameter from its marginal posterior conditional on the value of all other parameters. To meet the assumption that all cells share the same kinetic parameters we divide cells by clustering that is performed in the original

study and fit the model to counts in a single cluster of cells at a time. We also use imputed read counts, rather than the raw read counts since the technical biases in single cell RNA sequencing is not a simple binomial sampling process. We use scVI [43] for the imputation. Since MCMC is dependent on initial conditions, we fit the model independently for three times, for each cell cluster and each imputation method. We thinned the MCMC chain to reduce the effect of autocorrelation, and combined all results to obtain the final distribution of kinetic parameters to use for the simulation. Ranges of kinetic parameters from literature

We look into literature for the ranges of kinetic parameters k_{on}, k_{off}, and s which are experimentally measured [113, 114, 115, 116, 117, 118, 119, 120]. The range of burst size, or s, from these studies ranges from 2-4000. And the k_{on} and k_{off} values ranges from 0.0001 to 1 per minute, and the half life of mRNA varies from 1-10 hours, which correspond to 0.001 to 0.01 per minute. This means that k_{on}/d and k_{off}/d could take values from 10-2 to 103.

Simulation of technical steps from mRNA capturing to sequencing

We simulate two categories of library preparation protocols, one does not use UMIs (unique molecular identifiers) [23] and sequences full length mRNAs (using procedures in Smart-seq2 [110] as template), and the other uses UMIs and sequences only the 3' end of the mRNA (using the Chromium chemistry by 10x Genomics as template). In the pre-amplification step, we provide option of using linear amplification to mimic the CEL-seq protocol. As shown in **Figure1.5(a)**, we take one transcript with 16 molecules as an example. To implement the UMIs, each original molecule has a variable to its count at each step. The technical steps include the following:

1) Capturing step: molecules are captured from the cell with probability α.

2) Pre-amplification step: if using non-CEL-Seq protocols, this step involves N rounds of PCR amplifications. We introduce sequence-specific biases during amplification, which includes transcript length bias and other bias assigned randomly. Parameter lenslope can be used to control the amount of length bias, and $MaxAmpBias$ is used to tune the total amount of amplification bias. The transcript lengths can be sampled from a pool of lengths from human genome or mouse genome. If using CEL-Seq protocol this step is the in vitro transcription (IVT) linear amplification.

3) Fragmentation step: the mRNAs are chopped into fragments for sequencing. If sequencing full-length mRNA, all fragments with acceptable length are kept for sequencing. If sequencing only the 3' end for UMI protocols, only fragments on the 3' end are kept for sequencing. For each transcript length, we calculate a distribution of number of fragments given expected fragment length, and use this distribution to generate the number of fragments during our simulation of the fragmentation step. When simulating the fragmentation step, we need the number of fragments obtained from a transcript. This number is dependent on the transcript length (denoted by L), the read length (r), maximum fragment length (f) and expected gap size (g) of the reads assuming we use paired-end sequencing. The fragmentation efficiency

which is the probability with which a cut happens to a position on the transcript is:

$$e = 1/(2 * r + g)$$

.
For nonUMI protocols where full length mRNA is sequenced, for each transcript length, we simulate the fragmentation process many times with the probability e and remove resulting pieces which have length smaller than r or greater than f, and we obtain a distribution of number of valid fragments for a given transcript length. In SymSim, we just sample from this distribution. For UMI protocols, we only need the valid fragments at the 3' end. In this case, we can derive theoretical distributions of the probability that a mRNA copy gives rise to a fragment. The expressions are as follows:

$$\begin{cases} (1-e)^r(1-(1-e)^{(f-r-1)}), L \geq f \\ (1-e)^r, r < L < f \\ 0, L \leq r \end{cases}$$

So during SymSim we sample with these probabilities to get the number of 3' end fragments (which will be either 0 or 1). In our paper, we set $r = 100, f = 1000, g = 200$. Key parameters which give rise to the length bias patterns shown in **Figure1.5(b)** are: $\alpha = 0.05, lenslope = 0.023, nbins = 20, MaxAmpBias = 0.3, Depth = 1.3e6$.
4) Amplification step: fragments go through another k rounds of PCR amplifications for all protocols, including CEL-Seq and non-CEL-Seq protocols. 5) Sequencing step: amplified fragments from the previous step are randomly selected according to a given value of sequencing depth, which is the total number of reads (fragments) to sequence. This parameter is denoted by *Depth*. 6) After the sequencing step (assuming all reads are correctly sequenced and mapped to their original gene), we can get the UMI counts for UMI protocols and read counts for non-UMI protocols. We omit steps like reverse transcription, library cleaning up as the effect of these steps on the final read or UMI counts are relatively minor. Simulation of amplification biases and how UMI corrects this We divide amplification biases into two categories: the biases caused by transcript length (referred to as gene length bias) and biases caused by other factors including GC contents. For the former we have observed clear pattern from experimental data with the nonUMI simulations (**Figure1.5(b)**) so we use a linear model to simulate it. For the latter there is no clear pattern observed or reported so we use a random Guassian term to represent it. We first bin all gene lengths into *nbins* bins, and get the average value in each bin: $L = (l_{\text{bin}(1)}, l_{\text{bin}(2)}, ..., l_{\text{bin}(\text{nbins})})$. We use parameter lenslope to control the amount of gene length bias, then the linear function for gene length bias is

$$B_{\text{length}}(i) = lenslope \times \text{median}(L) - lenslope \times L(i)$$

. Denoting the total amount of amplification bias by $MaxAmpBias$, then the maximum amount of random bias $2 \times MaxAmpBias/(\text{nbins} - 1)$. The random bias term is generated

by $N(0, MaxRandBias)$ and rounded to $[-MaxRandBias, MaxRandBias]$.
For a given gene of length l, its PCR amplification rate is

$$\text{rate\_2PCR} + B_{\text{length}}(\text{bin}(l)) + B_{\text{rand}}$$

. This rate is used in all rounds of the pre-amplification step. The biases then get amplified as more PCR cycles are performed, where transcripts with higher amplification rate will likely get more molecules. Assigning a UMI to each molecule before amplification allows us to collapse all molecules with the same UMI after amplification, so different amplification rates will not affect the final molecule counts.

Fitting simulation parameters to real data

To find the best matching parameters to a real dataset, we simulate a database of datasets with a grid of parameters over a wide range. For each simulated dataset, we calculate the following statistics: mean, percent non-zero, standard deviation of genes over all cells. Then given a real dataset, we find the simulated dataset, which have the most similar distributions of the statistics to the real data, and return the corresponding parameter configurations.

Applying dimensionality reduction and clustering methods

We apply three different dimensionality reduction methods to cluster cells simulated from multiple discrete populations: PCA, scVI, and SIMLR. PCA is the naive baseline method that is also the most commonly seen in single-cell RNA-seq analysis. scVI is a more recent method that uses a zero-inflated negative binomial variational auto-encoder model to infer latent space for each single cell. For both the first two methods, cluster identities are then assigned using k-means clustering. The third method, SIMLR, performs dimensionality reduction and cluster identity iteratively to maximize cluster separation. The fourth method, implemented in Seurat, uses PCA for dimensionality reduction and the Louvain clustering.

Simulation of differentially expressed genes

Diff-EVFs give rise to differences between populations as well as DE genes between populations. DE genes by design are the ones with non-zero gene effect values corresponding to the Diff-EVFs (**Figure1.8(a)**), as the gene effect vectors are sparse with a majority of values being 0s. Nevertheless, in some cases, the actual expression values of genes with at least one Diff-EVFs might not differ since the effects of different Diff-EVFs or the effects of modifying different kinetic parameters may cancel out. Differential expression might also be blurred by a high within-population variability. Thus we also use the log2 fold change (LFC) of mean gene-expression from the two populations as another criteria. The mean expression can be calculated based on simulated true counts, which is subject to gene-expression intrinsic noise, or based on the kinetic parameters themselves, directly from the theoretical gene-expression

distribution. If the kinetic parameters of a gene in a cell is k_{on}, k_{off}, and s, the expected gene-expression of this gene in this cell is $s * k_{on}/(k_{on} + k_{off})$. We use multiple thresholds ranging from 0.6 to 1 on the |LFC| to define a gene is DE, in order to avoid being biased with one single artificial threshold.

Detection of differentially expressed genes

DE genes in observed counts are detected, respectively, with edgeR, DESeq2, Wilcoxon test, and Student t-test. For edgeR, we used the quasi-likelihood approach (QLF) with cellular detection rate (the fraction of genes that are detected with non-zero counts in each cell) as covariate. For DESeq2, we use local for the fittype parameter, and we evaluate its performance, respectively, based on the output p-values and adjusted p-values, which serve as filtering of genes.

The output from each DE method is a p-value for each gene, with smaller values meaning the gene is more likely to be a DE gene. We use two metrics to evaluate the performance of a DE method: (a) AUROC (area under receiver operating characteristic curve), where we apply different thresholds on the p-values to obtain different sets of predicted DE genes, and we can then plot ROC curves with different combinations of $specificity$ and $sensitivity$, thus calculate the area under the ROC curve. (b) Negative of Spearman correlation between the p-values of each detection method and the log fold difference of the true expression levels. Genes with high log fold change in true transcript counts should correspond to low p-value if the DE method works well. As the inferred p-values and log fold change in true counts are expected to be anti-correlated, we take the negative of this correlation, such that higher value corresponds to better performance.

Applying trajectory inference methods

We use the R packages dynwrap (https://github.com/dynverse/dynwrap, version 0.1.0) and dynmethods (https://github.com/dynverse/dynmethods, version 0.1.0) to run the three trajectory inference methods compared in this manuscript: Monocle (version 2.6.4), Slingshot (version 0.99.12), and MST (a basic method implemented in dynmethods). All methods were run with default parameters. Both dynwrap and dynmethods are under the collection of R packages dynverse used in the manuscript by Saelens et al [56].

Effect of parameter bimod on gene-expression levels

To investigate if increasing bimod will cause decrease in overall gene-expression levels of genes thus lead to the decrease in performance of both clustering and DE methods, we calculate the percentage change of total number of transcripts of genes from $bimod = 0$ to $bimod = 1$.

That is, for each gene, we calculate

$$\left(\sum_{j=1}^{m} x_j - \sum_{j=1}^{m} x\prime_j\right) / \sum_{j=1}^{m} x_j$$

where x_j is the number of transcripts of this gene in cell j when $bimod = 0$, x'_j is the number of transcripts of this gene in cell j when $bimod = 1$, and m is the number of cells. From **Figure1.10** we see that there is no consistent increase or decrease of total number of transcripts for the genes when changing *bimod*. Therefore, we conjecture that the drop in the performance of clustering and DE is rather caused by change in the distribution of gene-expression levels of genes instead of overall gene-expression levels.

Calculating the probability of detecting a population

Assuming all sequenced cells are correctly assigned to its original population, the probability that at least x cells are detected from a population only depends on the binomial sampling. Denote the total number of cells by N and the proportion of the cells in the given population by r, the probability that at least x cells are detected for the population is:

$$1 - \sum_{k=0}^{x-1} \left(\begin{array}{c} N \\ k \end{array} \right) r^k (1-r)^{(N-k)}$$

This formula is used to generate the black curves in **Figure 1.12**.

During our simulation to estimate the number of cells needed to detect a rare population, we simulate the random sampling process as follows: we start with a total of 10,000 cells for all five populations with 2000 cells for each population. We set probability vector of a cell belonging to each population as (0.25, 0.05, 0.25, 0.25, and 0.2), where Population 2 is the rare population with smallest probability. For each randomization and given total number of cells N ($N \leq 7000$), we randomly sample N cells from the pool of 10,000 cells according to the probability vector.

1.4 Conclusion

SymSim has the following features which are advantageous over existing simulators: (i) We simulate true transcript counts from a kinetic model that can be interpreted in terms of transcript synthesis rate, promoter activation and deactivation. (ii) When generating multiple discrete or continuous populations, instead of generating biological differences through directly altering the true transcript count distribution, we set Diff-EVFs, which can be interpreted as biological conditions which cause the differences between subpopulations of cells. This is a more natural and realistic way to simulate biological transcriptional differences. It generates desirable properties such as a larger number of differentially expressed genes

between more distantly related cell populations without manual specifications. (iii) When generating observed counts, we simulate key steps in real experimental protocols, which automatically gives us dropout events, length bias, and distribution of library sizes. We also provide choices to use UMI based protocols or non-UMI full length mRNA protocols, as the properties of data output from these two categories can be very different.
The main input parameters to SymSim are self-explanatory with their own biological or technical meanings, which facilitates decision of input parameters in practice. In particular, we allow users to input an experimental dataset and SymSim can return the parameter settings which best match the properties of the input dataset. The modular nature of SymSim provides possibilities to generalize its application. For example, the generation of true counts with EVFs and transcription kinetics can be replaced by learning a generative model from real data, with methods such as scVI [43]. This kind of extension will facilitate data- driven simulation between-subpopulation variability, albeit at the cost of using parameters that are less interpretable biologically.

The feature of finding best parameter configurations to match an experimental dataset not only yields large-scale simulated datasets, but also brings insights on the properties of experimental dataset through the parameters found, as the parameters are biologically or technically interpretable. For both the UMI and nonUMI datasets, we include the top parameter settings for both data sets in Supplementary Material Section 4. From these parameters, one can conjecture that the cortex data set is of more heterogeneous to the Th17 dataset (higher σ), and the capture efficiency of the UMI data is much lower. However, in this paper, we do not go into comparing these parameters to the biological truth, but rather focus on obtaining simulated datasets with similar statistical properties to the input experimental data.

1.5 Acknowledgement

This chapter is written in collaboration with Xiuwei Zhang and is supervised by Nir Yosef and was previously published in Nature Communication. X.Z. was supported by grant #220558 from the Ragon Institute of MGH, MIT and Harvard. C.X. and N.Y. were supported by NIH/NHLBI grant U19 AI-090023–09.

Chapter 2

Probabilistic Harmonization and Annotation of Single-cell Transcriptomics Data with Deep Generative Models

2.1 Introduction

Recent technological improvements in microfluidics and low volume sample handling [135] have enabled the emergence of single-cell transcriptomics [136, 45] as a popular tool for analyzing biological systems [137, 138, 139]. This growing popularity along with a continued increase in the scale of the respective assays [140] has resulted in massive amounts of publicly available data and motivated large scale community efforts such as the Human Cell Atlas [10], Tabula Muris [141] and the BRAIN Initiative Cell Census Network [142]. The next natural step in the evolution of this field is therefore to integrate many available datasets from related tissues or disease models in order to increase statistical robustness [143], achieve consistency and reproducibility among studies [144, 33], and ultimately converge to a common ontology of cell states and types [10, 145].

A fundamental step toward the ideal of a common ontology is data *harmonization*, namely integration of two or more transcriptomics datasets into a single dataset on which any downstream analysis can be applied. We use the term harmonization rather than *batch effect correction* in order to emphasize that the input datasets may come from very different sources (*e.g.*, technology, laboratory), and from samples with a different composition of cell types. A wide range of methods have already been developed for this fundamental problem, initially for microarrays and later on for bulk RNA sequencing, such as ComBat [146] and limma [147]. These approaches mainly rely on generalized linear models, with empirical Bayes shrinkage to avoid over-correction. More recently, similar methods have been pro-

posed specifically for single-cell RNA sequencing (scRNA-seq), such as ZINB-WaVE [148], which explicitly accounts for the overabundance of zero entries in the data. However, because of their linear assumptions, these approaches may not be appropriate when provided with a heterogeneous sample that includes different cell states, each of which may be associated with a different sample-to-sample bias [144]. With these limitations in mind, the next generation of methods turned to non-linear strategies. Broadly speaking, each of these methods includes a combination of two components: (i) joint factorization of the input matrices (each corresponding to a different dataset) to learn a joint low-dimensional latent representation. This is usually done with well established numerical methods, such as integrative non-negative matrix factorization (LIGER [149]), singular value decomposition (Scanorama [150]), or canonical correlation analysis (Seurat Alignment [33]); (ii) additional non-linear transformation of the resulting latent representations so as to optimally "align" them onto each other. This is usually done using heuristics, such as alignment of mutual nearest neighbors (MNN [144], Scanorama [150] and Seurat Anchors [34]), dynamic time warping (Seurat Alignment [33]) or quantile normalization (LIGER [149]). While this family of methods has been shown to effectively overlay different datasets, it suffers from two important limitations. First, an explicit alignment procedure may be difficult to tune in a principled manner and consequently result in over-normalization. This is especially relevant when the cell type composition is different between datasets and when technical differences between samples are confounded with biological differences of interest. Second, the alignment is done in an ad hoc manner and lacks probabilistic interpretability. Consequently, the resulting harmonized dataset is of limited use and cannot be directly applied for probabilistic decision-making tasks, for example differential expression.

Besides harmonization, another important and highly related problem is that of automated *annotation* of cell state. In principle, there are two ways to approach this problem. The first is *ab initio* labeling of cells based on marker genes or gene signatures [33, 53, 151]. While this approach is intuitive and straightforward, its performance may be affected in the plausible case where marker genes are absent due to limitations in sensitivity. The second approach is to "transfer" annotations between datasets. In the simplest scenario, we have access to one dataset where states have been annotated either *ab initio*, or using additional experimental measurements (e.g., protein expression [45, 90] or lineage tracing [152]) and another, unannotated dataset from a similar condition or tissue. The goal is to use the labeled data to derive similar annotations for the second dataset, whenever applicable. This task is often complicated by factors such as differences in technology (e.g., using Smart-Seq2 data to annotate 10x Chromium data), partial overlap in cell type composition (i.e., not all labels should be transferred and not all unannotated cells should be assigned a label), complex organization of the labels (e.g., hierarchy of cell types and sub-types [153], continuum along phenotypic or temporal gradients), partial labeling (i.e., only a subset of cells from the "annotated" dataset can be assigned a label confidently), and the need to handle multiple (more than 2) datasets in a principled and scalable manner. One way to address the annotation problem with this approach is learning a classifier [153, 65] in order to predict a

fixed stratification of cells. However, this approach might be sensitive to batch effects, which could render a classifier based on a reference dataset less generalizable to an unannotated dataset. Another, more flexible approach is to transfer annotations by first harmonizing the annotated and unannotated datasets, thus also gaining from the benefits of having a single dataset that can be subject to additional, joint, downstream analysis.

In this chapter, we propose a strategy to address several of the outstanding hurdles in both of the harmonization and annotation problems and use both the simulation data generated usin SymSim from Chapter 1 and published datasets to validate our strategy. We first demonstrate that single-cell Variational Inference (scVI) [43] a deep generative model we previously developed for probabilistic representation of scRNA-seq data — performs well in both harmonization and harmonization-based annotation, going beyond its previously demonstrated capacity to correct batch effects. We then introduce single-cell ANnotation using Variational Inference (scANVI), a new method that extends scVI and provides a principled way to address the annotation problem probabilistically while leveraging any available label information. Because scANVI is able to model cells with or without label information, it belongs to the category of semi-supervised learning algorithms. This flexible framework of semi-supervised learning can be applied to two main variants of the annotation problem. In the first scenario, we are concerned with a single dataset in which only a subset of cells can be confidently labeled (e.g., based on expression of marker genes) and annotations should then be transferred to other cells, when applicable. In the second scenario, annotated datasets are harmonized with unannotated datasets and then used to assign labels to the unannotated cells. both scVI and scANVI are used in later chapters for harmonization and cell type annotation tasks.

The inference procedure for both of the scVI and scANVI models relies on neural networks, stochastic optimization and variational inference [154, 155] and scales to large numbers of cells and datasets. Furthermore, both methods provide a complete probabilistic representation of the data, which non-linearly controls not only for sample-to-sample bias but also for other technical factors of variation such as over-dispersion, library size discrepancies and zero-inflation. As such, each method provides a single probabilistic model that underlies the harmonized gene expression values (and the cell annotations, for scANVI), and can be used for any type of downstream hypotheses testing. We demonstrate the latter point through a differential expression analysis on harmonized data. Furthermore, through a comprehensive analysis of performance in various aspects of the harmonization and annotation problems and in various scenarios, we demonstrate that scVI and scANVI compare favorably to current state-of-the-art methods.

2.2 Results

In the following we demonstrate that our framework compares favorably to state-of-the-art methods for the problems of harmonization and annotation in terms of accuracy, scalability, and adaptability to various settings. The first part of the paper focuses on the harmonization problem and covers a range of scenarios, including harmonization of datasets with varying levels of biological overlap, handling cases where the data is governed by a continuous (e.g., pseudotime) rather than discrete (cell types) form of variation, and processing multiple (> 20) datasets. While we demonstrate that scVI performs well in these scenarios, we also demonstrate that the latent space learned by scANVI provides a proper harmonized representation of the input datasets — a property necessary for guaranteeing its performance in the annotation problem.

In the second part of this manuscript we turn to the annotation problem and study its two main settings, namely transferring labels between datasets and *ab-inito* labeling. In the first setting we consider the cases of datasets with a complete or partial biological overlap and use both experimentally- and computationally- derived labels to evaluate our performance. In the second setting, we demonstrate how scANVI can be used effectively to annotate a single dataset by propagating high confidence seed labels (i.e., based on marker genes) and by leveraging a hierarchical structure of cell state annotations. Finally, we demonstrate that the generative models inferred by scANVI and scVI can be directly applied for hypotheses testing, using differential expression as a case study.

Joint modeling of scRNA-seq datasets

We consider a collection of scRNA-seq datasets (**Figure 2.1ab**).After using a standard heuristic to filter the genes and generate a common (possibly large) gene set of size G (**Method**), we obtain a concatenated dataset that may be represented as a matrix. Individual entries x_{ng} of this matrix measures the expression of gene g in cell n. Additionally, we use the integer s_n to denote the dataset of origin for each cell n. Finally, a subset of the cells may be associated with a cell state annotation c_n, which can describe either discrete cell types or hierarchical cell types. More complex structures over labels such as gradients are left as a future research direction.

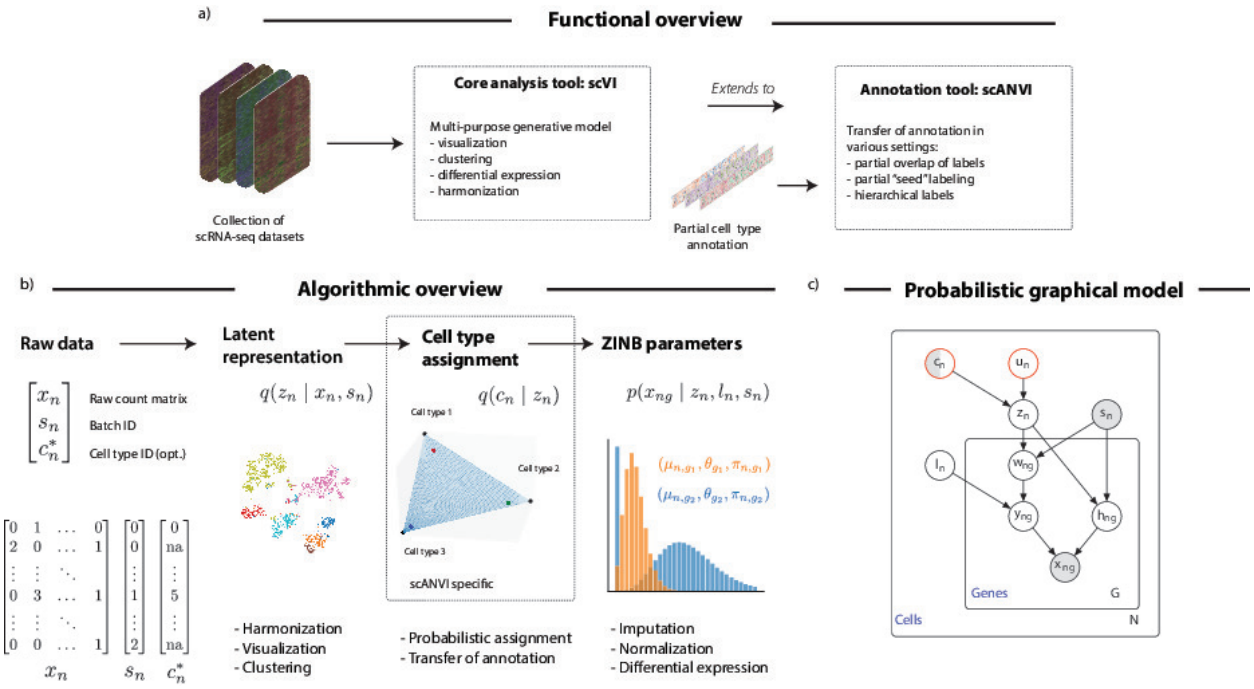


Figure 2.1: Harmonization and annotation of scRNA-seq datasets with generative models

Harmonization and annotation of scRNA-seq datasets with generative models. (a) Functional overview of the methods proposed in this manuscript. (b) Schematic diagram of the variational inference procedure in both of the scVI and scANVI models. We show the order in which random variables in the generative model are sampled and how these variables can be used to derive biological insights. (c) The graphical models of scVI and scANVI. Vertices with black edges represent variables in both scVI and scANVI, and vertices with red edges are unique to scANVI. Shaded vertices represent observed random variables. Semi-shaded vertices represent variables that can be either observed or random. Empty vertices represent latent random variables. Edges signify conditional dependency. Rectangles ("plates") represent independent replication. The complete model specification and definition of internal variables is provided in the **Method**

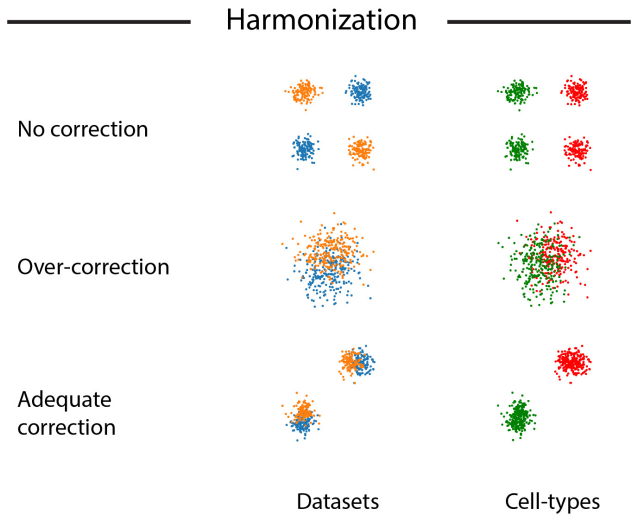


Figure 2.2: Schematic of the data harmonization problem

Schematic of the data harmonization problem
We are provided with two datasets (orange and blue), each consisting of two cell types (red and green). Our evaluation for the harmonization problem consists of two objectives: (1) mixing the two datasets well and (2) retaining the original structure in each dataset. Scenario 1 (top) is the case of under correction where objective (2) is achieved while objective (1) is not. Scenario 2 (middle) is the case of over correction where objective (1) is improved while objective (2) becomes worse. The bottom panel shows the desired scenario of mixing the datasets well while retaining the biological signal.

Since the problem of data harmonization of single-cell transcriptomics is difficult and can potentially lead to over-correction (**Figure 2.2**) [156], we propose a fully-generative method as a robust and principled approach to address it. In our previous work [43], we built single-cell Variational Inference (scVI), a deep generative model where the expression level x_{ng} is zero-inflated negative binomial (ZINB) when conditioned on the dataset identifier (s_n), and two additional latent random variables. The first, which we denote by l_n, is a one-dimensional random variable accounting for the variation in capture efficiency and sequencing depth. In practice, we noticed that this random variable is highly correlated to the library size [43]. The second, which we denote as z_n, is a low dimensional random vector that represents the remaining variability (**Figure 2.1(b)**). This vector is expected to reflect biological differences between cells, and can be effectively used for visualization, clustering, pseudotime inference and other tasks. Since the scVI model explicitly conditions on the dataset identifier (in the sense that it learns a conditional distribution, see **Method**), it provides an effective way of controlling for technical sample-to-sample variability. However, scVI is unsupervised and does not make use of the available annotations c_n, which can further guide the inference of an informative latent representation z_n. To this end, we present a more refined hierarchical structure for z_n. We draw z_n as a mixture conditioned on the cell annotation c_n and another latent variable u_n, accounting for further biological variability within a cell type (**Method**). We name the resulting approach single-cell ANnotation using Variational Inference (scANVI).

The variables z_n, inferred either with scVI or scANVI, provide an embedding of all cells in a single, joint latent space. Since this latent space is inferred while controlling for the dataset of origin (s_n), it inherently provides a way to address the harmonization problem. The annotation of unlabeled cells can therefore be conducted with scVI using their proximity to annotated cells in the joint latent space (e.g., using majority vote over the k-nearest neighbors). The scANVI model provides a more principled way to annotate cells, namely through a Bayesian semi-supervised approach. Once fitted, the model is able to provide posterior estimates for the unobserved cell state c_n, which can be particularly useful when labels cannot be entirely trusted. Because the marginal distribution $p(x_{ng}, c_n \mid s_n)$ if c_n observed (resp. $p(x_{ng} \mid s_n)$ otherwise) s not amenable to exact Bayesian computation, posterior inference is intractable. Consequently, we use variational inference parameterized by neural networks to approximate the posterior distribution [154] (**Method**).

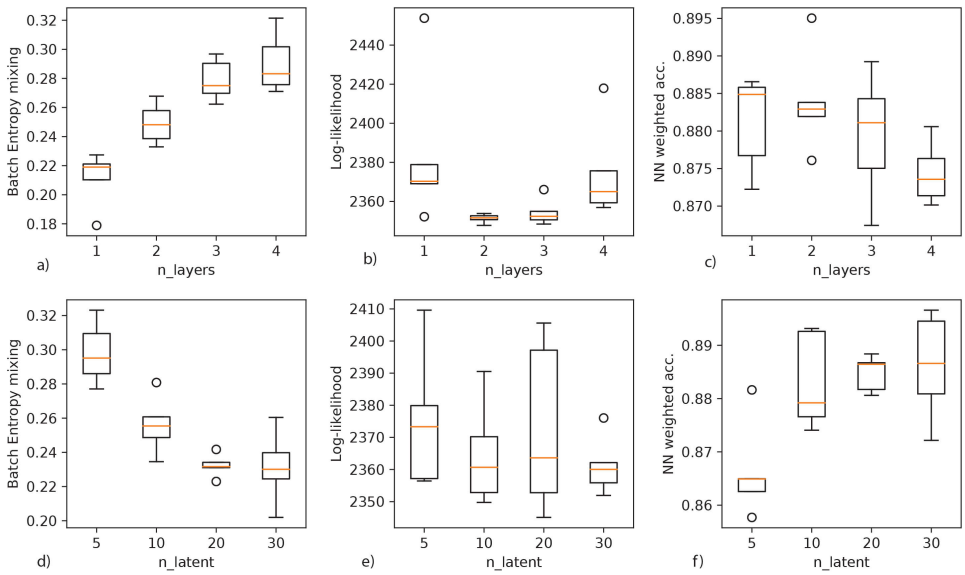


Figure 2.3: Robustness analysis for harmonization of the pair of datasets MarrowMT-10x / MarrowMT-ss2 with scVI

Robustness analysis for harmonization of the pair of datasets MarrowMT-10x / MarrowMT-ss2 with scVI. $(a-c)$ We augment the number of hidden layers in the neural network f_w and track across $n = 5$ random initializations for the batch entropy mixing (a), the held-out log likelihood (b) and the weighted accuracy of a nearest neighbor classifier on the latent space (c). $(d-f)$ We increase the number of dimensions for the latent variable z and track across $n = 5$ random initialization the batch entropy mixing (d), the held-out log likelihood (e) and the weighted accuracy of a nearest neighbor classifier on the latent space (f).

Notably, scANVI and scVI both have a certain number of hyperparameters. In the following evaluations, conducted on different datasets and different scenarios, we use the exact same set of hyperparameters in order to demonstrate that our methods can be applied with a minimal requirement of hyperparameter tuning (**Method**). We provide a robustness study for hyperparameters in the context of harmonization in **Figure 2.3**.

Datasets

We apply our method on datasets generated by a range of technologies (10x Chromium [45, 158], plate-based Smart-Seq2 [165], Fluidigm C1 [22], MARS-Seq [166], inDrop [24] and CEL-Seq2 [167]), spanning different numbers of cells (from a few thousand to over a hundred thousand cells), and originating from various tissues (mouse bone marrow, human peripheral mononuclear blood cells [PBMCs], human pancreas, human and mouse brain). Datasets are listed and referenced in **Table 2.1**.

Harmonizing pairs of datasets with a discrete population structure

We conducted a comparative study of harmonization algorithms on four different instances, each consisting of a pair of datasets. The first pair (PBMC-CITE [90], PBMC-8K [158]) represents the simplest case, in which the two datasets come from very similar biological settings (i.e., PBMCs) and are generated by the same technology (i.e., 10x) but in different labs (i.e., akin to batch correction). A second scenario is that of similar tissue but different technologies, which we expect to be more challenging as each technology comes with its own characteristics and biases [168]. For instance, some methods (10x, CEL-Seq2) profile the end of the transcript and use Unique Molecular Identifier (UMI) to mitigate inflation in counting, whereas others (e.g., most applications of Smart-Seq2) consider the full length of the transcript without controlling for this potential bias. Additionally, some protocols (e.g., Smart-Seq2) tend to have higher sensitivity and capture more genes per cell compared to others. Finally, studies using droplet based protocols tend to produce much larger numbers of cells compared to plate-based methods. We explore three such cases, including a bone marrow 10x and Smart-Seq2 pair from the Tabula Muris project (MarrowTM-10x, MarrowTM-ss2 [141]), a pancreas inDrop and CEL-Seq2 pair (Pancreas-InDrop, Pancreas-CEL-Seq2 [159]), and a dentate gyrus 10x and Fluidigm C1 pair (DentateGyrus-10x, DentateGyrus-C1 [161]).

Successful harmonization should satisfy two somewhat opposing criteria (**Figure 2.2**). On the one hand, cells from the different datasets should be well mixed; namely, the set of k-nearest neighbors (kNN) around any given cell (computed e.g., using euclidean distance in the harmonized latent space) should be balanced across the different datasets. For a fixed value of k, this property can be evaluated using the entropy of batch mixing [144], which is akin to evaluating a simple k-nearest neighbors classifier for the batch identifier (**Method**).

Briefly, the entropy of batch mixing is the average negative entropy of cell type proportion of the k-nearest-neighbors of each cell in the harmonized latent space. Higher value for this metric indicates that the harmonized latent space shows strong mixing: the neighbors of each cell is composed of cells from different batches. While this property is important, it is not sufficient, since it can be achieved by simply randomizing the data. Therefore, in our evaluation we also consider the extent to which the harmonized data retains the original structure observed with each dataset taken in isolation. Here, we expect that the set of k-nearest neighbors of any given cell in its original dataset should remain sufficiently close to that cell after harmonization. We evaluate this property using a measure we call k-nearest neighbors purity (**Method**), computed as the average percent overlap of the k-nearest-neighbors of each cell before and after harmonization. This metric takes value between 0 and 1 and higher values indicate better retainment of structure. This criteria is important, but is maximized by a trivial approach of simply concatenating the latent spaces. Of course, this will result in poor performance with respect to our first measure. Our evaluation therefore relies on both types of measures, namely mixing of data sets and retainment of the original structure. Since our results depend on the neighborhood size k, we consider a range of values - from a high resolution ($k = 10$) to a coarse ($k = 500$) view of the data.

Dataset Name	Tech.	n cells	Description	Ref.
PBMC-8K	10x	8,381	peripheral blood mononuclear cells (PBMCs) from a healthy donors; labels extracted from [157]	[158]
PBMC-CITE	10x	7,667	PBMCs obtained from CITE-seq; labels generated manually by inspection of protein marker level on Seurat clusters	[90]
PBMC-68K	10x	68,579	fresh PBMCs collected from healthy donor	[45]
PBMC-Sorted	10x	94,655	Bead-purified PBMCs collected from the same donor as PBMC68K	[45]
MarrowTM-10x	10x	4,112	Mouse bone marrow cells collected from two female mice	[141]
MarrowTM-ss2	Smart-seq2	5,351	FACS sorted cells (B cells, T cells, granulocytes and Kit (+), Sca-1 (+) and Lin (-) hematopoietic stem cells) from 3 male and 2 female mice,	[141]
Pancreas-InDrop	inDrop	8,569	Human Pancreas	[159]
Pancreas-CELSeq2	CEL-Seq2	2,449	Human Pancreas	[160]
DentateGyrus-10x	10x	5,454	Mouse Dentate Gyrus	[161]
DentateGyrus-C1	Fluid. C1	2,303	Mouse Dentate Gyrus	[161]
CORTEX	10x	160,796	Mouse Nervous System	[48]
HEMATO-Tusi	inDrop	4,016	Hematopoeitic Progenitor Mouse Cells	[162]
HEMATO-Paul	MARS-seq	2,730	Hematopoeitic Progenitor Mouse Cells	[163]
SCANORAMA	Mixture	105,476	human cells from 26 diverse scRNA-seq experiments across 9 different technologies	[150]
SN-human	10x	10,000	Subsampled brain cells from human Substantia Nigra	[149]
SN-mouse	Drop-seq	10,000	Subsampled Brain cells from mouse Substantia Nigra [164]	

Table 2.1: List of datasets used in this paper

List of dataset used in this paper. Note that for the PBMC-Sorted 11 cell types were collected according to the paper but only 10 are available from the 10x website [158].

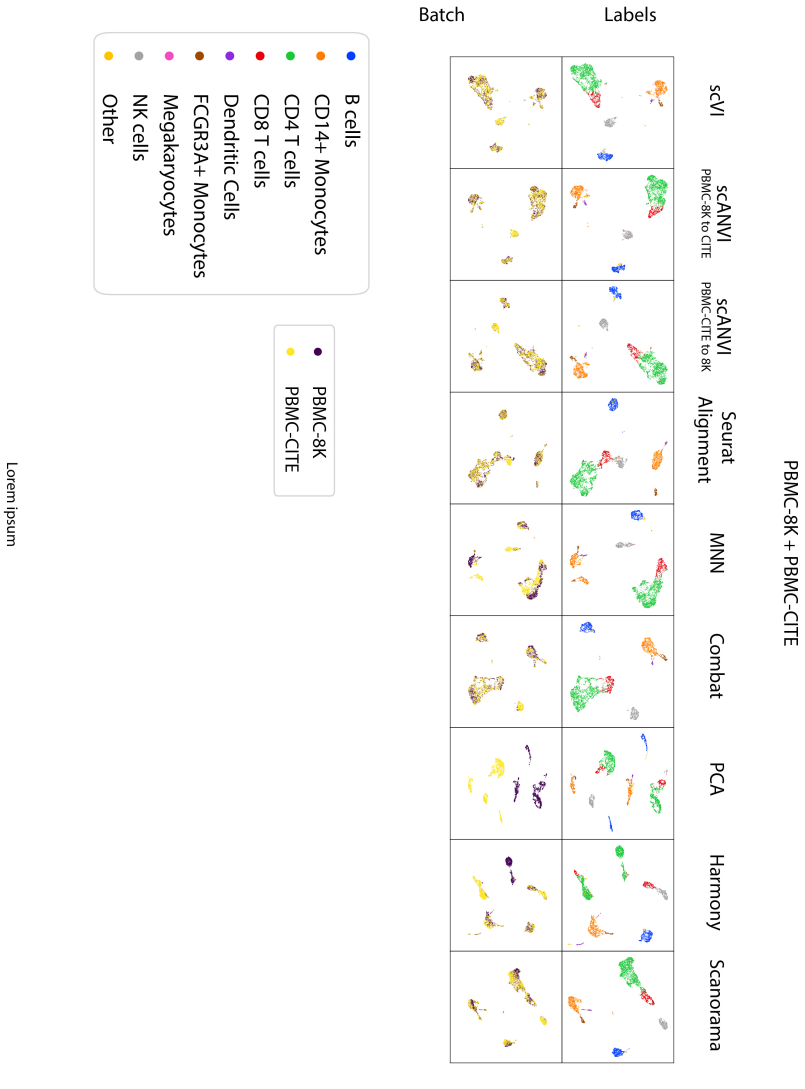


Figure 2.4: Visualization of the benchmark PBMC-8K / PBMC-CITE

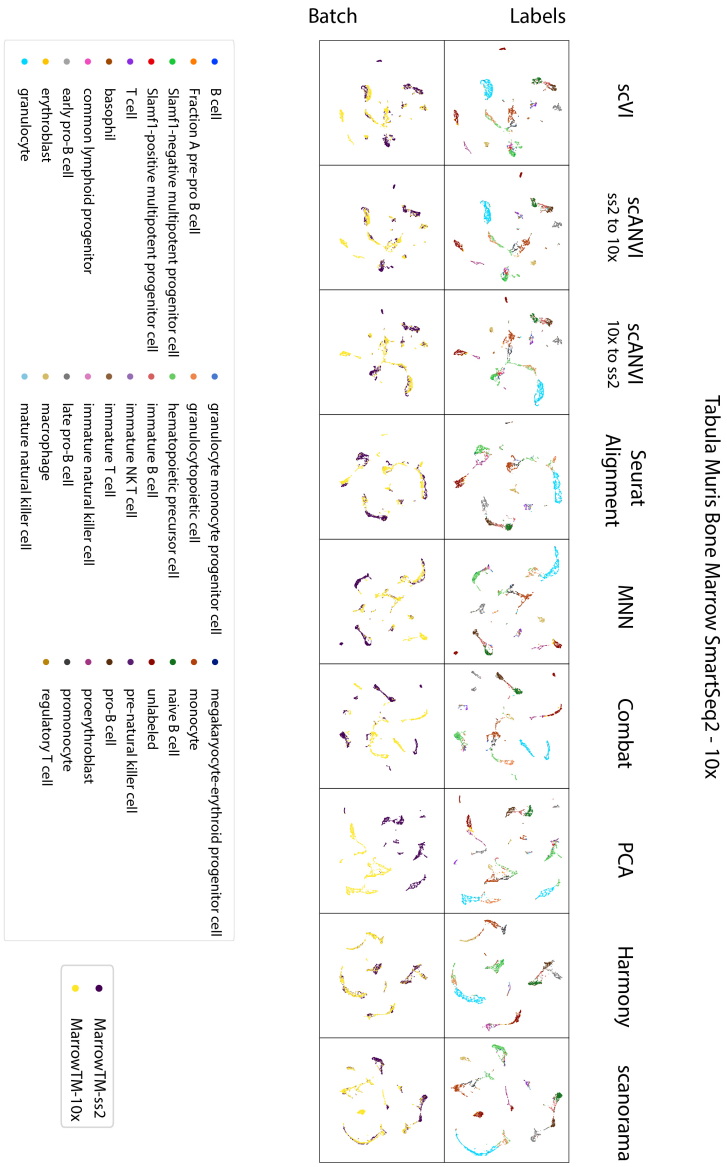


Figure 2.5: Visualization of the benchmark MarrowMT 10x / ss2

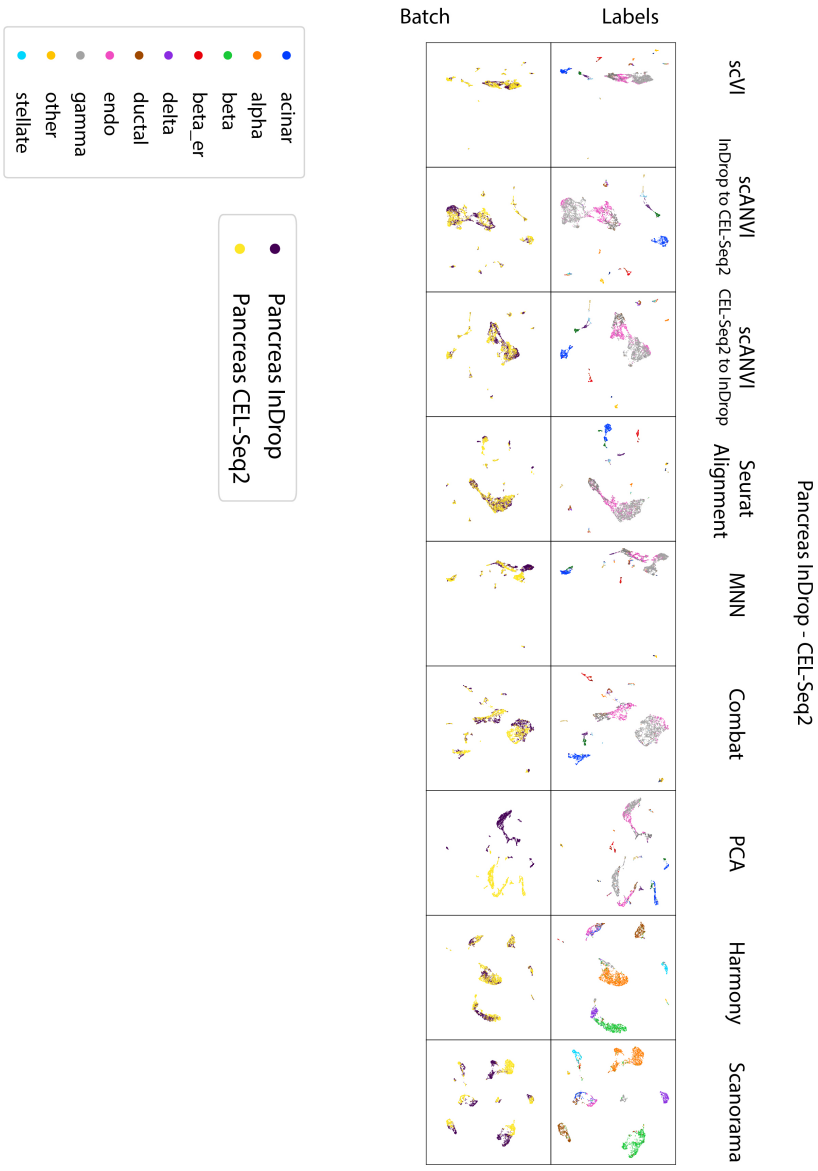


Figure 2.6: Visualization of the benchmark Pancreas InDrop / CEL-Seq2

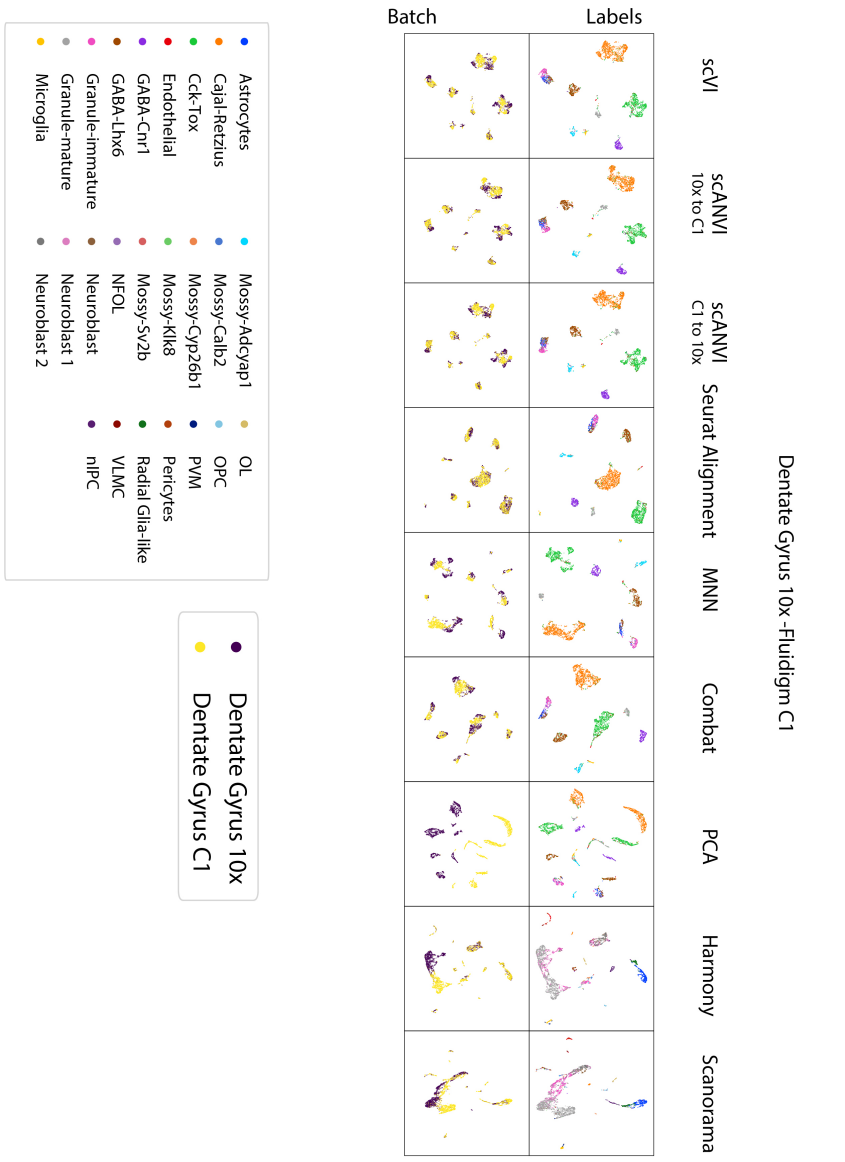


Figure 2.7: Visualization of the benchmark DentateGyrus10X - Fluidigm C1

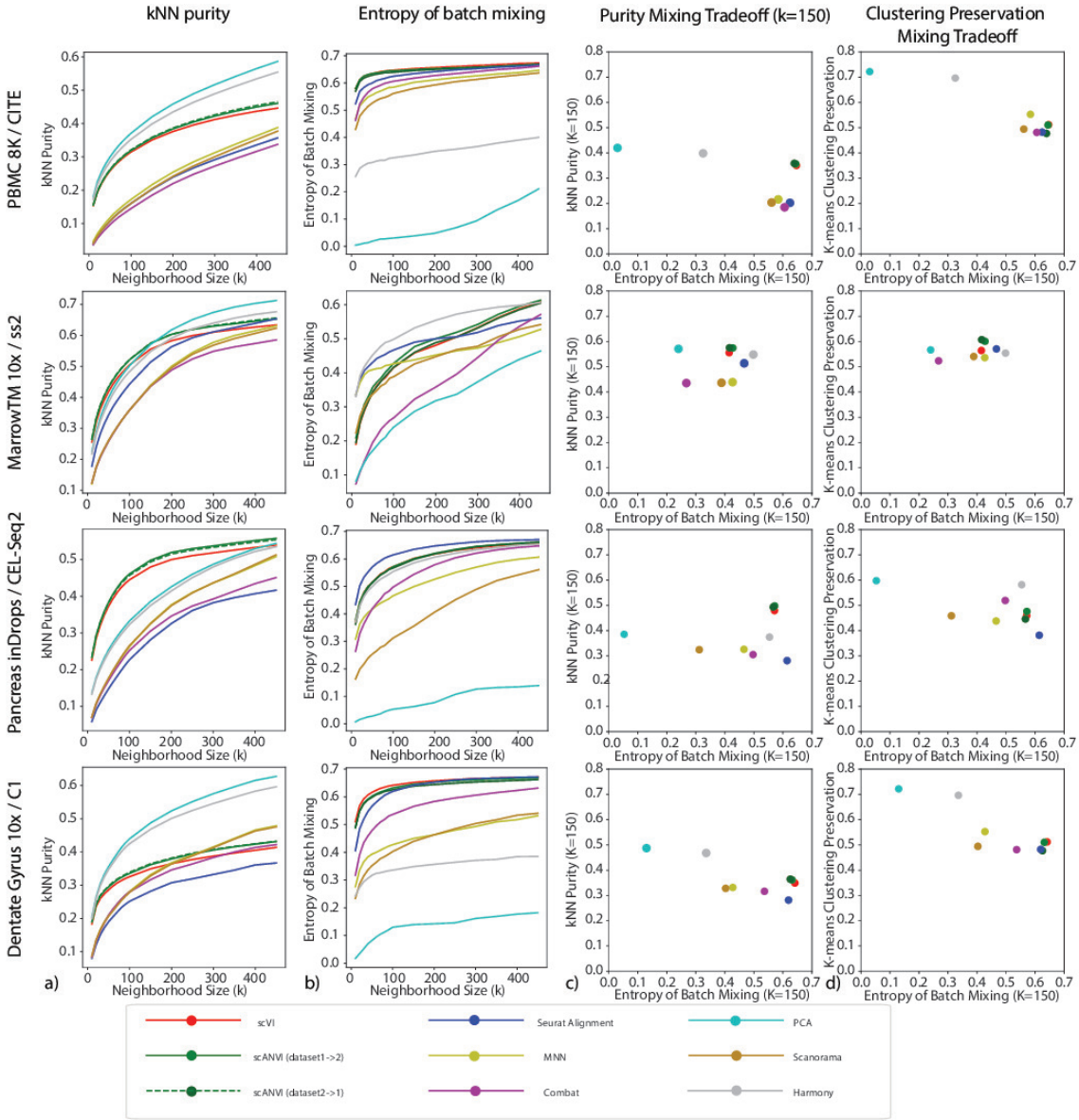


Figure 2.8: Benchmarking of scRNA-seq harmonization algorithms

Each row is a different dataset. Each column is a metric. (*a*) *k*-nearest neighbors purity that ranges from 0 to 1, with higher values meaning better preservation of neighbor structure in the individual datasets after harmonization. (*b*) Entropy of batch mixing where higher values means that the cells from different datasets are well-mixed. (*c*) The trade-off between the *k*NN purity and entropy of batch mixing for a fixed $K = 150$. Methods on the top right corner have better performances. (*d*) The trade-off between entropy of batch mixing and the preservation of biological information using an alternative unsupervised statistic k-means clustering preservation.

We compare scVI to several methods, including MNN [144], Seurat Alignment [33], ComBat [146], Harmony [169], Scanorama [150] and principal component analysis (PCA). In addition, and in order to compare our methods to unpaired data integration approaches based on generative adversarial networks [170], we also tested MAGAN [171]. However, even after manual tuning of the learning rate hyperparameter, the input datasets remain largely unmixed (data not shown). This might be due to the fact that MAGAN was not directly applied to harmonize pairs of scRNA-seq datasets and need more tuning to be applicable in that context. For each algorithm and pair of datasets, we report embeddings computed via a Uniform Manifold Approximation and Projection (UMAP) [172] (**Figure 2.4 - 2.7**) as well as quantitative evaluation metrics (**Figure 2.8**). Overall, we observed that scVI compares favorably to the other methods in terms of retainment of the original structure (**Figure 2.8(a)**) and performs well in terms of mixing (**Figure 2.8(b)**) for a wide range of neighborhood sizes and across all dataset pairs. The trade-off of these two aspects of harmonization for a fixed k is shown in **Figure 2.8(c)**, and again scVI and scANVI performs favorably and show up on the top right corner of the scatter plot. scANVI performs slightly better than scVI. Furthermore, because the conservation of k-nearest neighbors might be more indicative of a local stability of the algorithm and misses the clustering aspect of the data, we also quantified the conservation of cluster assignments. Towards this end, we used the adjusted Rand index to compare the agreement of a k-means clustering algorithm, before and after harmonization (**Figure 2.8(d)**). Reassuringly, our positive results for preservation of the output of a clustering algorithm indicates that scVI and scANVI are also stable with regards to more global aspects of the data.

While scANVI was designed for the problem of cell state annotation, we also wanted to evaluate its ability to harmonize datasets, which can be seen as a prerequisite. To evaluate this, we consider each dataset pair twice, each time using labels from one of the datasets (exploiting the semi-supervision framework of scANVI) and leaving the other one unlabeled. Reassuringly, we found that scANVI is capable of effectively harmonizing the datasets, with a similar performance to that of scVI in terms of entropy of batch mixing and retainment of the original structure (**Figure 2.8**). We further explore the performance of scANVI in the annotation problem in the subsequent sections.

Harmonizing datasets with a different composition of cell types

One of the primary challenges of the harmonization problem is handling cases in which the cell types present in the input datasets only partially overlap or do no overlap at all. Since this is a plausible scenario in many applications, it is important to account for it and avoid over-normalizing or "forcing" distinct cell populations onto each other. To evaluate this, we performed several stress tests in which we artificially manipulated the composition of cell types in the input datasets prior to harmonization. As our benchmark method we use Seurat Alignment, which performed better than the remaining benchmark methods in our

first round of evaluation (**Figure 2.8**).

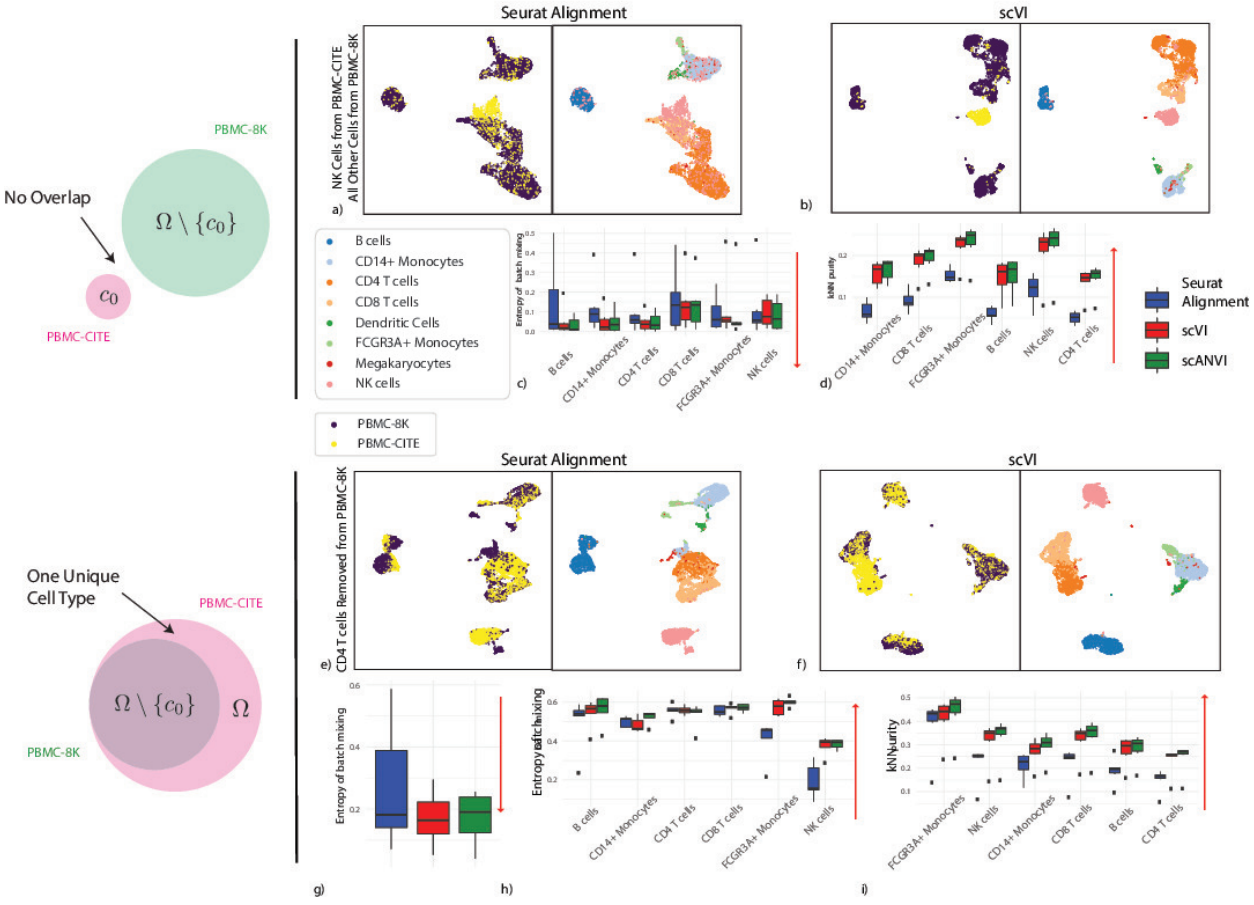


Figure 2.9: Harmonizing datasets with different cellular composition

$(a-d)$ show the case when no cell type is shared. PBMC-8K contains all cells other than cell type c_0 while PBMC-CITE contains only cell type c_0. $(a-b)$ UMAP visualization for the case where c_0 corresponds to natural killer cells. $(c-d)$ entropy of batch mixing and k-nearest neighbors purity, aggregating the six experiments (setting c_0 to a different cell type in each experiment). $(e-i)$ show the case when cell type c_0 is removed PBMC-8K but not from PBMC-CITE. $(e-f)$ UMAP visualization for the case where c_0 corresponds to CD4+ T cells. (g) entropy of batch mixing for the removed cell type. (h) entropy of batch mixing for the remaining cell types. (i) k-nearest neighbors purity, aggregating all 6 experiments. Red arrows indicate the desired direction for each performance measure (e.g., low batch entropy is desirable in (d)). The boxplots are standard Tukey boxplots where the box is delineated by the first and third quartile and the whisker lines are the first and third quartile plus minus 1.5 times the box height. The dots are outliers that fall above or below the whisker lines.

cell-type	PBMC-8K proportion	PBMC-CITE proportion
NK cells	0.036	0.178
CD8 T cells	0.119	0.091
B cells	0.133	0.104
FCGR3A+ Monocytes	0.028	0.029
CD14+ Monocytes	0.186	0.159
CD4 T	0.421	0.436
Dendritic Cells	0.026	0
Megakaryocytes	0.008	0
Other	0.043	0.004

Table 2.2: Composition of cell-types in the PBMC-8K and the PBMC-CITE dataset

As a case study, we used a pair of PBMC datasets (PBMC-CITE [90], PBMC-8K [158]) that initially contained a similar composition of immune cell types (**Table 2.2**). We were first interested in the case of no biological overlap (**Figure 2.9a-d**). To test this, for a given cell type c_0 (e.g., natural killer cells), we only keep cells of this type in the PBMC-CITE dataset and remove all cells of this type from the PBMC-8K dataset. In **Figure 2.9a-b**, we show an example of UMAP visualization of the harmonized data, with natural killer cells as the left out cell type c_0. Evidently, when harmonizing the two perturbed datasets with scVI, the natural killer cells appear as a separate cluster and are not wrongly mixed with cells of different types from the other dataset. Conversely, we see a larger extent of mixing in the latent space inferred by Seurat Alignment. A more formal evaluation is provided in **Figure 2.9c-d**, which presents our harmonization performance metrics for each cell type averaged across all perturbations (in each perturbation, c_0 is set to a different cell type). We also included scANVI with the true number of cell types ($C = 6$) in this analysis, using the cell labels from the PBMC-CITE dataset.

Under the ideal scenario of a successful harmonization, we expect both a low entropy of batch mixing (since the datasets do not overlap), and retainment of the original structure. Evidently, both scVI and scANVI exhibit a consistently low level of batch mixing that is better or comparable to that of Seurat Alignment, while retaining the original structure more accurately.

As an additional scenario, we investigated the case where the input datasets contain a

similar set of cell types, with the exception of one cell type that appears in only one of the datasets. To simulate this, for a given cell type c_0, we removed cells of this type from the PBMC-8K dataset, and then harmonize the remaining cells with the unaltered PBMC-CITE (which still contains c_0). We show an example of UMAP visualization in **Figure 2.9e-f**, removing CD4+ T cells from the PBMC-8K dataset. Evidently, in the scVI latent space, the PBMC-CITE "unique" CD4+ T cell population is not wrongly mixed with cells from the perturbed PBMC-8K dataset, but rather appears as a distinct cluster. For a more formal analysis, **Figure 2.9g-i** shows the harmonization statistics for perturbing the six major cell types present in the PBMC datasets. As above, we also evaluated scANVI in this context, using the labels from the unperturbed (PBMC-CITE) dataset.

Figure 2.9(g) shows that the entropy of batch mixing from the "unique" population (averaging over all six perturbations) is low in all three methods (scVI, scANVI and Seurat Alignment), with a slight advantage for scVI and scANVI. **Figure 2.9h-i** shows the harmonization statistics for each population, averaging over all shared cell types between the two datasets. Evidently, for the populations that are indeed common to the two datasets, scVI and scANVI are capable of mixing them properly, while preserving the original structure, comparing favorably to Seurat Alignment on both measures. Overall, the results of this analysis demonstrate that scVI and scANVI are capable of harmonizing datasets with very different compositions, while not forcing erroneous mixing. These results are consistent with the design of scVI and scANVI, which aim to maximize the likelihood of a joint generative model, without making *a priori* assumptions about the similarity in the composition of the input datasets.

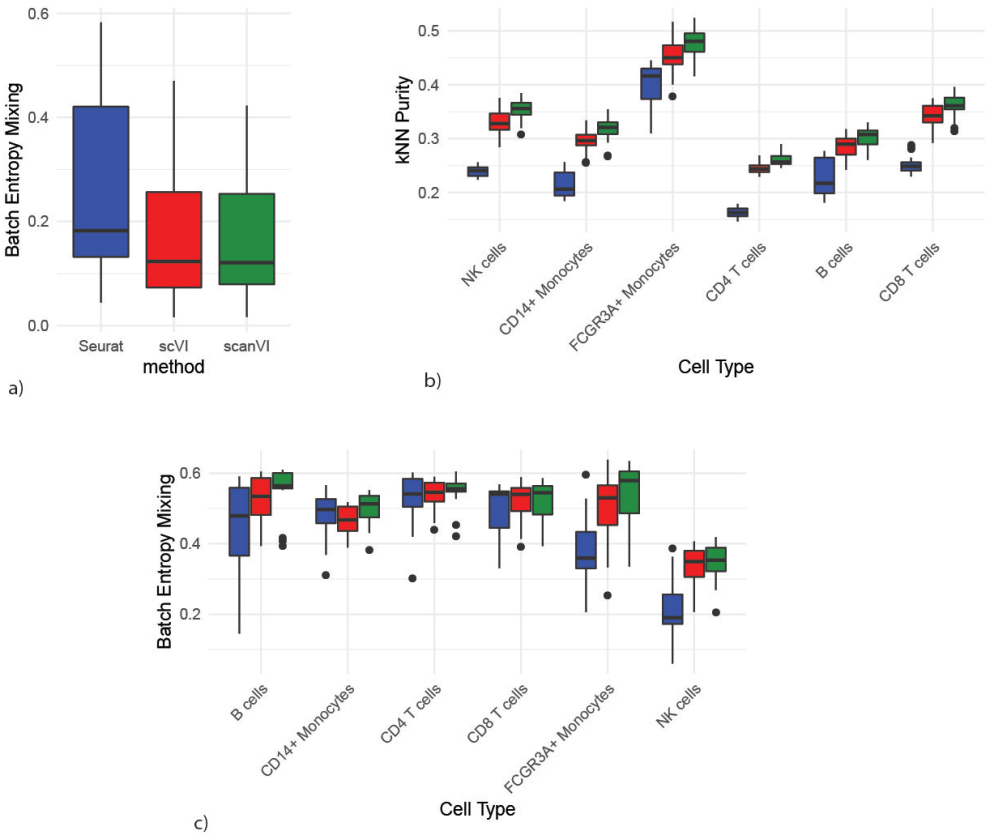


Figure 2.10: Supplementary study of harmonizing datasets with different cellular composition

Supplementary study of harmonizing datasets with different cellular composition. We show here the case where each of the two datasets has a unique cell types and share all the others. For each box plot, we report over all the possible combinations of left-out cell types. (a) Entropy of batch mixing for the unique population (lower is better). (b) k-nearest neighbor purity (unique and non-unique; higher is better). (c) Entropy of batch mixing for the non-unique populations (higher is better). The boxplots are standard Tukey boxplots where the box is delineated by the first and third quartile and the whisker lines are the first and third quartile plus minus 1.5 times the box height. The dots are outliers that fall above or below the whisker lines.

In a similar but more complex experiment, we also study the case when the two datasets both have their own unique cell types but also share several common cell types. Populations unique to each dataset have low mixing (**Figure 2.10(a)**), especially with scVI and scANVI. Conversely, the shared populations have a substantially higher mixing rate (**Figure 2.10(c)**). Specifically, scANVI and scVI both mix shared populations better than Seurat, with a better overall performance for scANVI. Finally, the preservation of original structure is higher scVI and scANVI when compared to Seurat across all cell types, especially for B cells, NK cells and FCGR3A$^+$ Monocytes (**Figure 2.10(b)**). Overall, these results demonstrate that our methods do not tend to force wrong alignment of non-overlapping parts of the input datasets.

Harmonizing continuous trajectories

While so far we considered datasets that have a clear stratification of cells into discrete sub-populations, a conceptually more challenging case is harmonizing datasets in which the major source of variation forms a continuum, which inherently calls for accuracy at a higher level of resolution.

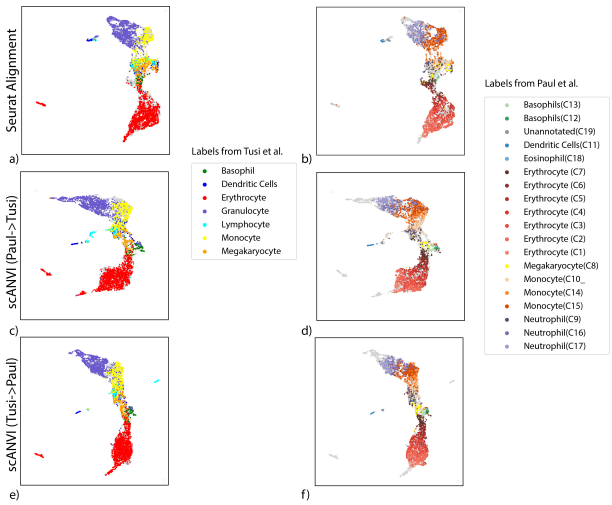


Figure 2.11: Follow-up analysis on continuous trajectory harmonization with scANVI

Follow-up analysis on continuous trajectory harmonization with scANVI ($a-b$) Continuous trajectory obtained by the Seurat Alignment procedure for the HEMATO-Tusi and the HEMATO-Paul datasets. ($c-d$) Continuous trajectory obtained by the scANVI using the Tusi cell type labels for semi-supervision. ($e-f$) Continuous trajectory obtained by the scANVI using the Paul cluster labels for semi-supervision. All locations for scatter plots are computed via UMAP in their respective latent space.

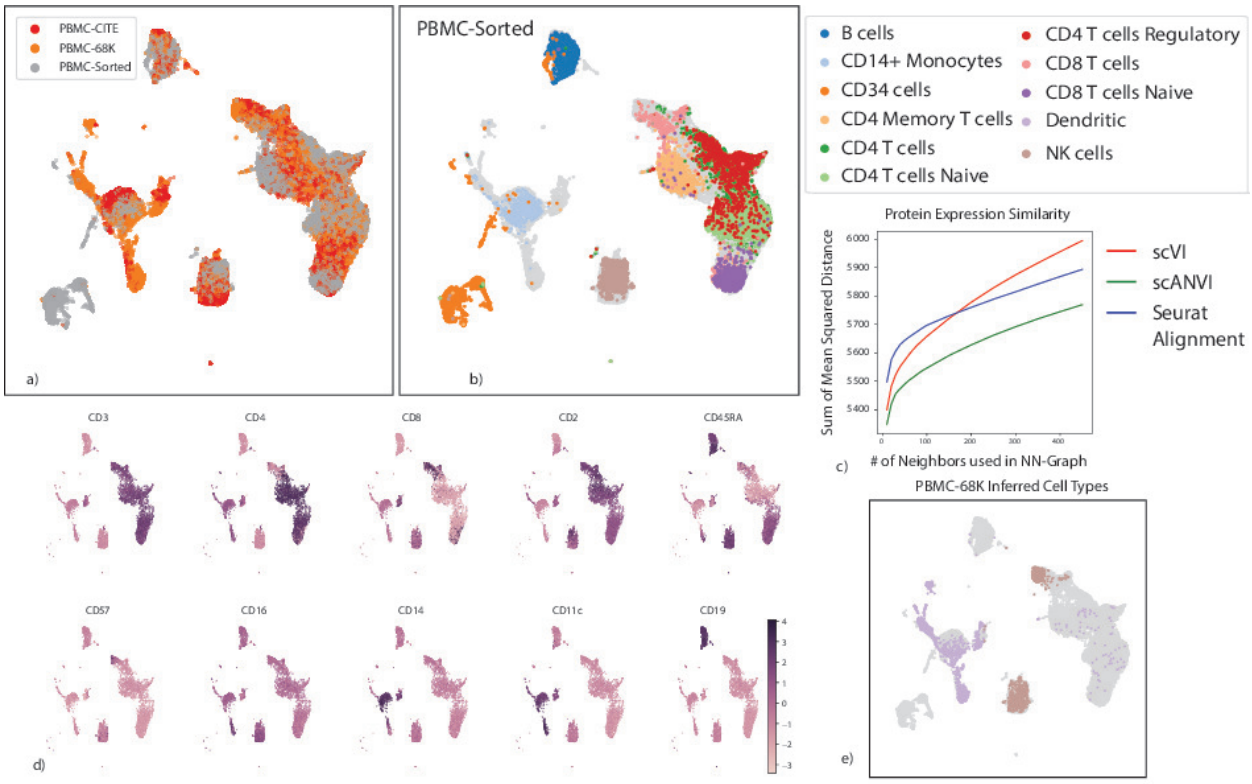


Figure 2.12: Harmonizing developmental trajectories

Harmonizing developmental trajectories

$(a-b)$ UMAP visualization of the scVI latent space, with cells colored by the original labels from either the HEMATO-Paul (a) or HEMATO-Tusi (b) studies. The cells from the other dataset are colored in gray. (c) Entropy of batch mixing along 20 bins of the HEMATO-Tusi cells, ordered by the potential of each cell. Potential is a pseudotime measure that describes the differentiation state of a cell using the population balance analysis algorithm (center: common myeloid progenitors; moving left: erythrocyte branch; moving right: granulocyte branch). (d) k-nearest neighbors purity for scVI, Seurat, and scANVI. (e) Expression of marker genes that help determine the identity of batch-unique cells.

To explore this, we use a pair of datasets that provides a snapshot of hematopoiesis in mice (HEMATO-Tusi [162], HEMATO-Paul [163]; **Figure 2.12**). These datasets consist of cells along the transition from common myeloid progenitor cells (**Figure 2.12a-b**; middle) through two primary differentiation trajectories myeloblast (top) and erythroblast-megakaryocyte (bottom). Notably, the HEMATO-Tusi dataset contains cells that appear to be more terminally differentiated, which are located at the extremes of the two primary branches. This can be discerned by the expression of marker genes (**Figure 2.12(e)**). For instance the HEMATO-Tusi unique erythroid cell population expresses *Hba-a2* (hemoglobin subunit) and *Alas2* (erythroid-specific mitochondrial 5-aminolevulinate synthase) that are known to be present in reticulocytes [173, 174]. At the other end, the granulocyte subset that is captured only by HEMATO-Tusi expresses *Itgam* and *S100a8*. *S100a8* is a neutrophil specific gene predicted by Nano-dissection [175] and is associated with GO processes such as leukocyte migration associated with inflammation and neutrophil aggregation. *Itgam* is not expressed in granulocyte-monocyte progenitor cells but is highly expressed in mature monocytes, mature eosinophils and macrophages [176]. We therefore do not expect mixing to take place along the entire trajectory. To account for this, we evaluated the extent of batch entropy mixing at different points along the harmonized developmental trajectory. As expected, we find that in most areas of the trajectory the two datasets are well mixed, while at the extremes, the entropy reduces significantly, using either scVI or Seurat Alignment (**Figure 2.12(c)**). Overall, we observe that scVI compares well in terms of both mixing the differentiation trajectories in each dataset and preserving their original, continuous, structure (**Figure 2.12a-d**).

To validate scANVI in this context as well, we provided it with the categorical labels of cells along the two developmental trajectories, indicating their cell state (**Figure 2.12c-d** and **Figure 2.11**). Even though this labeling scheme does not explicitly account for the ordering between states, we observe that scANVI is capable of mixing the two datasets, while retaining their original structure, achieving a level of accuracy comparable to that of scVI and better than that of Seurat Alignment. We also test the effect of low quality data in this example where cell types are not clearly demarcated. We observe consistent results, in terms of relative performance between methods, for decreasing rates of sampling in **Figure 2.13**.

Harmonizing datasets across species

Another more challenging data harmonization scenario is when the two datasets come from different species. Although species share homologous genes, more dataset specific expression patterns are expected in across-species comparison. We harmonized two datasets from mouse [164] and human [149] Substantia Niagra after mapping homologous genes using the Informatics Web Site [177]. We visualized the UMAP of the harmonized latent space by scVI and Seurat Alignment (**Figure 2.14(a)**). Both methods perform well in terms of preserving the cluster structure in the original mouse dataset, as well as mixing the cells from different species. We compare the different harmonization methods more systemati-

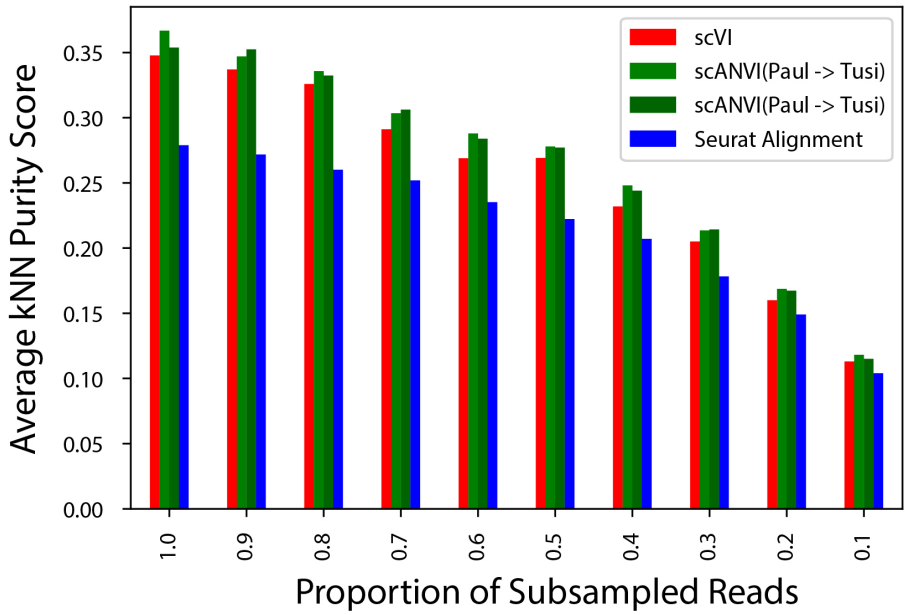


Figure 2.13: Evaluation of harmonization metric when data quality is corrupted

Evaluation of harmonization metric when data quality is corrupted
Average kNN purity by scVI, scANVI and Seurat Alignment when lower quality data is simulated by downsampling to 10-90% of the original transcript counts. 10% of the reads are removed from the dataset at each step, and the change in average kNN Purity score is shown on the y-axis.

cally using the kNN purity and entropy of batch mixing (**Figure 2.14(b)**). In this test, we find consistently superior performance of scVI and scANVI.

Rapid integration of multiple datasets

To demonstrate the scalability of our framework in the context of harmonizing multiple (and possibly large) dataset, we ran scVI to integrate a cohort of 26 datasets spanning 105,476 cells from multiple tissues and technologies, which was made available by the authors of Scanorama (a method based on truncated singular value decomposition followed by nearest neighbor matching [150]). Using the hardware specified in the original paper [150] (Intel Xeon E5-2650v3 CPU limited to 10 cores with 384 GB of RAM), Seurat Alignment and MNN required over 24 hours, while Scanorama completed its run in 20 minutes. Using a simpler configuration (eight-core Intel i7-6820HQ CPU with 32 GB RAM) along with one NVIDIA Tesla K80 GPU (GK210GL; addressing 24 GB RAM), we found that scVI integrates all datasets and learns a common embedding in less than 50 minutes. This running time is

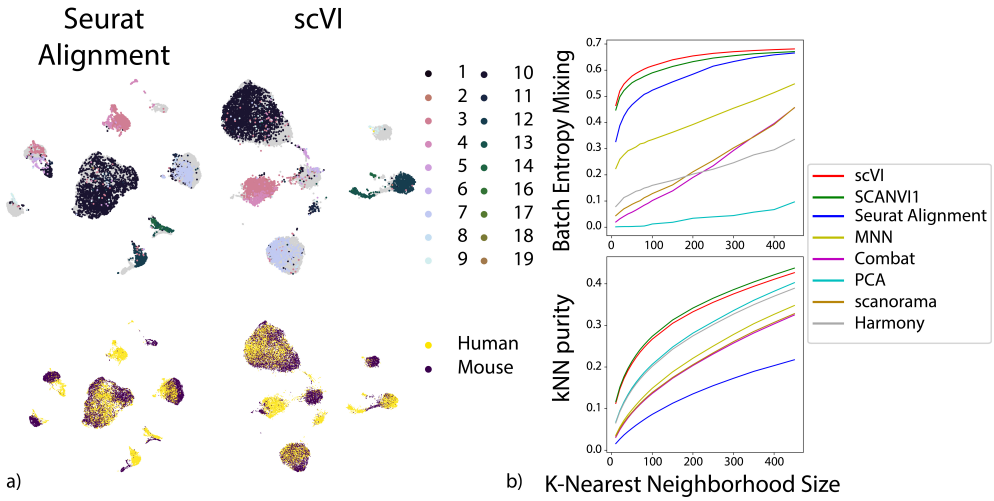


Figure 2.14: Evaluation of harmonization metrics in cross-species comparison

Evaluation of harmonization metrics in cross-species comparison. The harmonization performance of scVI and scANVI on datasets from human and mouse Substantia Nigra. (a) shows the distribution of mouse cell clusters and species origin on scVI and Seurat alignment latent space visualized by UMAP. The mouse cell cluster ids are provided by the original publication. (b) shows kNN purity and Batch Entropy Mixing of different methods on the cross species comparison as a function of the K-Nearest Neighborhood size.

competitive considering the reduced memory availability and the increased complexity of our model, compared to that of Scanorama. Notably, all the downstream analyses, such as annotation, differential expression or visualization can be operated by accessing the latent space or via forward passes through the neural networks. Since these access operations can be conducted very efficiently [43], the dominant factor, on which we focused our run time analysis, is the time required for model fitting. Considering the results, the latent space of scVI recapitulates well the major tissues and cell types (**Figure 2.15**), and the position of cells in the latent space provides an effective predictor for the cell type label (**Figure 2.15** and **Method**).

We also evaluated the runtime of scVI and scANVI on the four smaller dataset pairs we used for benchmarking. We report this metric as a function of the size of the dataset, and compared it to other models used in this paper. The runtime of scVI and scANVI increases as the number of genes increases (**Appendix Table 2.3**), but depends largely on the computational resources available at the time, and scales sublinearly. It is thus feasible to run scVI and scANVI with a much larger geneset. However, using more genes does not guarantee better performance, as performance decreases when the number of genes becomes comparable to the number of cells [178].

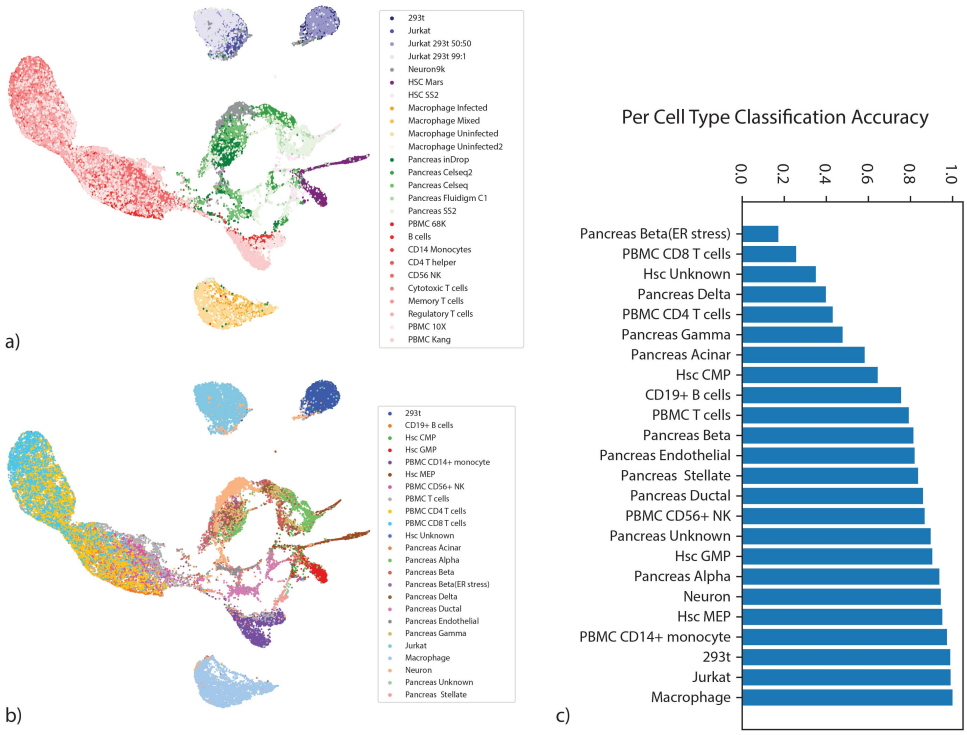


Figure 2.15: Large-scale data integration with scVI

Large-scale data integration with scVI.
$(a-b)$ UMAP visualization of the scVI latent space colored by datasets (a) and by cell types (b).
(c) accuracy of a nearest neighbor classifier based on scVI latent space

Transferring cell type annotations between datasets

We next turned to evaluate scVI and scANVI in the context of harmonization-based annotation. Here, we test the extent to which annotations from a previously annotated dataset can be used to automatically derive annotations in a new unannotated dataset. For scVI and Seurat Alignment, we derive the annotations by first harmonizing the input datasets and then running a k-nearest neighbors classifier (setting k to 10) on the joint latent space, using the annotated cells to assign labels to the unannotated ones. Conversely, scANVI harmonizes the input datasets while using any amount of available labels. The prediction of unobserved labels is then conducted using the approximate posterior assignments $q_\Phi(c \mid x)$ of cell types, directly derived from the model (**Method**). An alternative approach that we benchmark against was taken by scmap-cluster [65]. scmap directly builds a classifier based on the labeled cells (instead of performing harmonization first) and then applies this classifier to the unlabeled cells. Finally, we also applied the domain adaptation method Correlation Alignment for Unsupervised Domain Adaptation (CORAL, [179]). This method was not

	Dentate Gyrus				
ngenes	**733**	**1,527**	**3,146**	**6,100**	**10,665**
Seurat	13.7	20.7	26.5	46.0	78.7
PCA	1.6	1.6	1.7	1.8	1.7
scVI	223.8	241.9	250.4	268.4	281.2
scANVI1	126.9	137.4	158.9	169.1	275.4
scANVI2	59.6	66.2	76.2	81.3	130.6
SCMAP	52.1	51.6	53.1	66.7	69.6
MNN	154.7	273.9	591.4	1141.6	2060.3
Combat	11.4	7.8	26.2	33.2	52.7
scanorama	50.3	29.1	37.5	61.7	37.8
Harmony	13.0	6.2	11.1	10.5	6.5

	MarrowTM				
ngenes	**876**	**1,731**	**3,407**	**6,546**	**11,224**
Seurat	21.9	34.5	42.4	71.6	123.4
PCA	2.0	2.0	2.0	2.0	2.1
scVI	289.4	290.5	298.0	305.1	419.6
scANVI1	159.5	151.2	162.2	178.5	236.7
scANVI2	115.3	129.5	127.8	143.0	187.5
SCMAP	74.8	78.5	82.5	84.9	134.1
MNN	410.6	769.1	1424.9	2471.2	4082.2
Combat	6.3	11.0	21.4	44.9	100.6
scanorama	48.0	50.9	78.9	66.2	105.5
Harmony	20.6	20.3	21.0	20.9	27.8

	PBMC8KCITE				
ngenes	**664**	**1309**	**2699**	**5623**	**10352**
Seurat	36.8	63.0	67.2	111.6	186.0
PCA	3.3	3.1	3.0	3.0	3.2
scVI	409.0	441.0	421.3	458.6	498.7
scANVI1	208.6	298.5	230.2	254.9	293.5
scANVI2	182.0	172.8	245.0	214.5	235.3
SCMAP	48.8	50.9	49.1	56.3	64.5
MNN	866.4	1412.1	1547.8	4915.6	8644.6
Combat	12.6	11.9	17.1	36.9	61.4
scanorama	39.5	49.9	48.1	51.8	58.6
Harmony	11.4	11.5	10.6	12.9	13.3

	Pancreas				
ngenes	**688**	**1,346**	**2,674**	**5,326**	**10,481**
Seurat	23.3	33.5	38.8	61.9	109.1
PCA	2.2	2.2	2.3	2.3	2.2
scVI	324.6	332.7	340.6	350.4	346.9
scANVI1	211.4	210.0	243.9	239.1	260.6
scANVI2	182.0	172.8	245.0	214.5	235.5
SCMAP	75.9	73.0	75.5	78.3	87.0
MNN	211.2	436.5	721.2	1335.2	2790.0
Combat	14.3	15.0	27.2	37.0	66.8
scanorama	52.7	54.2	55.1	58.6	60.0
Harmony	24.9	25.4	25.3	25.7	25.7

Table 2.3: Runtime in seconds for all the algorithms considered in this study.

initially developed for single-cell analysis but is an insightful benchmark from the machine learning literature.

We start by exploring the four dataset pairs in **Figure 2.8**, which have been annotated in their respective studies. In each experiment, we "hide" the cell type annotations from one dataset and transfer the second dataset labels to the first one. As a measure of performance, we report the weighted accuracy, which is the percent of cells that were correctly assigned to their correct (hidden) label, averaging over all labels (**Method**). Importantly, the annotations in this first set of case studies were derived computationally. For example, by first clustering the cells, looking for marker genes expressed by each cluster and then assigning labels to the clusters accordingly. This level of annotation therefore makes the prediction problem relatively easy, and indeed, while we find that overall scANVI predicts unobserved labels more accurately, the differences between the methods are mild (**Figures 2.16 and 2.17**). Notably, CORAL achieves overall competitive performance except when transferring labels on the MarrowTM pairs, from 10x to Smart-Seq2. In this specific instance, CORAL maps

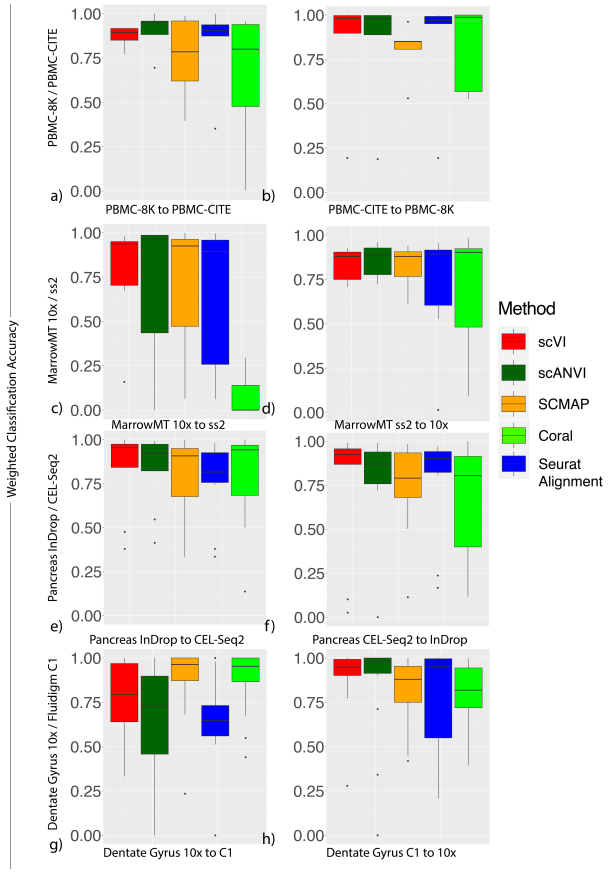


Figure 2.16: Annotation results for all four dataset pairs (boxplot)

Annotation results for all four dataset pairs.
PBMC-8K / PBMC-cite ($a - b$), MarrowMT-10X / MarrowMT-SS2 ($c - d$), Pancreas InDrop-CELSeq2 ($e - f$) and Dentate Gyrus 10X / Fluidigm C1 ($g - h$). Accuracies for transferring annotations from one dataset to another from a k-nearest neighbors classifier on Seurat Alignment, and scVI latent space, scANVI, SCMAP and CORAL classifier are shown. The aggregated results across for cell types that are shared between the two datasets is shown in box plots.

most of the cells to a single label (incidentally, while this label marks cells that are transcriptionally similar, it is defined by the authors as an unknown class "NA", corresponding to cells that cannot be confidently assigned or low quality cells according to the authors of [141]), which might be due to its linear transformation of the feature space.

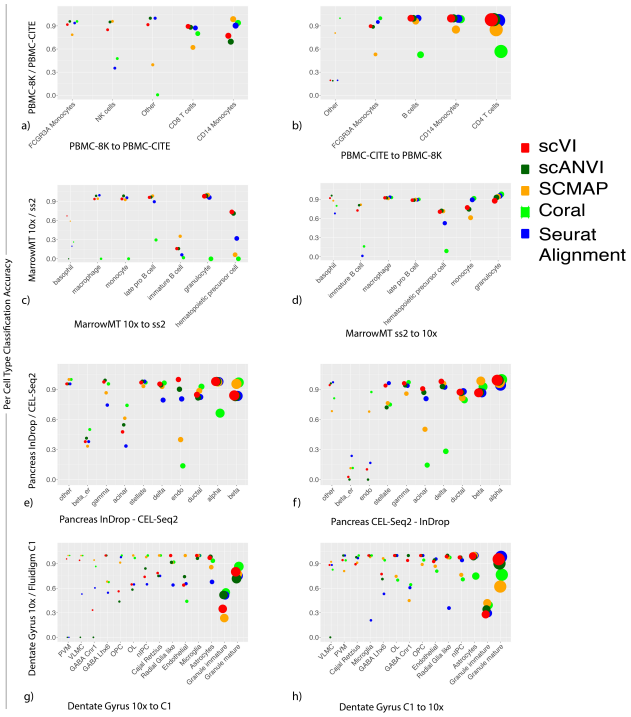


Figure 2.17: Annotation results for all four dataset pairs (bubbleplot)

Annotation results for all four dataset pairs. PBMC-8K / PBMC-cite ($a - b$), MarrowMT-10X / MarrowMT-SS2 ($c - d$), Pancreas InDrop-CELSeq2 ($e - f$) and Dentate Gyrus 10X / Fluidigm C1 ($g - h$). Accuracies for transferring annotations from one dataset to another from a k-nearest neighbors classifier on Seurat Alignment, and scVI latent space, scANVI, SCMAP and CORAL classifier are shown. The prediction accuracy for each cell type that is shared between the two datasets is shown on the y-axis and the size of the dots are proportional to the proportion of a cell type in the total population.

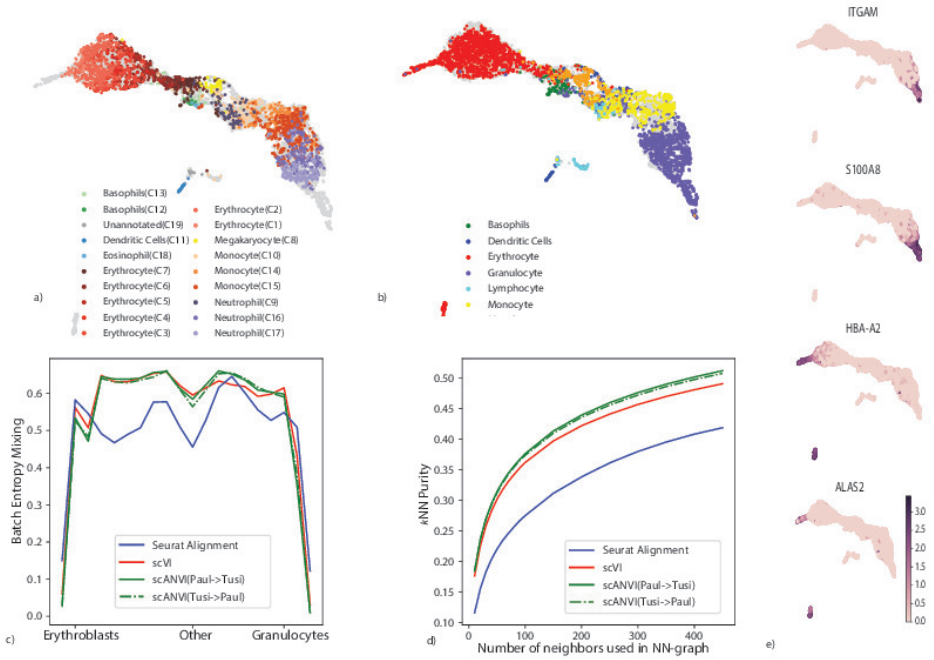


Figure 2.18: Validation of cell type annotations using additional metadata

Validation of cell type annotations using additional metadata. (a-b) UMAP plot of the scANVI latent space inferred for three harmonized datasets: PBMC-CITE, PBMC-sorted, and PBMC-68K. Cells are colored by the dataset of origin (a) and the PBMC-sorted labels (b). Cells from the PBMC-CITE and PBMC-68K are colored in gray in (b). (c) The consistency of the harmonized PBMC-CITE mRNA data with the respective protein measurements, evaluated by mean squared error and for different neighborhood size. Lower values indicate higher consistency. (d) UMAP plot of the scANVI latent space, where cells are colored by normalized protein measurement. Only PBMC-CITE cells are displayed. (e) UMAP plot of the scANVI latent space, with cells from the PBMC-68k dataset colored according to their original label. For clarity of presentation, only cells originally labeled as dendritic cells or natural killer cells are colored. Evidently, a large number of these cells are mapped to a cluster of T-cells (right side of the plot).

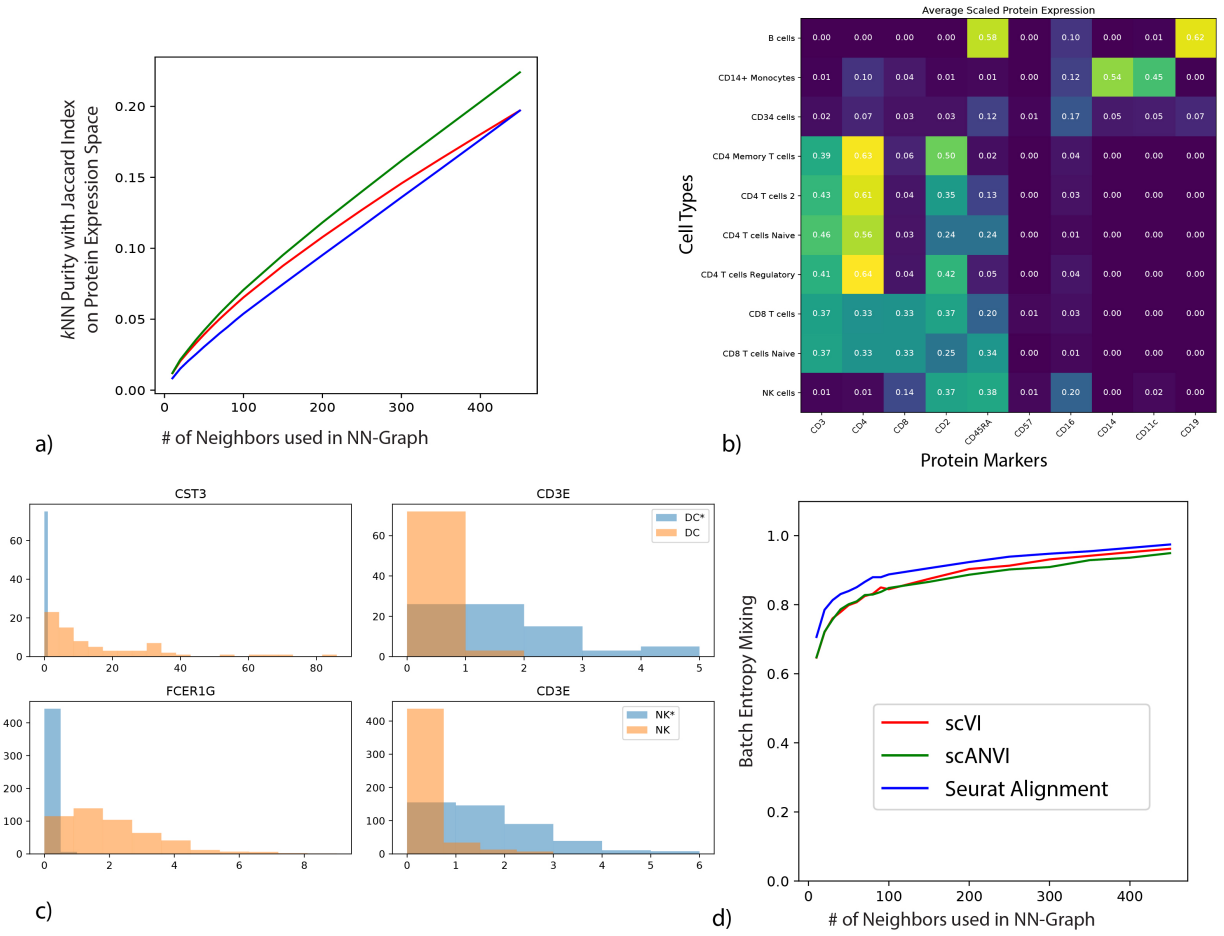


Figure 2.19: Supplementary study of labels concordance. (*a*) *k*-nearest neighbors purity of the merged latent space on the protein expression space as a function of the size of the neighborhood. (*b*) Protein expression heatmap showing consistency of PBMC-Sorted labels and protein expression in PBMC-CITE. The protein expression per cell type is based on *k*-nearest neighbors imputation from the harmonized latent space obtained from scANVI trained with pure population labels. (*c*) We select individual cells that were labeled as dendritic cells or Natural Killer cells in the original publication of the respective datasets, and compare the raw transcript count from cells inside the scANVI T cells cluster (DC*, NK*) against cells outside the T cells cluster (DC, NK). The expression of marker genes suggest that DC* and NK* is more likely to be T cells and thus the scANVI latent space is more accurate. (*d*) The batch entropy mixing of the three datasets in scVI, scANVI and Seurat Alignment merged space.

Cell Types	# cells
B cells	10,085
CD14+ Monocytes	2,612
CD34+ cells	9,232
CD4 T cells	11,213
CD56 NK cells	8,385
CD8 T cells	10,209
Memory T cells	10,224
Naive CD8 T cells	11,953
Naive T cells	10,479
Regulatory T cells	10,263

Table 2.4: Cell types present in the PBMC-sorted dataset

To evaluate the accuracy of annotations without the need for computationally-derived labels, we turned to the PBMC-CITE dataset which includes measurements of ten key marker proteins in addition to mRNA [90], and the PBMC-sorted dataset [45], where cells were collected from bead purifications for eleven cell types (**Appendix Table 2.4**). We applied scVI and scANVI to harmonize and annotate these two datasets along with a third dataset of PBMC (PBMC-68K [45]). Our analysis contains a combined set of $n = 169,850$ cells from the three datasets altogether. To generate a realistic scenario of cell type annotation, we only provide access to the experimentally-based labels from the PBMC-sorted dataset (**Figure 2.18a-c**). As an additional benchmark, we also evaluate Seurat Alignment, which was tested after removal of a randomly selected subset (40%) of the two large datasets (PBMC-68K and PBMC-sorted) due to scalability issues. Considering our harmonization performance measures (i.e., retainment of the original structure and batch mixing), we observe as before that scVI and scANVI perform similarly and compare favorably to Seurat Alignment. We then evaluated the accuracy of assigning unobserved labels, focusing on the PBMC-CITE dataset. Instead of using the labels from the original PBMC-CITE study as ground truth (which were computationally derived), we used the protein data, which provides an experimentally-derived proxy for cell state. To this end, we quantified the extent to which the similarity between cells in the harmonized mRNA-based latent space is consistent with their similarity at the protein level (**Method**). We first computed the average discrepancy (sum of squared differences) between the protein measurements in each cell and the average over its k-nearest neighbors. As a second measure we computed for each PBMC-CITE cell the overlap between its k-nearest PBMC-CITE neighbors in the harmonized mRNA-based space and in the protein space. We then report the average across all cells in **Figure 2.19**. Evidently, scANVI outperformed both scVI and Seurat Alignment for a wide range of neighborhood sizes, providing a representation for the mRNA data that is more consistent with the protein data (**Figure 2.18(c)**).

Cell type annotation in a single dataset based on "seed" labels

An important variant of the annotation problem lies within the context of an *ab initio* labeling of a single dataset where only a subset of the cells can be confidently annotated based on the raw data. This increasingly prevalent scenario may result from limited sensitivity of the scRNA-seq assay, where marker genes may only be confidently observed in a small subset of cells. One common way to address this problem is to compute some form of a distance metric between cells (e.g., after embedding with scVI or using Seurat PCA) and then assign labels based on proximity to annotated cells [45]. To benchmark our methods, we consider two such predictors: the first is clustering the cells and taking a majority vote inside each cluster, and the second is taking the majority vote of the k-nearest neighbors around each unannotated cell ($k = 10$). While these approaches are quite straightforward, their accuracy might suffer when the data do not form clear clusters [162], or when differences between labels are too subtle to be captured clearly by a transcriptome-wide similarity measure. To address these issues, scANVI takes an alternative approach, namely learning a latent embedding that is guided by the available labels, and then producing posterior probabilities for assigning labels to each cell.

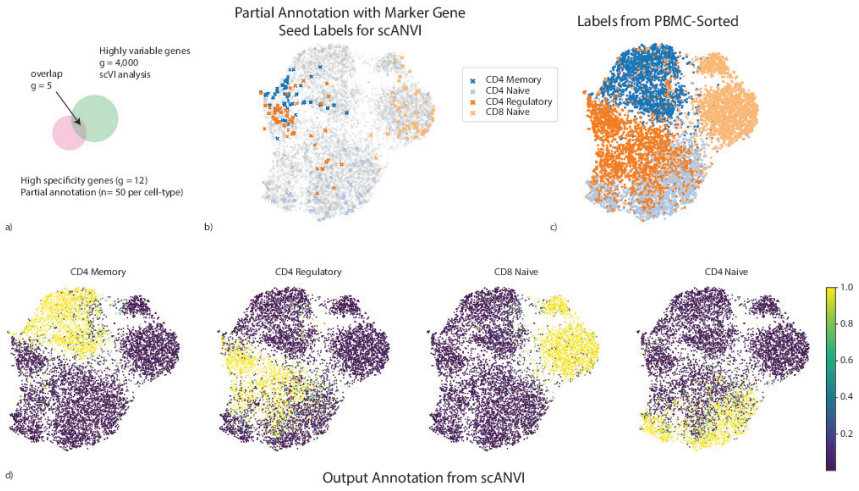


Figure 2.20: Cell type annotation in a single dataset using "seed" labeling

Cell type annotation in a single dataset using "seed" labeling. (a) discrepancies between marker genes that can be used to confidently label cells and highly variable genes in scRNA-seq analysis. ($b - d$) UMAP plot of the scVI latent space. (b) Seed cells are colored by their annotation (using known marker genes). (c) PBMC-sorted cell type labels from the original study based on marker-based sorting (d) The posterior probability of each cell being one of the four T cell subtypes obtained with scANVI.

As a case study, we compiled a dataset consisting of several experimentally sorted and labeled subsets of T cells from the PBMC-sorted dataset, including CD4 memory, CD4 naive, CD4 regulatory and CD8 naive. To make our analysis more realistic, we assume that the labels are completely unknown to us and therefore assign each T cell to its respective subset using marker genes (12 altogether; see **Method**). Notably, several important biomarkers (*CD4*, *CTLA4*, and *GITR*) are detected in less than 5% of the cells. This renders their use for annotation not straightforward. Furthermore, many of these biomarkers are sparsely expressed to the extent that they are likely to be filtered out in the gene selection step of most harmonization procedures (**Figure 2.20(a)**).

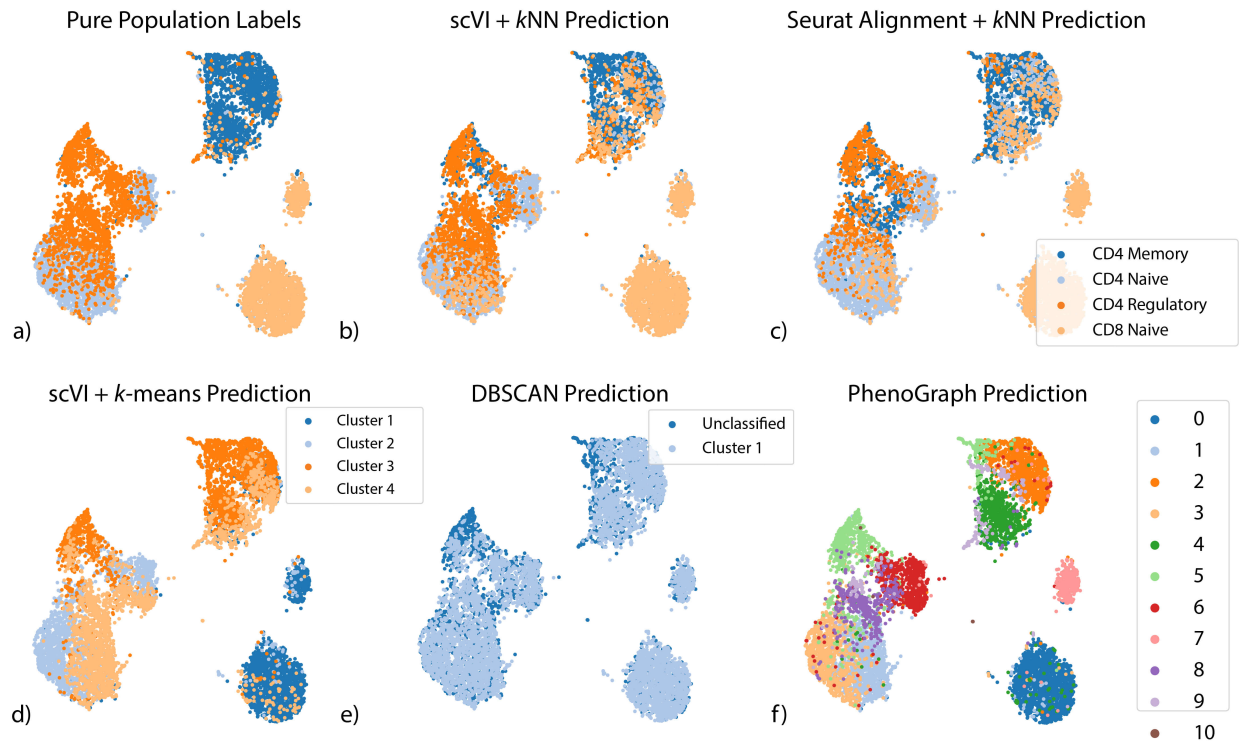


Figure 2.21: Other methods of classifying T-cell subsets of the PBMC-Pure dataset

Other methods of classifying T-cell subsets of the PBMC-Pure dataset. Coordinates for the scatter plots are derived from UMAP embedding based on the latent space of scANVI. (a) Ground truth labels from the purified PBMC populations (b) k-nearest neighbors classification labels when applied on scVI latent space from the seed set of cells (c) k-nearest neighbors classification labels when applied on Seurat Alignment latent space (d) k-means clustering based labels when applied to scVI latent space (e) DBSCAN clustering based labels when applied to scVI latent space. DBSCAN returns only one cluster but return some cells as unclassified. (f) PhenoGraph clusters on scVI latent space

To analyze this dataset, we first computed a signature score for each cell and for each label (i.e., T cell subset) using the scaled raw expression values of the respective marker genes (**Method**). We then designated the top 50 scoring cells in each subset as the seed set of cells that are confidently annotated for that subset (**Figure 2.20(b)**). Reassuringly, this partial annotation is in agreement with the experimentally-derived cell type labels available for this dataset (**Figure 2.20(c)**). However, this dataset does not form clear clusters, and in particular the seed sets of cells are not well separated. Such an observation makes clustering-based approaches potentially less precise. Indeed, using k-means clustering on the scVI and Seurat PCA latent space, we find that 74% and 72% of the cells were assigned with their correct label. Similar analysis with two additional popular clustering algorithms (DBSCAN [180] and PhenoGraph [181]) further emphasizes the challenge of a cluster-based approach on this data. Specifically, DBSCAN does not partition the data into more than one cluster (scanning through a large number of parameter values; **Method**), and PhenoGraph predicts 9 clusters and achieves an accuracy of 41% (**Figure 2.21**).

Consistent with these results, the application of a k-nearest neighbors classifier resulted in a similar level of accuracy in the Seurat PCA latent space (71%), which is slightly improved when replacing it with the scVI latent space (73%; **Figure 2.21**). Conversely, after fitting the scANVI model based on this partial labeling, the annotation posterior $q_\Phi(c \mid z)$ (**Figure 2.20(d)**) provides a substantially more accurate cell type assignment, with 84% of cells annotated correctly.

While scANVI has been designed to handle discrete (but not continuous) labels, we hypothesized that gradual transition between cell states may still be captured by the uncertainty of label assignment. We tested it using simulated data [87] that consists of a set of "end-point" states along with intermediary states that connect them (**Method**, **Figure 2.22**a). We provided labels only to end-point cells, and investigated the label assignment scores calculated for the intermediary cells. We find that scANVI provides a range of assignment probability values and that these values are proportional to the distance from the respective end points (**Figure 2.22**b-g). Conversely, the scores provided by scmap tend to be more extreme (**Figure 2.22**h-i), thus less reflecting the continuous nature of the data.

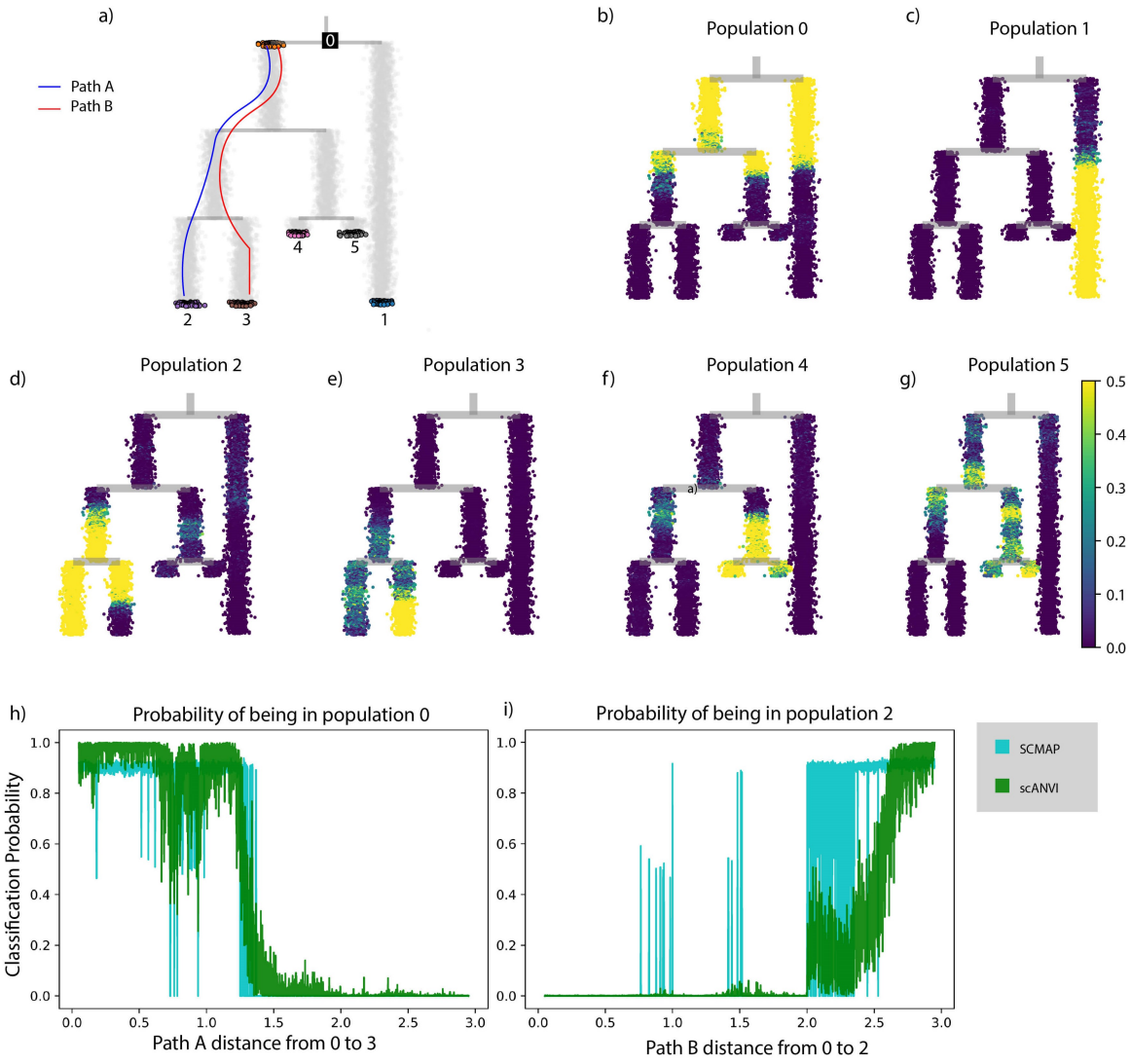


Figure 2.22: Continuous trajectory simulated using SymSim

Continuous trajectory simulated using SymSim.
(a) Tree structure from which the cells are sampled. Each grey dot represent a cell sampled along the trajectory. Colored dots with a black edge are treated as labeled, while the others are treated as unlabeled. Each path simulates a continuous phenotypical variation. ($b-g$) The same tree with each cell colored by the posterior probability of being assigned to a specific label. ($h-i$) Another visualization of the gradual change of posterior probability by plotting the posterior probability of root (h) and population 3 (i). The x-axis represents the pathwise distance (paths are defined in (a)), and the y-axis represents the probability, or confidence of the assignment.

Cell type taxonomy and hierarchical classification with scANVI

Another subtle yet important variation of the annotation problem is when the labels are not mutually exclusive but rather form a taxonomy of cell types or states. To effectively annotate cells in this setting, we extended scANVI to perform hierarchical classification, which as before we carry out from first principles, relying on probabilistic graphical models (**Method**). To demonstrate this extended version, we use a dataset of the mouse nervous system [48] that was annotated using a cell type taxonomy with several levels of granularity. At the lowest (most granular) level, the cells are stratified into 265 cell sub-types. At the second lowest level of granularity these 265 subtypes are grouped into 39 subsets, each corresponding to a more coarse definition of a cell type.

We evaluate the ability of scANVI as well as the competing methods at inferring the most granular level of labels when provided with partial "seed" annotation — namely label information for 5 randomly selected cells per label (which accounts for an overall of 0.8% of the cells). We first observe that Seurat PCA followed by a k-nearest neighbors classifier provides a weighted accuracy of 23% (averaging over all cell types). While this might seem like a low accuracy, it is in fact far from trivial since the expected weighted accuracy of a random classifier or a constant predictor is of around $1/265 \approx 0.3\%$. Such low numbers are due to the high number of labels at this highly granular scale. scVI provides a substantially better, yet still low level of accuracy at 32%. Interestingly, when scANVI is used without accounting for hierarchy, its performance is similar to the unsupervised scVI (at 32%), which might result from very large number of labels that may require hyperparameter tuning (e.g., increasing the number of classifier training epochs. However, when we take the hierarchy of the labels into account, the performance of scANVI increases to 37%, thus outperforming the other methods by a significant margin. Notably, while we tested the extrapolation of seed labeling and the hierarchical mode only in the context of a single dataset, this variation of the scANVI model can also be directly applied in the context of multiple datasets (i.e., transferring hierarchical annotations between datasets).

Hypotheses testing in harmonized datasets: the case of differential expression

With their probabilistic representation of the data, scVI and scANVI each provide a natural way of performing various types of hypotheses testing (**Method**). This is different from other approaches [144, 33, 149, 150, 34] where the dataset alignment procedures do not carry direct probabilistic interpretation, and the resulting harmonized data can thus not be directly used for these purposes.

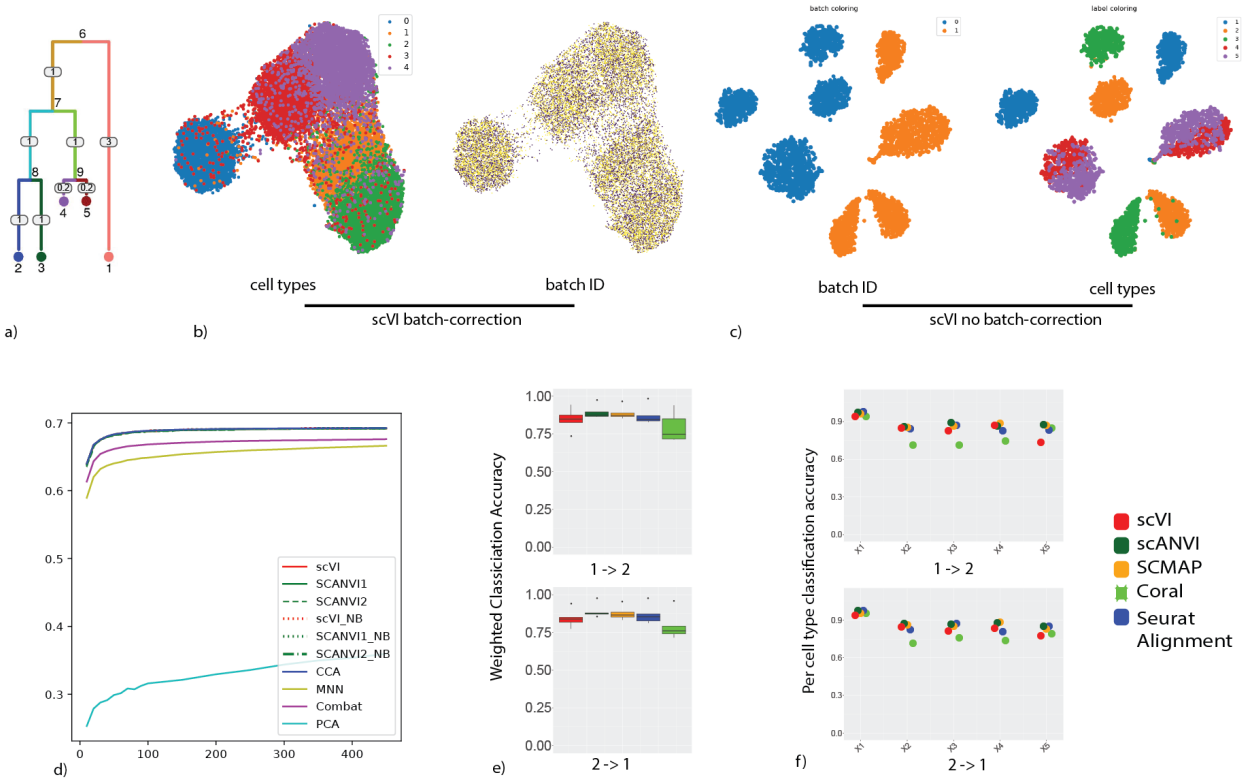


Figure 2.23: Presentation of the simulated dataset used for differential expression benchmarking

Presentation of the simulated dataset used for differential expression benchmarking. (a) The tree used to sample the cells in SymSim. We sample cells from the five leaves nodes representing five different cell types derived from the same root node. (b) UMAP of scVI latent space colored by cell types and batch identifier (c) UMAP of scVI latent space without batch correction, proving that the data is indeed subject to batch effects. (d) Entropy of batch mixing for all the algorithms (e) Weighted accuracy using a k-nearest neighbors classifier on the latent space (f) Per cell type accuracy for the label transfer.

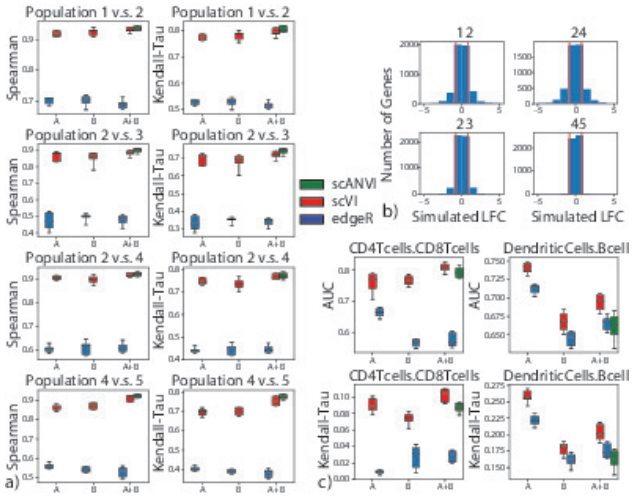


Figure 2.24: Differential Expression on multiple datasets with scVI

Differential Expression on multiple datasets with scVI. (a) distribution of true log fold change between all pairs of cell types for the simulated data. The pairs of cells are chosen to represent different levels of distance on the tree as in **Figure 2.23(a)**. The pairs of population from most distant to least distant are '12', '24', '23', '45'. (b) Evaluation of consistency with rank correlation and Kendall-Tau is shown for comparisons of multiple pairs of cell types in the simulated data. (c) Evaluation of consistency with the AUROC and Kendal Tau metric is shown for comparisons of CD4 vs CD8 T cells and B cells vs Dendritic cells on the PBMC-8K only (A), the PBMC-68k only (B) and the merged PBMC-8K / PBMC-68K (A+B) for scVI and edgeR. Error bars are obtained by multiple subsampling of the data to show robustness. boxplots are standard Tukey boxplots where the box is delineated by the first and third quartile and the whisker lines are the first and third quartile plus minus 1.5 times the box height. The dots are outliers that fall above or below the whisker lines. (d) Mislabeling experiment in differential expression in both the SymSim simulated datasets and in the PBMC8K and PBMC68K dataset. The top row shows differential expression results for the correctly labeled population pair (Population 1 v.s. Population 2 in simulated dataset and CD4 T cells v.s. CD8 T cells in PBMC dataset. The bottom row shows differential expression results for the mislabelled population pair (Population 2 v.s. Population 3 in simulated dataset and Dendritic Cells v.s. B cells in PBMC dataset). For all, x-axis represents the proportion of flipped labels.

To demonstrate this, we focus on the problem of differential expression. As a first case study, we use two of the PBMC datasets (PBMC-8K and PBMC-68K) and looked for differentially expressed genes in two settings: comparing the B cells to dendritic cells, and similarly for CD4+ versus CD8+ T cells. For evaluation, we used reference sets of differentially expressed genes that were obtained from published bulk-level analysis of similar cell subsets (microarrays, [182, 183], as in [43]). While this benchmark relies on real data, a clear caveat is the lack of a well defined ground truth. To address this, we used a second benchmark based on simulations with Symsim [87]. The simulated data consists of five subpopulations of varying degrees of transcriptional distance, profiled in two different "batches" of different technical quality (**Method**). This framework allowed us to derive an exact log fold changes (LFC) between every pair of simulated subpopulations, which enable a more accurate evaluation of performance (**Figure 2.24 a**).

In both benchmark studies, we assume that labels are only available for one of the two input batches or datasets (in the real data we assume that PBMC-8K is the annotated one). To apply scVI, we first harmonized the input pair of datasets and transferred labels using a k-nearest neighbors classifier on the joint latent space ($k = 10$). We then consider these annotations (predicted and pre-labeled) as fixed and sample 100 cell pairs, each pair consisting of one cell from each population. For each cell pair we sample gene expression values from the variational posterior, while marginalizing over the different datasets, to compute the probability for differential expression in a dataset-agnostic manner. Aggregating across all selected pairs results in approximate Bayes factors that reflect the evaluated extent of differential expression (**Method**). Since scANVI assigns posterior probability for associating any cell to any label, it enables a more refined scheme. Specifically, instead of sampling pairs of cells we are sampling pairs of points in the latent space, while conditioning on the respective label. This approach therefore does not assume a fixed label for each cell (or point in latent space) as in the scVI scheme, but rather a distribution of possible labels thus making it potentially more robust to mis-labeling. For reference, we also included edgeR [184] using the same labels as scVI. Notably, edgeR was shown to perform well on scRNA-seq data [185] and uses a log-linear model to control for technical sample-to-sample variation.

In our simulations, we considered differential expression between every possible pair out of the five simulated subpopulations. For evaluation, we computed the Spearman and Kendall rank correlation coefficients between the true LFC and the inferred Bayes factors (for scVI and scANVI) or estimated LFC (for edgeR). Our results in **Figure 2.24(a)** show that with this artificial, yet more clearly defined objective, scVI was substantially more accurate than edgeR and that in the harmonized data scANVI provided more exact and stable estimates than scVI. The difficulty of each paired comparison is visualized by histograms of the simulated LFC (**Figure 2.24 b**).

To evaluate performance on the real data, we defined genes as differentially expressed if the adjusted p-value in the reference bulk data (provided by [182, 183]) was under 5%. Considering these genes as positive instances, we calculated the area under the ROC curve (AUROC) based on rank ordering the inferred Bayes factors (for scVI and scANVI) or p-values (for edgeR). Since the definition of positives genes required a somewhat arbitrary

threshold, we also used a second score that evaluates the reproducibility of gene ranking (bulk reference vs. single-cell; considering all genes), using the Kendall rank correlation coefficient (**Figure 2.24(c)**). As a reference, we look at the accuracy of differential expression analysis in each PBMC dataset separately (using their prior annotations to define the sets of cells we are comparing), which can be computed with scVI (as in [43]) and edgeR. Reassuringly, we observe that the performance of scVI on the joint data is not lower than it is in either dataset in isolation. We also find that while scVI performs moderately better than scANVI, both methods compare favorably to edgeR in terms of accuracy.

Mislabeling of a certain proportion of cells in a dataset is a plausible scenario that may occur in any study. An important challenge is therefore to maintain the validity of downstream analysis despite such "upstream"annotation errors. To evaluate robustness in this setting, we repeated the simulation analysis, while introducing labeling errors at different rates. Specifically, prior to evaluating differential expression between two simulated subpopulations, we flip the labels of a certain proportion (up to 30%) of the respective cells in the annotated batch. We then proceed as before and assign labels to cells in the unannotated batch by scVI or scANVI, followed by differential expression analysis. Our results (**Figure 2.24(d)**) suggest that scANVI is clearly more robust to this type of mislabeling than scVI (or edgeR, applied on the scVI- derived labels). Repeating the same analysis on the PBMC data (where the differential expression ground truth is obviously not available), we observe similar level of robustness in scANVI, albeit with not much difference compared to scVI and edgeR.

Overall, our results demonstrate that both scVI and scANVI are capable of conducting differential expression effectively, while working directly on a harmonized dataset. Furthermore, we observe that both methods and especially scANVI are robust to mislabeling, providing further motivation for explicitly modeling label uncertainty.

2.3 Method

scANVI: an extension to scVI for semi-supervised annotation

scVI is a hierarchical Bayesian model [186] for single-cell RNA sequencing data with conditional distributions parametrized by neural networks. The graphical model of scVI (**Figure 2.1(c)**) is designed to disentangle technical signal (i.e., library size discrepancies, batch effects) and biological signal. We propose in this manuscript an extension of the scVI model to include information about cell types in the generative model. We name this extension scANVI (single cell ANnotation using Variational Inference).

The generative model for scANVI

In our generative model, we assume each cell n is an independent realization of the following generative process. Let K be the number of datasets and C be the number of cell types across all datasets (including cell types that are not observed). Let $\mathbf{c}$ describe the expected

proportion of cells for each cell type. As in general this information is not available to the user, we consistently use a non-informative prior $\mathbf{c} = 1/C$ in the manuscript. Although some prior information about proportions of cell type is generally accessible, we observe that using the non-informative prior allows us to recover the correct proportion of cells. In addition, in comparative studies such as disease case-control comparisons, or between tissue comparisons of immune cells [58] we might not want to bias the estimate of cell-type proportion by prior knowledge. All in all, adjustment of the prior $\mathbf{c}$ is not required. Latent variable

$$c_n \sim \text{Multinomial}(\mathbf{c}), \tag{2.1}$$

describes the cell type of the cell n. Latent variable

$$u_n \sim \mathcal{N}(0, I), \tag{2.2}$$

is a low-dimensional random vector describing cell n within its cell type. Conceptually, this random variable could describe cell-cycles or sub-cell types. By combining cell type information c_n and random vector u_n, we create a new low-dimensional vector

$$z_n \sim \mathcal{N}(f_z^\mu(u_n, c_n), f_z^\sigma(u_n, c_n)), \tag{2.3}$$

where f_z^μ and f_z^σ are two functions parametrized by neural networks. Let s_n encode the dataset information. Given $l_\mu \in \mathbb{R}_+^K$ and $l_\nu \in \mathbb{R}_+^K$ specified per dataset as in [43], latent variable

$$l_n \sim \text{LogNormal}(l_\mu^{s_n}, l_\nu^{s_n}), \tag{2.4}$$

encodes a cell-specific scaling factor. As the prior are adjusted per dataset, our inference procedure will shrink the posteriors towards dataset specific values. This is particularly useful when aligning datasets with dramatically different library size values. Let $\theta \in \mathbb{R}_+^G$ encode a gene specific inverse dispersion parameter (inferred as in [43]). Conditional distribution $x_{ng} \mid z_n, l_n, c_n, s_n$ is conform to the one from the scVI model

$$w_{ng} \sim \text{Gamma}(f_w^g(z_n, s_n), \theta_g) \tag{2.5}$$

$$y_{ng} \sim \text{Poisson}(l_n w_{ng}) \tag{2.6}$$

$$h_{ng} \sim \text{Bernoulli}(f_h^g(z_n, s_n)) \tag{2.7}$$

$$x_{ng} = \begin{cases} y_{ng} & \text{if } h_{ng} = 0 \\ 0 & \text{otherwise} \end{cases}, \tag{2.8}$$

where f_w and f_h are functions parametrized by neural networks. f_w has a final softmax layer to represent normalized expected frequencies of gene expression as in [43]. Let us note that the resulting distribution for the counts is zero-inflated negative binomial. However, it is straightforward using our implementation to use a negative binomial or a Poisson noise model instead. In this model, annotation c_n can be either observed or unobserved following [155, 187], which is useful in our applications where some datasets would come partially labeled or unlabeled. Only the first part of the generative model, as separated above, differs from the original scVI formulation. This corresponds to the top part of the new representation of the graphical model in **Figure 2.1(b)**.

Approximate posterior inference for scANVI

We rely on collapsed variational inference, a standard approximate Bayesian inference procedure that consists in analytically integrating over some of the random variables [188] before optimizing the parameters. As we proved in [43], we can integrate the random variables $\{w_{ng}, y_{ng}, h_{ng}\}$ to simplify our model at the price of a looser though tractable lower bound ($x_{ng} \mid z_n, l_n, s_n$ is zero-inflated negative binomial). This procedure reduces the number of latent variable and avoids the need for estimating discrete random variables, which is a harder problem. We then use variational inference, neural networks and the stochastic gradients variational Bayes estimator [154] to perform efficient approximate inference over the latent variable $\{z_n, u_n, c_n, l_n\}$. We assume our variational distribution factorizes as:

$$q_\Phi(c_n, z_n, l_n, u_n \mid x_n, s_n) = q_\Phi(z_n \mid x_n) q_\Phi(c_n \mid z_n) q_\Phi(l_n \mid x_n) q_\Phi(u_n \mid c_n, z_n). \tag{2.9}$$

Following [155, 187], we derive two variational lower bounds: one $\mathcal{L}$ in the case of c_n observed for $p_\Theta(x_n, c_n \mid s_n)$ and a second $\mathcal{U}$ in the case of c_n non-observed for $p_\Theta(x_n \mid s_n)$ where Θ are all the parameters (neural networks and inverse-dispersion parameters). We optimize the sum *evidence lower bound* (ELBO) ELBO $= \mathcal{L} + \mathcal{U}$ over the neural networks parameters and the inverse-dispersion parameters (in a variational Bayesian inference fashion). Remarkably, the approximate posterior $q_\Phi(c_n \mid z_n)$ can be used as a classifier, assigning cells to cell types based on the location on the latent space.

We sample from the variational posterior using the reparametrization trick [154] as well as "mini-batches" from the dataset to compute unbiased estimate of the objective gradients' with respect to the parameters. We use Adam [189] as a first-order stochastic optimizer to update the model parameters.

Choice of hyperparameters

For all harmonization tasks in this paper, we consistently use the same set of hyperparameters. Each network has exactly 2 fully-connected layers, with 128 nodes each. The number of latent dimensions is 10, the same as other algorithms for benchmarking purposes (e.g., the number of canonical correlation vectors used in Seurat Alignment). The activation functions between two hidden layers are all ReLU. We use a standard link function to parametrize the distribution parameters (exponential, logarithmic or softmax). Weights for the first hidden layer are shared between f_w and f_h. We use Adam with $\eta = 0.001$ and $\epsilon = 0.01$. We use deterministic warmup [190] and batch normalization [191] in order to learn an expressive model. When we train scANVI, we therefore assume that the data come from a set of $C_{\text{observed}} + C_{\text{unobserved}}$ populations, each generated by a different distribution of z_n values. This set includes the C_{observed} populations for which annotated cells are available, and $C_{\text{unobserved}}$ population that accounts for cell types for which an annotation is not available to the algorithm.

Hierarchical classification of cells onto a cell type taxonomy

For hierarchical label propagation in scANVI, we propose a extension of the formerly presented model by modifying the variable c_n to be a tuple where each entry denotes the label at a given level of the hierarchy. Our approach is similar to previous work in robustness to noisy labels [192] and hierarchical multi-labels flavors of classification problems [193]. We extended scANVI to handle a two-level hierarchical structure for the cell types annotation though our approach can in principle be adapted to arbitrary depths. This can in principle be adapted to any arbitrary tree representation of cell types taxonomy, but is left for future work. In our setting, the taxonomy needs to be hard-coded and known *a priori*. We do not modify the generative model but only the structure of the variable c_n in the variational distribution. Notably, we formally pose:

$$c_n = (y_n, y_n^g) \in \{0, \cdots, C\} \times \{0, \cdots, C^g\}, \tag{2.10}$$

where C denotes the number of cell types and C^g the number of cell type groups. The parametrization of the full variational distribution $q(c \mid z) = q(y, y^g \mid z)$ must be further defined. For this, we notice that the prior taxonomy knowledge encapsulates whether the assignment $(y^g, y) = (i, j)$ is biologically possible (i.e cell type i is a sub-population of group cell type j). We encode this biological compatibility into a parent function $\pi : \{0, \cdots, K\} \to \{0, \cdots, K^g\}$ that maps a cell type to its parent in the hierarchy. We note for simplicity:

$$q(y_i, y_j^g \mid z) = q(y = i, y^g = j \mid z). \tag{2.11}$$

We then use two neural networks f and f_g (with softmax non-linearities) to map the latent space to the joint approximate posterior $q(y, y^g \mid z)$ with the following rules:

$$\begin{aligned} q(y_i, y_j^g \mid z) &= \begin{cases} f_i(z) & \text{if } \pi(i) = j, \\ 0 & \text{otherwise.} \end{cases} \\ q(y_j^g \mid z) &= f_j^g(z). \end{aligned} \tag{2.12}$$

Then, we can derive the marginal probability over finer cell types classes using the chain rule and Bayes rule:

$$q(y_i \mid z) = q(y_i \mid y_{\pi_i}, z) q(y_{\pi_i} \mid z) \tag{2.13}$$

$$= \frac{q(y_i, y_{\pi_i} \mid x)}{q(y_{\pi_j} \mid x)} q(y_{\pi_i} \mid z) \tag{2.14}$$

$$= \frac{q(y_i, y_{\pi_i} \mid x)}{\sum_{j \in c(\pi_i)} q(y_j, y_{\pi_j} \mid x)} q(y_{\pi_i} \mid z) \tag{2.15}$$

$$= \frac{f_i(z)}{\sum_{j \in c(\pi_i)} f_j(z)} f_{\pi_i}^g(z), \tag{2.16}$$

where $c(\pi_i)$ denotes the set of children of node i children.

Bayesian differential expression

Extending differential expression for scVI to the case of multiple batches For each gene g and pair of cells $(z_a,\ z_b)$ with observed gene expression (x_a, x_b) and dataset identifier (s_a, s_b), we can formulate two mutually exclusive hypotheses:

$$\mathcal{H}_1^g := \mathbb{E}_s f_w^g(z_a, s) > \mathbb{E}_s f_w^g(z_b, s) \quad \text{vs.} \quad \mathcal{H}_2^g := \mathbb{E}_s f_w^g(z_a, s) \leq \mathbb{E}_s f_w^g(z_b, s), \tag{2.17}$$

where the expectation $\mathbb{E}_s$ is taken with the empirical frequencies. Notably, we propose a hypothesis testing that do not calibrate the data to one batch but will find genes that are consistently differentially expressed. Evaluating which hypothesis is more probable amounts to evaluating a Bayes factor [194] (Bayesian generalization of the p-value) which is expressed as:

$$K = \log_e \frac{p(\mathcal{H}_1^g \mid x_a, x_b)}{p(\mathcal{H}_2^g \mid x_a, x_b)}. \tag{2.18}$$

The sign of K indicates which of $\mathcal{H}_1^g$ and $\mathcal{H}_2^g$ is more likely. Its magnitude is a significance level and throughout the paper, we consider a Bayes factor as strong evidence in favor of a hypothesis if $|K| > 3$ [195] (equivalent to an odds ratio of $exp(3) \approx 20$). Notably, each of the probabilities in the likelihood ratio for K can be written as:

$$p(\mathcal{H}_1^g \mid x_a, x_b) = \sum_s \iint_{z_a, z_b} \mathbb{1}_{f_w^g(z_a,s) \leq f_w^g(z_b,s)} p(s) dp(z_a \mid x_a) dp(z_b \mid x_b), \tag{2.19}$$

where $p(s)$ designated the relative abundance of cells in batch s and all of the measures are low-dimensional. Since we cannot in principle achieve efficient posterior sampling, the naive Monte Carlo estimator obtained by replacing the real posterior $p(z \mid x)$ by the variational posterior $q_\Phi(z \mid x)$ is biased. The resulting Bayes factors are therefore approximate though yield very competitive performance, as explained in the original publication of scVI [43]. Since we assume that the cells are independently distributed, we can average the probabilities for the hypotheses across a large set of randomly sampled cell pairs, one from each subpopulation. The Bayes factor from the averaged probability will provide an estimate of whether cells from one subpopulation tend to express g at a higher frequency.

Differential expression with scANVI In the case of scANVI, we need not rely on specific cells since labels are given during the training. We still use the generative model but with the following probability for $p(\mathcal{H}_1^g \mid c_a, c_b)$ where c_a (resp. c_b) is the first (resp. second) cell type of interest:

$$p(\mathcal{H}_1^g \mid c_a, c_b) = \sum_s \int \mathbb{1}_{f_w^g(z_a,s) \leq f_w^g(z_b,s)} p(s) dp(z_a \mid u_a, c_a) dp(z_b \mid u_b, c_b) dp(u_a) dp(u_b). \tag{2.20}$$

Notably, we draw here data from the prior distribution and not the posterior for given cells. As a consequence, these Bayes factors can be approximated in a unbiased fashion using a

naive Monte Carlo estimator. We noticed in the case of the real dataset that the aggregate posterior on u might not perfectly match the prior for rare cell types. Consequently, we replaced the prior by the aggregate posterior for all the analyses in this manuscript.

Datasets

We report an extensive list of datasets at **Appendix Table 2.1**. For all UMI based datasets we took the raw counts without any normalization as input to scVI.

Gene Selection A common practice in data harmonization is to perform gene selection prior to harmonization. This assumption is critical when the number of genes that can be taken into account by the algorithm is small and potentially biological signal could be lost. scVI is however designed for large datasets which do not fall into the high-dimensional statistics data regime [43]. Remarkably, there is no need for crude gene filtering as part of our pipeline and we adopt it as part of this publication only for concerns of fairness in benchmarking. For real datasets, we calculated the dispersion (variance to mean ratio) for all genes using Seurat in each dataset and selected $g = 1,000$ genes with the highest dispersion from each. The performance of scVI is not as affected by gene set and we use the same gene selection scheme as in [33] to ensure fairness in our comparison. We then took the union of these gene list as input to Seurat Alignment, MNN and scANVI. One exception is the differential expression study for which we kept the gene set ($g = 3,346$) to have it match the bulk reference as in [43].

Cell type labeling for the Tabula Muris Dataset For the Tabula Muris dataset, cell types are defined by first reducing the dimensions of the data by principal component analysis and then performing nearest-neighbor-graph-based clustering. The labels for Smart-Seq2 and 10x data are derived independently. All cells in both dataset are labeled, but there is also a possibility that they are mislabelled since the labels are computationally derived. Since cells used in Smart-Seq2 are first FACS sorted into each plate, some cell types might have been lost during the sorting process, resulting in incomplete overlap in cell types between the two datasets.

Hierarchical cell type labeling for the mouse nervous system dataset The multi-level labels are generated through an iterative process that is described in detail in the original publication [48]. The clustering was performed with strict quality filters, takes into account anatomical information and were validated at different levels using existing scRNAseq dataset, osmFISH, RNAscope and others. The cell types taxonomy is derived differently for each level and the details can be found in the original publication. Cell type clusters were obtained by Louvain clustering on a multiscale k-nearest neighbors graph and DBSCAN. The first level separates neurons and non-neuronal cells. The second level separates peripheral neuronal system from central neuronal system. The third layer separates

anterior posterior domain, and the fourth layer is split by excitatory versus inhibitory neurotransmitter. At this level, all cells are divided into 39 subsets, each corresponding to a coarse cell type definition. Then, within each subset the authors defined (N=28) enriched genes and used linkage (correlation distance and Ward method) to construct the dendrogram.

Normalization of CITE-seq data Since we did not explicitly model the CITE-seq data in scVI or scANVI, we normalized it by fitting a Gaussian mixture model to each individual protein with two components. We then transformed each individual protein count as $x \mapsto (x - \frac{\mu_1+\mu_2}{2})_+$ where μ_1 and μ_2 designate the mean of the mixtures and $._+$ is the positive part of a real number.

Normalization of SmartSeq2 data For the MarrowMT-ss2 dataset, we normalized the read counts per gene by relative transcript length (average transcript lengths of a gene divided by average gene length over all genes), and subsequently took the integer part of the normalized count. This is different from standard normalization procedures in that we do not normalize by cell size because cell size normalization can be performed by scVI. And we only keep the integer part of the counts, due to the distributional assumptions made by scVI. The scVI model can to be extended to fit data with amplification bias, however we have not done so for this paper and thus have to perform this normalization heuristic.

Simulation of continuous gene expression using SymSim First we simulated the true expression matrix for a tree with 5 cell types using the function SimulateTrueCounts. Instead of sampling cells only from the leaf populations, we uniformly sample cells along all branches by using the parameter evf_type="continuous". We then added noise to the data with the function True2ObservedCounts with the parameters

```
protocol="nonUMI", alpha_mean = 0.1, alpha_sd = 0.05,
rate_2PCR = 0.7, nPCR1 = 16,depth_mean=1e5, depth_sd=3e3
```

Simulation for DE benchmark using SymSim First we simulated the true expression matrix for 20,000 cells from 5 cell types using the function SimulateTrueCounts. We then randomly split the cells into two batches. We then added noise to the data the function True2ObservedCounts with the parameters

```
Batch 1: protocol="UMI", alpha_mean=0.03, alpha_sd=0.009, gene_len=gene_len,
depth_mean=5e5, depth_sd=1.5e4

Batch 2: protocol="UMI", alpha_mean=0.1, alpha_sd=0.03, gene_len=gene_len,
depth_mean=1e6, depth_sd=1.5e5
```

Algorithms for benchmarking

Seurat Alignment We applied the Seurat Alignment procedure from the R package Seurat V2. The number of canonical correlation vectors is 10 for all the datasets, which is also identical to the number of latent dimensions used for scVI and scANVI.

Seurat PCA We applied the Seurat PCA procedure from the R package Seurat V2. This method is a simple PCA based after normalization by Seurat. Seurat PCA is used to obtain the individual dataset latent space to evaluate the k-nearest neighbors purity for all non-scVI based methods. The number of principal components is 10.

Matching Mutual Nearest Neighbors We used the mnnCorrect function from `https://rdrr.io/bioc/scran/man/mnnCorrect.html` with default parameters. In order to compare with other methods, we applied a PCA with 10 principal components on the output of the batch-corrected gene expression matrix.

scmap We applied the scmap-cluster procedure from the R package scmap. As the scmap manuscript insists heavily on why the M3Drop [196] gene filtering procedure is crucial to overcome batch effects and yield accurate mapping, we let scmap choose its default number of genes ($g = 500$) with this method.

ComBat We used the R package sva with default parameters.

UMAP We used the umap class from the UMAP package with a default parameters and spread=2.

DBSCAN We used the DBSCAN algorithm from the Python package from the python package scikit-learn V0.19.1 and we searched for an optimal hyperparameter combination by a grid search over `eps` and `min_samples` from the range of $0.1 - 2$ and $5 - 100$ respectively. Although some combinations of parameters yield more than one clusters, the smaller clusters comprise of less than 1% of the data. We then evaluated DBSCAN with `eps=1.23, min\_samples=10` and default values for all other hyper-parameters.

PhenoGraph We used the phenograph.cluster function from the Python package PhenoGraph 1.5.2 downloaded from `https://github.com/jacoblevine/PhenoGraph` with its default parameters.

CORAL We used the implementation from `https://github.com/jindongwang/transferlearning/tree/master/code/traditional/CORAL`.

MAGAN We used the implementation from `https://github.com/KrishnaswamyLab/MAGAN`.

Harmony We used the implementation from `https://github.com/immunogenomics/harmony`.

Scanorama We used the implementation from `https://github.com/brianhie/scanorama`.

Evaluations metrics

Entropy of batch mixing Fix a similarity matrix for the cells and take U to be a uniform random variable on the population of cells. Take B_U the empirical frequencies for the 50 nearest neighbors of cell U being a in batch b. Report the entropy of this categorical variable and average over $T = 100$ values of U.

k-nearest neighbors purity Compute two similarity matrices for cells from the first batch, one from the latent space obtained with only cells from the first batch and the other from the latent space obtained using both batches of cells. We always rely on the euclidean distance on the latent space. Take the average ratio of the intersection of the k-nearest neighbors graph from each similarity matrix over their union. Compute the same statistic for cells from the other batch and report the average of the two.

Weighted and unweighted accuracy We evaluate the accuracy of cell type classification algorithms by comparing the predictions to previously published labels. The unweighted accuracy is the percentage of cells that have the correct label. The weighted accuracy corresponds to first calculating accuracy for each cell type, and then averaging it across cell types. The weighted accuracy assigns the same weight to each cell type and thus weighs correct prediction of rare cell types more heavily than the unweighted accuracy. We report the weighted accuracy throughout this manuscript.

Maximum Posterior Probability We evaluate the performance of the scANVI classifier at transferring labels from an annotated dataset to an unannotated dataset by looking at the maximum posterior probability for the observed classes. By default scANVI classifier sets the number of classes to the same number of cell types in the merged dataset. In the case of N observed labels from the annotated dataset and one unannotated dataset (thus the cell type label is "Unlabeled") scANVI assumes $N + 1$ classes. For each cell, scANVI assigns a posterior probability for each of the $N + 1$ classes. The maximum posterior probability for the observed classes is the highest probability of a cell being assigned to one of the N observed classes.

Signature for sub-division of T cells in human PBMCs

Gene sets For ranking the cells, we used both positive and negative sets of genes:

- **CD4 Regulatory**: *GITR+ CTLA4+ FOXP3+ CD25+ S100A4- CD45- CD8B-*
- **CD4 Naive**: *CCR7+ CD4+ S100A4- CD45- FOXP3- IL2RA- CD69-*
- **CD4 Memory**: *S100A4+ CD25- FOXP3- GITR- CCR7-*
- **CD8 Naive**: *CD8B+ CCR7+ CD4-*

Signature calculus To compute the signature of a cell, we followed the normalization procedure from [151] which consists in dividing by total numbers of UMIs, applying a entry-wise transformation $x \mapsto \log(1 + 10^4 x)$ and z-score normalization for each gene. Then, we aggregated over the genes of interest for each cell by applying the sign from the gene-set and averaging.

2.4 Discussion

In this study, we demonstrated that scVI provides a principled approach to harmonization of scRNA-seq data through joint probabilistic representation of multiple dataset, while accounting for technical hurdles such as variable library size and limited sensitivity. We have demonstrated that scVI compares favorably to other methods in its accuracy and that it scales well, not only in terms of the number of cells (as in [43]) but also the number of input datasets (as opposed to other methods that work in a pairwise fashion and therefore scale quadratically with dataset size [150]). We have also shown that the harmonization step of scVI provides an effective baseline for automated transfer of cell type labels, from annotated datasets to new ones.

While the performance of scVI in the annotation problem compares favorably to other algorithms, it does not make use of any existing cell state annotations during model training, but rather after the latent space has been learned. To make better use of these annotations (which may be available for only some of the input datasets or only some cells within a dataset), we developed scANVI, a semi-supervised variant of scVI. While the latent space of scVI is defined by a Gaussian vector with diagonal unit variance, scANVI uses a mixture model, which enables it to directly represent the different cell states (each corresponding to a mixture component; see **Method**) and provide a posterior probability of associating each cell with each label. We have demonstrated that similar to scVI, scANVI is capable of harmonizing datasets effectively. In addition, scANVI provides a way to address a number of variants of the annotation problem. Here, we have first shown that it performs well in the most prevalent application of transferring labels from a reference dataset to an unannotated one. We then demonstrated that scANVI can be used in the context of a single unannotated dataset, where high confidence ("seed") labels are first inferred for a few cells (using marker

genes) and then propagated to the remaining cells. Finally, we have shown that scANVI is especially useful in the challenging case where the differences between cell states are too subtle to be captured clearly by a transcriptome-wide similarity measure, as well as in the case where the labels are organized in a hierarchy.

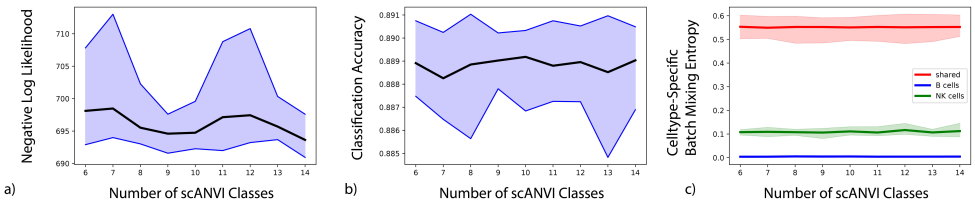


Figure 2.25: The effect of the choice of number of classes on the scANVI model

The effect of the choice of number of classes on the scANVI model likelihood (a), classification accuracy (b) and entropy of batch mixing (c). We trained scANVI using PBMC8K as the labelled dataset, and varied the number of classes in scANVI from 6(true number of labelled cell types) to 14. The thicker line show the mean of 9 replicates, while the colored shading show the 95% confidence interval. We used a subsampled PBMC8K-CITE dataset, where NK cells are removed from the PBMC8K dataset and B cells are removed from the PBMC-CITE dataset. As we expect, the two unique dataset have low mixing in (c) while the other cell types have high mixing. Although there is no labelled B cells, scANVI does not cluster B cells from the PBMC8K dataset with other cell types in PBMC-CITE. The three metrics we use to evaluate scANVI performance are minimally affected by the increase in the number of classes.

Notably, although scANVI achieves high accuracy when transferring labels from one dataset to another, it was not designed to automatically identify previously unobserved labels. Indeed, in **Figure 2.25** we demonstrate that increasing the number of labels in the model (C) to values beyond the number of observed labels does not alter the results much. Nevertheless, we observed that unannotated cell populations that have an unobserved label are associated with low levels of mixing between the input datasets. We therefore advocate that clusters from an unannotated dataset that do not mix well should be inspected closely and, if appropriate, should be manually assigned with a new label.

One concern in applying methods based on neural networks [197, 198, 199, 200, 201] in single-cell genomics and other domains is the robustness to hyperparameters choices [202]. This concern has been addressed to some extent by recent progress in the field, proposing search algorithms based on held-out log-likelihood maximization [200]. In this manuscript, we used an alternative approach that is more conducive for direct and easy application of our methods — namely we fix the hyperparameters and achieve state-of-the-art results on a substantial number of datasets and case studies.

An important distinguishing feature of both scVI and scANVI is that they rely on a fully-probabilistic model, thus providing a way to directly propagate uncertainties to any downstream analysis. While we have demonstrated this for differential expression analysis and cell type annotation, this can be incorporated to other tasks, such as differential abundance of sub-populations in case-control studies, correlation between genes and more. We therefore expect scVI, scANVI and similar tools to be of much interest as the field moves toward the goal of increasing reproducibility and consistency between studies and converging on to a common ontology of cell types. In particular, we expect scANVI to be especially useful for transferring labels while taking into account the uncertainty, or in the case of a more complex label structure such as hierarchical cell types. Finally, as recent preprints propose proof of concepts for integrating single cell data across different data modalities such as Single molecule fluorescent in situ hybridization (smFISH), RNA-seq, ATAC-seq and DNA methylation [149, 34], further work can utilize probabilistic graphical models that quantify measurement uncertainties in each assay, as well as the uncertainties of transferring information between modalities (e.g., predicting unmeasured gene expression in smFISH data as in [203]).

2.5 Acknowledgement

This chapter is written in collaboration with Romain Lopez, Edouard Mehlman, Jeffrey Regier and is supervised by Nir Yosef and Michael I Jordan and was previously published in Molecular Systmes Biology. RL, EM, JR and NY conceived the statistical model. EM developed the software. CX, RL and EM applied the software to real data analysis. CX, RL, JR, NY, and MIJ wrote the manuscript. NY and MIJ supervised the work.

CX, RL, and NY were supported by grant U19 AI090023 from NIH-NIAID and U19

MH114821 NIMH. We thank Maxime Langevin, Yining Liu and Jules Samaran for helpful discussions and early work on the scVI codebase as well as Allon Wagner and Chao Wang for their help on the choice for high specificity genes in the T cell study. Additionally, we would like to thank Adam Gayoso and Galen Xing for discussions around the stability of the algorithm as well as the API within the new scvi-tools package.

Chapter 3

Integrated single cell analysis of blood and cerebrospinal fluid leukocytes in multiple sclerosis

3.1 Introduction

Single-cell transcriptomics is a transformative and rapidly evolving technology that has mostly been used to re-define the heterogeneity of complex tissues from healthy rodents or humans [25, 204, 205]. Diseased tissues have also been analyzed with single-cell technologies [32, 206]. Proponents of the technology posit that insights from single-cell transcriptomics are likely to enable precision medicine in the not-too-distant future [207, 208]. However, outside of the field of cancer, few studies have used the technology to compare tissue samples from disease-affected vs. control donors in a clinically relevant setting [49]. This leaves many methodological and conceptual questions unexplored. In this chapter we use the data harmonization method scVI tested in Chapter 2 to conduct a tissue-specific case-control study to understand the cellular mechanisms of Multiple Sclerosis. All the sequencing procedures and experiments outside of scRNAseq are conducted by my collaborators and the results are only shown to confirm the scRNAseq analysis results.The code for reproducing the results in this chapter that are unique contributions from me has been deposited at `https://github.com/chenlingantelope/MSscRNAseq2019.git`

Cerebrospinal fluid (CSF) is a clear liquid that envelops and protects the central nervous system (CNS) [209] and forms a unique local immune compartment [210]. Under healthy conditions, the non-cellular fraction of CSF is mostly an ultra-filtrate of serum [211]. In contrast, CSF cells that derive exclusively from the hematopoietic lineage exhibit a tightly controlled cellular composition considerably different from the blood [212, 213], but the underlying mechanisms remain largely unexplored [214]. Clinically, CSF facilitates the diagnosis of inflammatory and degenerative diseases of the CNS. However, the concentration of

CSF cells is approximately 1,000-fold lower than in blood and limited volumes can be safely sampled in every patient. Technical approaches must therefore be compatible with low input and a comprehensive transcriptional characterization of single CSF cells under homeostatic conditions and in inflammatory CNS diseases is unavailable [215, 216].

Multiple sclerosis (MS) is a paradigmatic chronic inflammatory, demyelinating disorder of the central nervous system (CNS) causing substantial disability [217]. This complex disease is likely of autoimmune origin, but many questions remain unanswered despite a vast amount of available literature. In fact, evidence supports the involvement of both T cells and B cells in MS, but the relative contribution of each cell type to disease aetiology is unknown. On the one hand, production of immunoglobulins and expansion of B lineage cells [218, 212] occurs in the CSF with evidence of antigen-driven maturation [219, 220] and B cell depleting therapies are effective in MS [221]. On the other hand, T cells are abundant in MS lesions [222, 223] and T cells are affected by many established MS treatments and induce an MS-like condition named experimental autoimmune encephalomyelitis (EAE) in rodents [224]. Whether a pathological interaction of T cell and B cell subsets may occur locally in human CSF remains unknown.

Here, we apply single-cell transcriptomics to blood and CSF cells from patients with MS and controls and validate key findings. First, we identify a compartment-specific composition and transcriptome including an unknown enrichment of myeloid dendritic cells in the CSF. Second, we find that MS mainly affects the cellular composition of the CSF, but the transcriptional phenotype of blood cells. We also identify an expansion of $CD4^+$ T cells with a cytotoxic phenotype and late-stage B lineage cells in the CSF in MS. Third, we newly introduce cell set enrichment analysis (CSEA) to identify cluster-independent cellular changes and thereby observe an expansion of B cell-helping T follicular helper (TFH) cells. In a reverse-translational approach, we fourth confirm that such TFH cells promote CNS autoimmunity and local B cell infiltration in two distinct animal models of MS. We thus demonstrate how an unbiased approach aids our understanding of a unique human immune compartment and identifies mechanisms locally driving CNS disease.

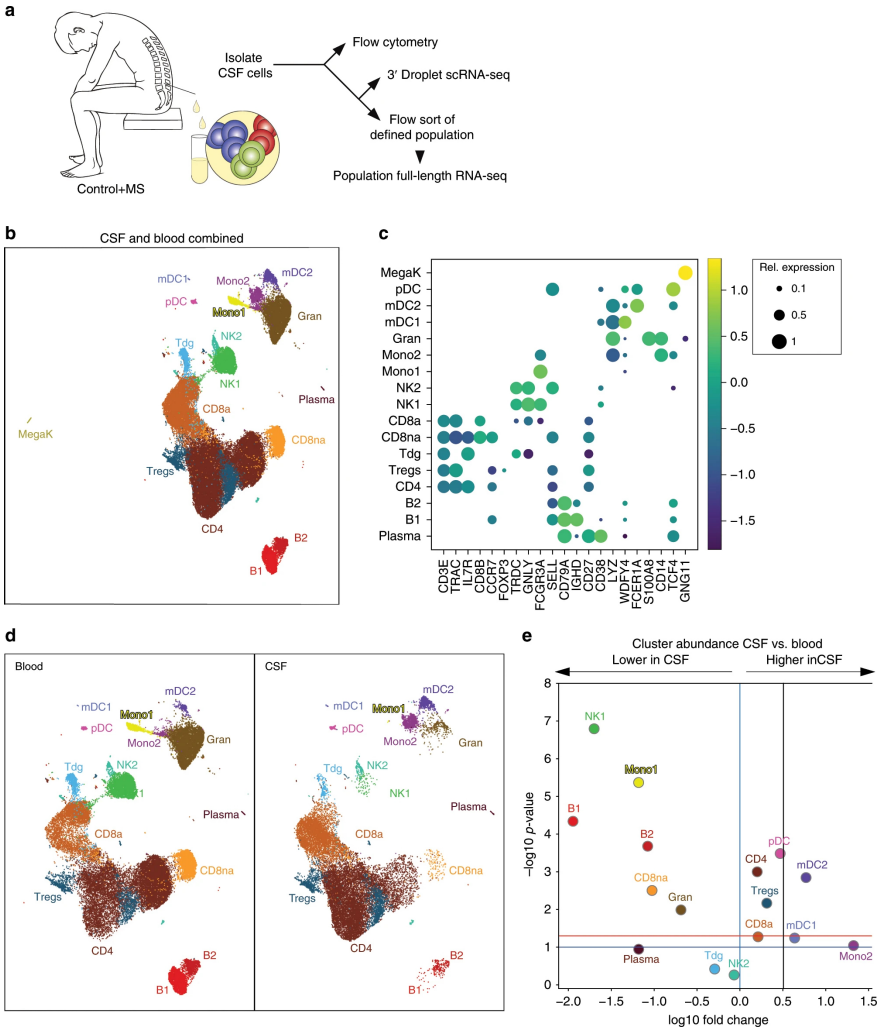
3.2 Results

Single cell transcriptomics reconstructs cell types in cerebrospinal fluid and blood

We first aimed to identify the compartment-specific composition and expression of CSF cells compared to blood using an unbiased approach (**Figure3.1(a)**). We recruited patients with idiopathic intracranial hypertension (IIH) as controls and treatment-naïve patients with clinically isolated syndrome (CIS) or relapsing-remitting MS (together termed MS, **Methods**)

donating blood and CSF. Both cohorts were well matched and CSF parameters exhibited known MS-associated changes. Negativity for oligoclonal bands was 18% in accordance with early MS [225]. Using microfluidics-based single cell RNA-sequencing (scRNA-seq) we obtained in total $42,969$ blood single cell transcriptomes (5 control vs. 5 MS donors) and $22,357$ corresponding CSF single cell transcriptomes (4 control vs. 4 MS donors). Genes detected per donor were 934.4 ± 379.1 SEM in PBMCs and $1,021.4 \pm 374.0$ SEM in CSF. After filtering and normalization, we performed multi-step clustering of the merged $65,326$ blood/CSF cell dataset. We thereby classified 61,051 single cells into 17 final cell clusters (**Figure3.1(b)**) after removal of red blood cells and low quality cell clusters (**Methods**).

Figure 3.1: Single cell transcriptomics reconstructs cell types in cerebrospinal fluid and blood



Single-cell transcriptomics reconstructs the compartment-specific leukocyte composition of CSF and blood.

a. Schematic of the study design (**Methods**). *b* Uniform Manifold Approximation and Projection (UMAP) plot representing 17 color-coded cell clusters identified in merged single cell transcriptomes of blood (42,969) and CSF (22,357) cells from control (n = 4) and multiple sclerosis (MS) (n = 4) patients (**Methods**). Cluster names were manually assigned. *c* Dotplot depicting selected marker genes in cell clusters. Dot size encodes percentage of cells expressing the gene, color encodes the average per cell gene expression level. *d* UMAP plots comparing blood (left) and CSF (right) cell clustering. MegaK cluster is disregarded for higher resolution. *e* Volcano plot depicting differences of cluster abundance in CSF compared to blood plotting fold change (log10) against p-value (-log10) based on beta-binomial regression (**Methods**). Horizontal line indicates significance threshold.

Based on marker gene expression (**Figure3.1(c)**,selected protein names in non-italic), we identified $\alpha\beta$ T cells (CD3E, LCK, TRAC, TRAJ16) subsetting into CD4$^+$ T cells (*IL7R*, *CD4*), activated CD8+ T cells (CD8a, *CD8B*, *CCL5*), non-activated CD8+ T cells (CD8na; *CD8B*, *CCR7*), regulatory T cells (*FOXP3*, *CTLA4*) and a small cluster of $\gamma\delta$ T cells (TRDC). Two NK cell clusters (*GNLY*, *NKG7*) most likely represented the more cytotoxic and mature CD56$^{\text{dim}}$ (NK1; *FCGR3A*/CD16, *PRF1*) and more naive CD56bright (NK2; S*ELL*/CD62L, XCL1) subsets. Three B lineage clusters (*CD74*, *CD79A*, *IGH* gene family) corresponded to naive B cells (B1; *CD37*, *IGHD*), activated B cells (B2; *CD27*, *IGHM*), and plasma blasts (plasma; *IGHG*, *CD38*, *TNFRSF17*/CD269; negative for *MS4A1*/CD20, *SDC1*/CD138). Myeloid lineage cells (LYZ) separated into myeloid dendritic cells (mDC) type 1 (mDC1; *WDFY4*, *XCR1*,*BATF3*), mDC type 2 (mDC2; *FCER1A*, *CD1C*, *CLEC10A*), and granulocytes (granulo; *S100A8*, *S100A9*). Two additional monocyte cell clusters were mostly blood-derived (Mono1; *FCGR3A*/CD16) or CSF-derived (Mono2; *CD14*). Additional clusters represented plasmacytoid dendritic cells (pDC; *TCF4*/E2-2, *TNFRSF21*/DR6) and megakaryocytes (MegaK; *GNG11*, *CLU*). Microfluidics-based scRNA-seq thus successfully reconstructed leukocyte lineages from CSF and blood.

Cerebrospinal fluid leukocytes exhibit a compartment-specific composition and transcriptome

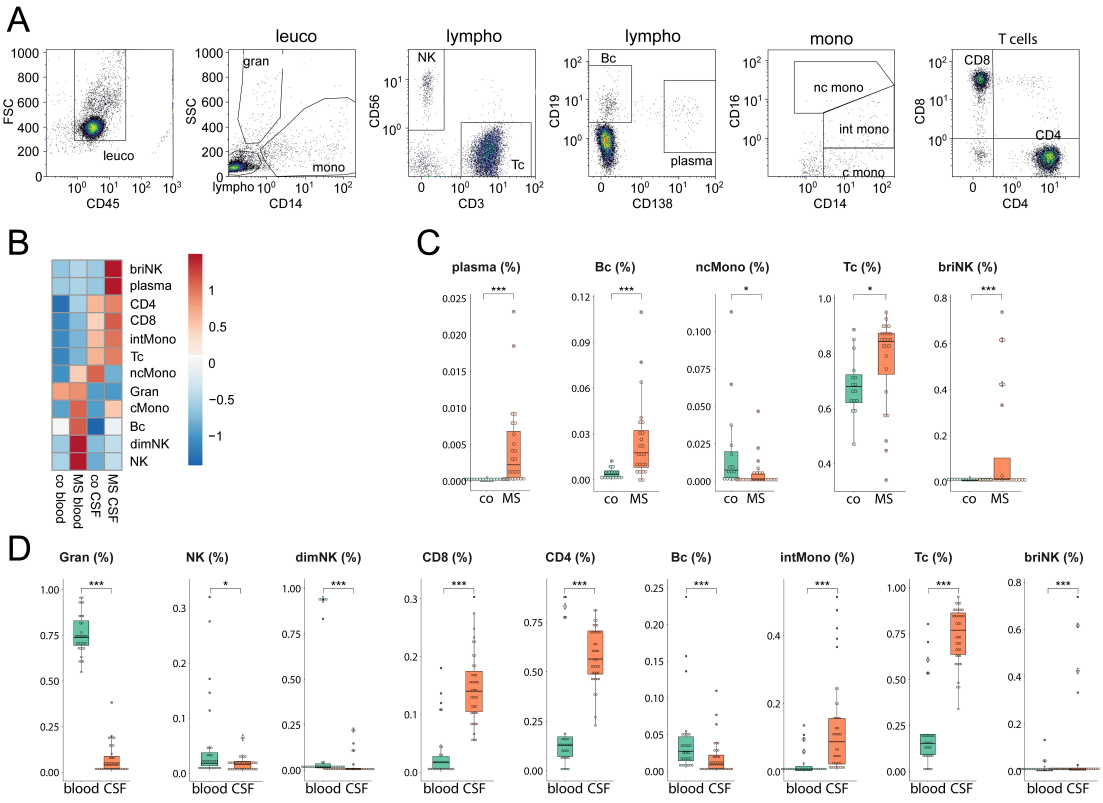
CSF cells have not been characterized with unbiased approaches. We therefore next analyzed the compartment-specific cell type composition identified by unbiased scRNA-seq in CSF compared to blood. As expected for CSF [212, 226], non-hematopoietic cells (e.g. neurons, glia, ependymal cells), megakaryocytes, granulocytes, and RBC (removed from final clustering) were absent or strongly reduced compared to blood (**Figure3.1de**). We also found CD56$^{\text{dim}}$ NK1 cells reduced among CSF cells, while the NK2 cluster was not different (**Figure3.1de**). Both the mDC1 and mDC2 clusters had a significantly higher proportion in CSF than in blood (**Figure3.1de**). Notably, mDC1 cells expressed markers indicating cross-presenting capacity (*XCR1*, *WDFY4*, [227]; **Figure3.1(c)**). Among T cells, total CD4 cells and Tregs were more abundant in the CSF, while CD8 T cell clusters were not different (**Figure3.1de**). Flow cytometry confirmed this unique composition of CSF leukocytes **Figure3.1abc**). Cell proportions in CSF and blood did not correlate by either scRNA-seq or flow cytometry (data not shown) supporting an independent regulation of their cell composition. In summary, we confirmed a highly compartment-specific composition of CSF cells and identified a previously unknown enrichment of mDC1 and Tregs in the CSF.

We also found a CSF-specific pattern of myeloid lineage cells. The Mono2 cluster was almost exclusively CSF-derived (**Figure3.1(d)**) and canonical markers indicated an intermediate *CD14*+*FCGR3A*/CD16int phenotype (**Figure3.1(c)**) as described for CSF [228]. It also expressed a unique transcriptional signature including genes previously identified in classical

(*CD9*, *CD163*, *EGR1*, *BTG2*) and in non-classical (*C1QA*, *C1QB*, *MAF*, *CSF1R*/CD115) monocytes [229]. Notably, the CSF-derived Mono2 cluster also expressed (Suppl. Tab. 4) markers of perivascular macrophages (*LYVE1*; [230]), microglia (*TREM2*, *TMEM119*, *GPR34*; [231]) and CNS border associated macrophages (*STAB1*, *CH25H*; [232, 233]) previously identified in rodents. In a systematic comparison (**Methods**), the Mono2 gene signatures resembled homeostatic microglia described previously [234]. We thus identified a distinct phenotype of CSF monocytes.

We next aimed to identify further compartment-specific gene expression signatures on a per cluster level (Suppl. Tab 5). We focused on genes identified independently as differentially expressed (DE) by two methods (Mann-Whitney U test, edgeR [235]) and supported by Bayesian model comparison in scVI ([43], **Methods**). Due to the stringency of this approach, most of such 'triple-consistent' genes were DE in CSF vs. blood cells in only one (18.9% of all expressed genes) or two (5.1%) clusters , although measures of differential expression were positively correlated especially between related clusters **Figure3.3** indicating co-regulated gene modules in related cell types.

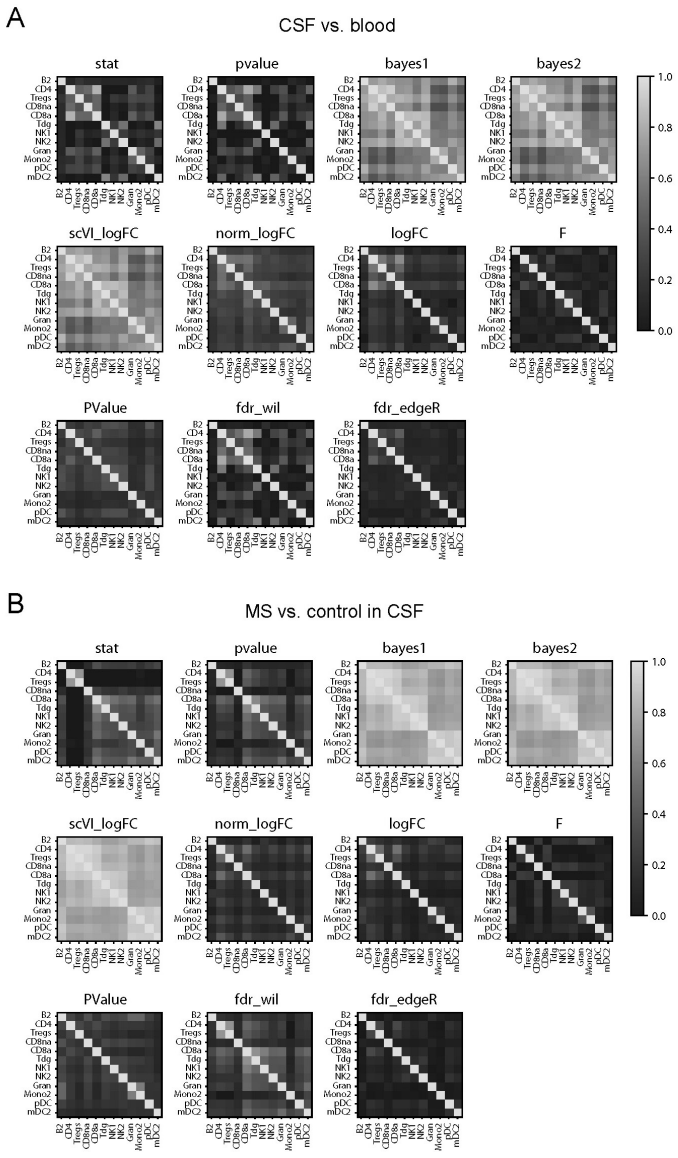
Figure 3.2: Flow Cytometry Validation



Flow cytometry characterization of all CSF and blood samples

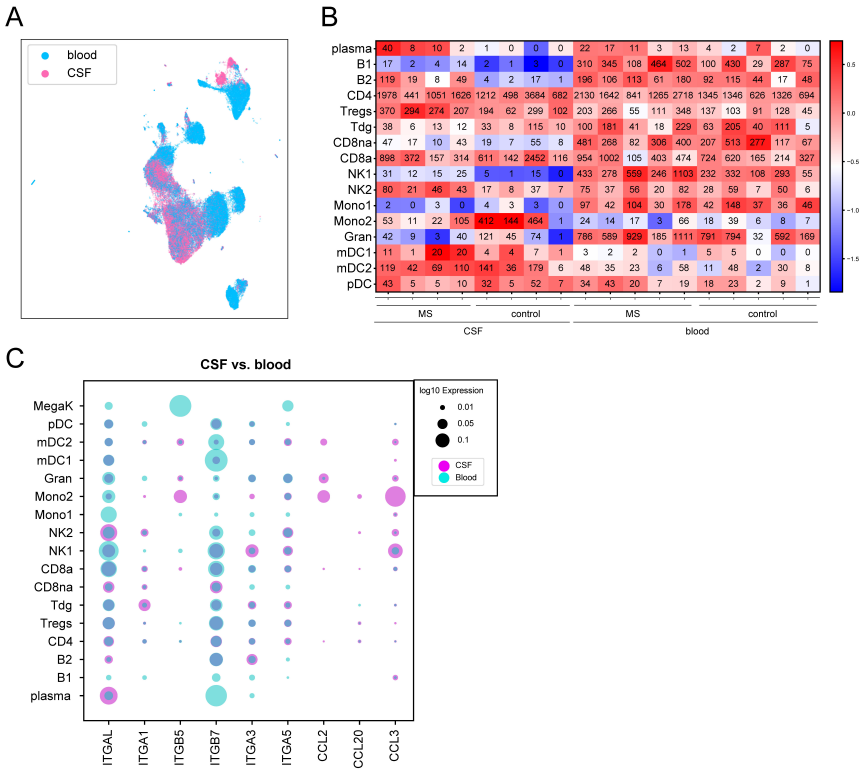
(*a*) Representative gating strategy for identifying and quantifying cell types by flow cytometry in the CSF. Population names are indicated next to the respective gates. (*b*) Heatmap depicting the average proportion of cells in each population in control (co, n = 22) and multiple sclerosis (MS, n = 26) patients as quantified in all samples by flow cytometry. Heatmap color is scaled in each row to row average with color indicating higher (red) and lower (blue) than average. (*c*) CSF flow cytometry data are depicted as dot-boxplots if significantly (t-test statistics) different between control and MS samples. Comparisons not depicted are not significantly different. Please note that none of the analyzed flow cytometry parameters was different between co and MS in blood. (*d*) Flow cytometry data are depicted as dot-boxplots if significantly (t-test statistics) different between blood and CSF. * $P < 0.05$, ** $P < 0.01$, *** $P < 0.005$. Bc B cells, CD4 $CD4^+$ T cells, CD8 CD8+ T cells, dimNK / briNK $CD56^{dim}$ / CD56bright natural killer cells, Bc B cells, plasma plasma cells, class mono / int mono / nc mono classical / intermediate / non-classical monocytes, granulos granulocytes.

Figure 3.3: Differential Gene Expression Correlations



Correlation between different measures of differential expression.
(*a*) Matrix of Spearman rank correlation r of the different measures of differential expression of individual genes within clusters in the CSF vs. blood comparison. For example, the Bayes1 factor of all genes in the CD4 cluster is more closely correlated (i.e. yellow) with all genes in the CD8na cluster than with all genes in the Gran cluster. (*b*) Matrix of Spearman rank correlation r of the different measures of differential expression of individual genes within clusters in the MS vs. control comparison within CSF.

Figure 3.4: Inter-individual donor heterogeneity of cell cluster abundance



Inter-individual donor heterogeneity of cell cluster abundance.
(*a*) UMAP plot depicting all cells shown in Fig. 1A color-coded by tissue of origin after 2nd level clustering with blue indicating blood cells and purple indicating CSF cells. (*b*) Heatmap depicting cell numbers in each cell cluster in each donor sorted by tissue of origin and by disease-status. Numbers in the heatmap represent cell numbers in each cluster after 2nd level clustering. Color code indicates the relative abundance of cell types per donor, compared to the row average. (*c*) Dotplot depicting selected genes differentially expressed in at least one cluster of CSF cells compared to blood. Purple dot size encodes the average expression in CSF and turquoise dot size encodes the average expression in blood. Dots are partially transparent thus overlap is dark blue. Purple edge around dark blue circle indicates higher expression in CSF, while turquoise edge around dark blue circle indicates lower expression in CSF compared to blood.

Genes induced in multiple (i.e. > 3) CSF clusters included *FGF9*, previously implicated in inflammatory CNS tissue damage [236] and Metallothionein E, potentially involved in CSF metal ion homeostasis [237]. Cell cycle (e.g. *CCNC*/Cyclin-C) genes were induced in CD4$^+$ T cells in line with their activated phenotype in CSF [238, 239]. Genes induced in CD4$^+$ T cells in the CSF were also related to lipid antigen recognition (*CD1E*), interaction with antigen-presenting cells (*CD81*, *CD83*, *CD84*, *CD209*) and adhesion and migration (*CD99*). In fact, CSF T cells expressed a specific pattern of chemokine and integrin transcripts including an induction of *CXCL16* and *CXCR5* and downregulation of *ITGAL*/VLA4 in CSF CD4$^+$ T

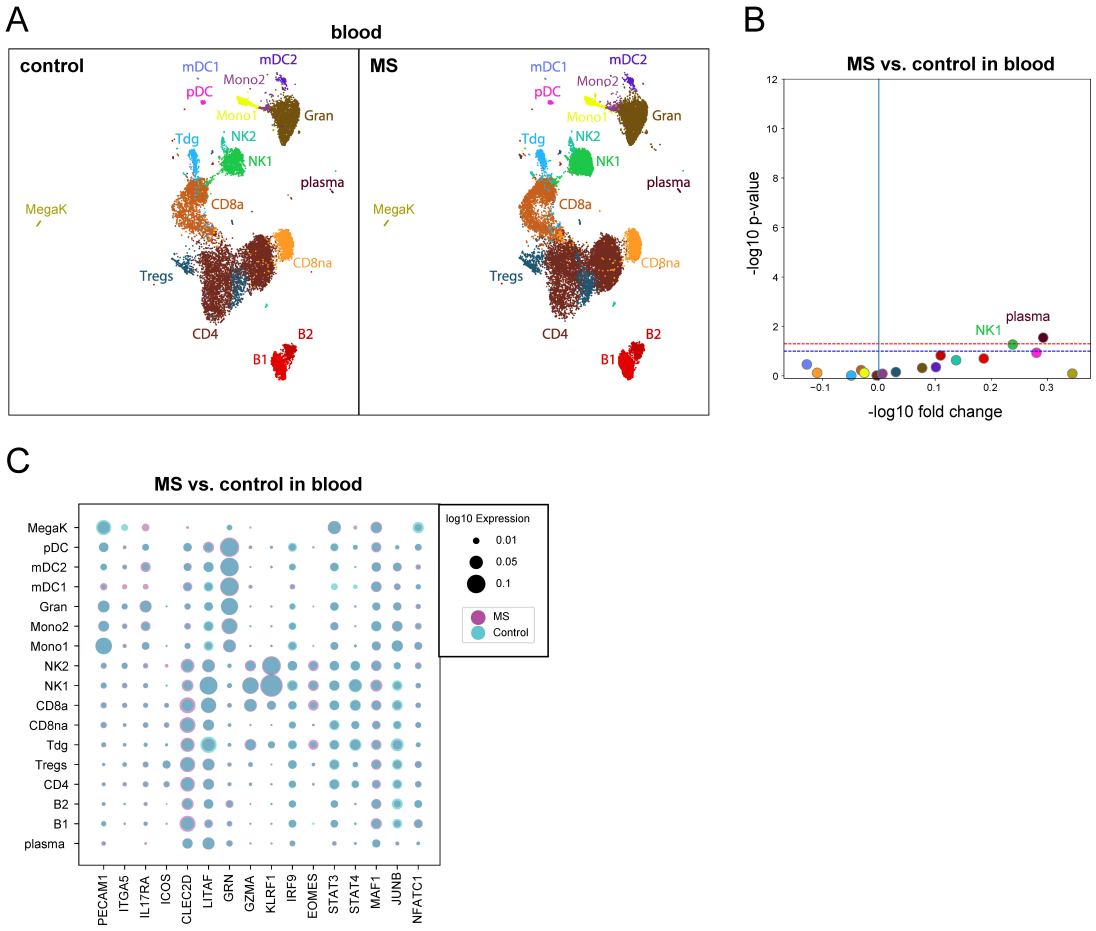
cells and of *ITGB7* in myeloid cells (**Figure3.4**). Genes consistently downregulated in CSF T were associated with naïve cell state (*SELL*/CD62L), cytokine responses (*IL2RG*/common γ chain). Interestingly, CD48 previously associated with CSF translocation of bacteria [240] was upregulated in CSF T cells. In accordance, GSEA showed enrichment of pathogen response pathways in CSF induced genes (e.g. KEGG pathways hsa05169, hsa05168). B cell clusters (B1, B2, plasma) showed no transcriptional changes between compartments. Genes associated with memory formation (*ID3,CCR2*) were induced in the CD8a cluster. Single cell transcriptomics thus identified a location-specific transcriptional phenotype and trafficking molecule expression of CSF leukocytes (**Supplementary Table 3.1**).

Multiple sclerosis preferentially alters transcription of blood and composition of CSF cells

Next, we analysed our dataset for MS-associated changes. Blood cells exhibited no significant differences in composition in MS compared to control (**Figure3.5ab**) as confirmed by flow cytometry (**Figure3.2**). In contrast, blood cells exhibited diverse 'triple-consistent' (see above and **Methods**) transcriptional changes including an induction of activation markers (*ICOS*), specific cytokine receptors (*IL17RA*), and trafficking molecules (*PECAM1*/CD31, *ITGA5*/α5 integrin) in T cells (**Figure3.5(c)**).

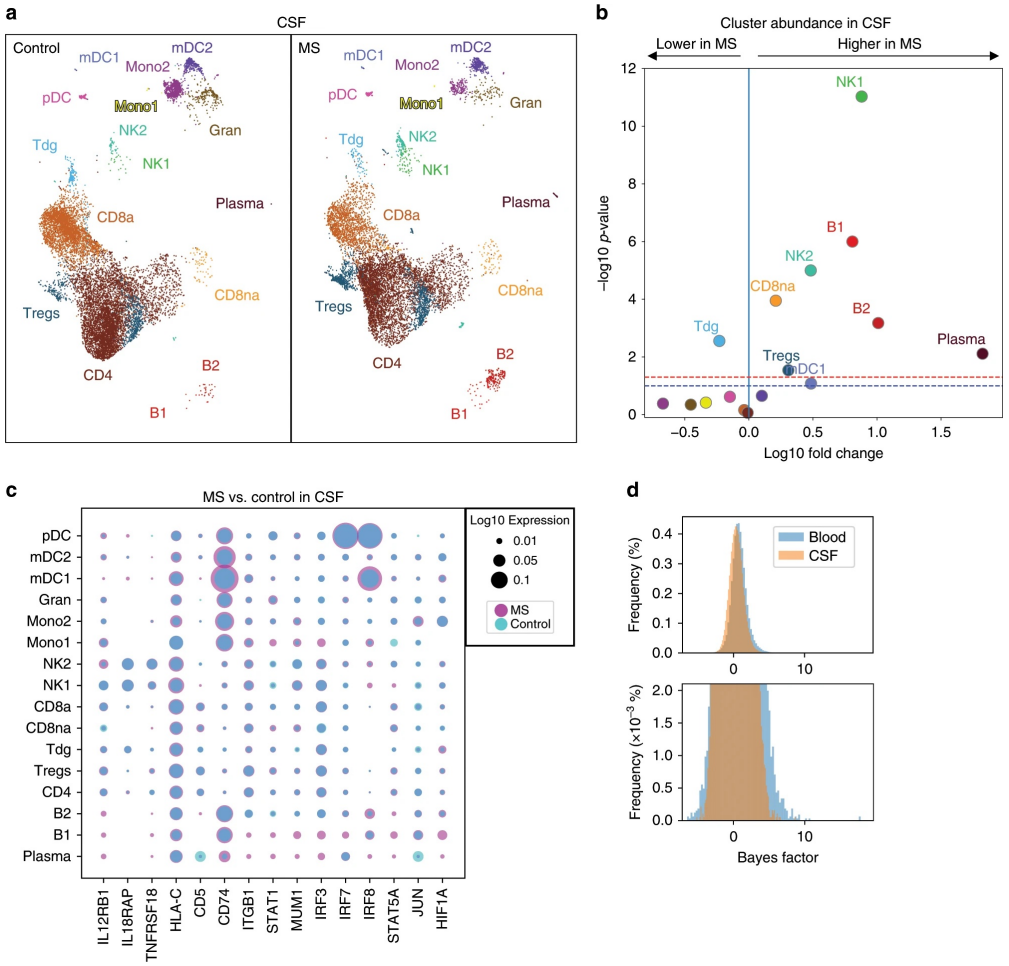
In contrast to blood, the cell type composition of CSF was clearly different in MS patients compared to controls (**Figure3.6**). Using binomial regression modelling (**Methods**), all B lineage cell clusters (B1, B2, plasma) significantly expanded in the CSF in MS compared to controls (**Figure3.6ab**) in accordance with flow cytometry (**Figure3.2bc**) and previous studies [218, 241, 242]. Heavy chain gene expression in mature B cell clusters (B2, plasma) was dominated by IGHG/IgG genes, although some cells expressed IGHA/IgA genes (Suppl. Fig. 6A-D). Most B lineage cells in the CSF are thus class-switched because heavy chain usage in blood evolves from IGHD to IGHM to IGHG/IGHA during maturation. The IGKC/κ-to-IGLC/*lambda* ratio was at 2.75 in CSF and 1.92 in blood. Additional comparison with published signatures confirmed our B cell cluster annotation and suggested some germinal center (GC) and plasmablast phenotype cells in the plasma cluster.

Figure 3.5: MS vs. Control UMAP, Cell Abundance and Expression Levels in Blood



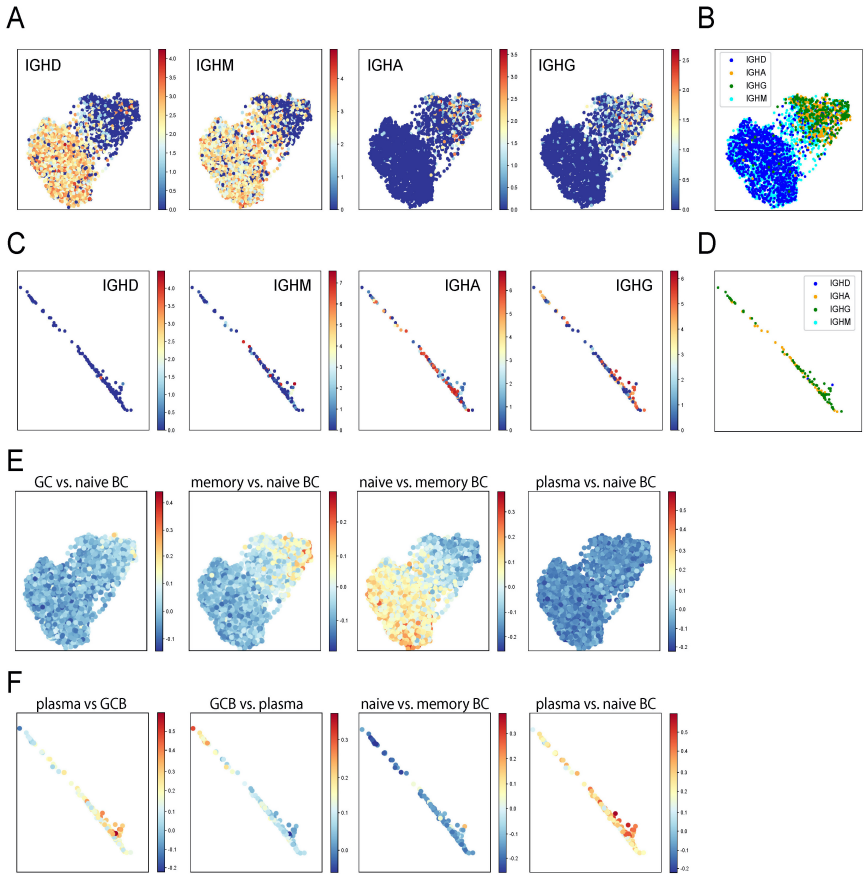
Unlike CSF, multiple sclerosis does not affect cluster composition in blood (*a*) UMAP plot depicting all blood cell clusters separated by disease status from control (left) and multiple sclerosis (MS, right) patients. (*b*) Volcano plot depicting differences of cluster abundance among all blood cells in MS samples compared with control plotting fold change (log10) against p-value (-log10) based on binomial regression modeling (Methods). Horizontal line indicates significance threshold. (*c*) Dotplot depicting selected genes differentially expressed in some clusters of blood cells in MS compared to controls. Purple dot size encodes the average expression in MS and turquoise dot size encodes the average expression in controls. Dots are partially transparent thus overlap is dark, blue purple edge around dark blue circle indicates higher expression in MS, while turquoise edge around dark blue circle indicates lower expression in controls.

Figure 3.6: MS vs. Control UMAP, Cell Abundance and Expression Levels in CSF



MS predominantly alters CSF cell composition and blood cell transcription (*a*) Comparative UMAP plots depicting only CSF cells from control (12,705 cells, left plot) and MS (9,652 cells, right plot) donors. Color coding and cluster names are as in Figure 1. (*b*) Volcano plot showing differences of cluster abundance of only CSF cells in MS samples compared to controls plotted as fold change (log10) against p-value (-log10) based on beta-binomial regression. (*c*) Dotplot depicting selected genes differentially expressed in at least one cluster of MS cells compared to controls in CSF. Dot size encodes percentage of cells expressing the gene. Purple indicates higher, turquoise indicates lower expression in MS, respectively. (*d*) Bayes Factor (BF) frequency histogram in all cluster-specific case-control differential expression (DE) analyses colored by tissue. Higher magnification in bottom panel. Only clusters with a minimum of 10 cells per tissue per disease state are included. Please note that the BF is proportional to the likelihood of differential expression (i.e. higher BF indicates more likely DE) [43].

Figure 3.7: Late B lineage cells accumulate in the CSF in MS



Late B lineage cells accumulate in the CSF in MS

(*a*) Feature plot showing the expression level of heavy chain genes IGHD/IgD, IGHM/IgM, IGHA/IgA, IGHG/IgG in the B cell clusters identified as B1/B2 in Figure 1A. Please note that IGHG summarizes IGHG1-4 genes and IGHA summarizes IGHA1-2 genes. (*b*) Cells expressing the respective IGH genes in the B cell clusters at maximum are highlighted. (*c*) Feature plots showing expression of heavy chain genes in the plasma cluster identified in Figure 1A. (*d*) Maximum expression of heavy chain genes in the plasma cluster. (*e*) B cell-related gene signatures described previously [243] were obtained from GEO, accession number GSE12366. UMAP feature plots for B cell clusters representing VISION signatures with significant VISION consistency scores ($P < 0.01$), from left to right: GC vs. naive B cells , memory vs. naive B cells, naive vs. memory B cells and plasma vs. naive B cells. (*f*) Feature plots for the plasma cell cluster representing VISION signatures with significant VISION consistency scores ($P < 0.01$), from left to right: plasma vs. GCB, GCB vs. plasma, naive vs. memory B cells and plasma vs. naive B cells. GC germinal center, GCB germinal center B cells.

Among other cell lineages, both $CD56^{dim}$ NK1 and $CD56^{bri}$ NK2 cell clusters and the CD8na cluster increased in the CSF in MS compared to controls (**Figure3.6**) as confirmed by flow cytometry (**Figure3.2bc**) and in line with a previous study [244]. In addition, we identified a previously undescribed increase of mDC1 cells and Tregs in the CSF in MS, while $\gamma\delta$ T cells (Tdg) were significantly decreased (**Figure3.6ab**). Alternative t-test statistics returned comparable results (data not shown). MS thus induced complex changes of the composition of CSF leukocytes that are characterized by a simultaneous expansion of cell types with the capacity for antibody production (B1, B2, plasma), cytotoxicity (CD8na, $CD56^{dim}$ NK1) and with regulatory potential (Tregs, $CD56^{bri}$ NK2).

We next tested for disease-associated 'triple-consistent' transcriptional changes in CSF cell clusters. In CSF T cells, we found an induction of genes associated with immune activation (*HLA-C, CD5*) and with interferon responses (*IL12RB1, IL18RAP*) and related downstream signaling molecules (*IRF3, IRF8*) (**Figure3.6(c)**). Specific trafficking molecules (e.g. *ITGB1*/integrin-β1) were also up-regulated in MS. The CD8a cluster showed signs of increased memory formation (ID3). The Treg cluster showed induction of the transcription factor STAT1 and some interferon-regulated genes (*MUM1, NUCB2*). The mDC2 cluster induced B cell related genes (e.g. *CD79A, CD74*) and signs of IL-2 signaling (*STAT5A*) and a co-inhibitory molecule (*TNFRSF18*/GITR). B cell clusters did not exhibit differentially expressed genes potentially indicating that MS preferentially induces numerical rather than phenotypic differences in B lineage cells in the CSF. The MS-associated cellular response in CSF was thus diverse and lineage specific and showed signs of interferon-regulated responses.

When directly comparing effects of MS between CSF and blood, we found that surprisingly a greater proportion of genes was differentially expressed in blood than in CSF. For example, when performing the MS vs. control comparison, more genes (n = 354) were differentially expressed (DE) within the CD8a cell cluster in blood than within the same cluster in the CSF (n = 24). This trend towards more DE genes in blood than in MS was maintained across all cell clusters. Overall, when plotted across all clusters and genes, the Bayes factor (a measure of likelihood of differential expression that does not depend on sample size) of the MS vs. control comparison showed more extreme values in blood than in CSF (**Figure3.6**d). Then we subsampled each cluster to have the same number of cells in blood and CSF and ran the Mann-Whitney U test and observed that the blood case-control had more significant P-values and those P-values were more extreme (data not shown). In blood, MS thus preferentially increased transcriptional diversity, while in CSF it preferentially increased cell type diversity suggesting compartment-specific disease mechanisms.

T helper cells with cytotoxic phenotype are increased in multiple sclerosis

We had tentatively handled the CD4$^+$ T cell cluster as one cell type, because this population did not form clearly distinct sub-clusters (**Figure3.1(b)**) and because many well-established T cell protein markers are expressed lowly on transcript level. We therefore next aimed to better characterize the CD4$^+$ T cells using dedicated approaches. We performed sub-clustering of the CD4$^+$ T cell cluster (**Figure3.8**). As expected for an unsupervised clustering approach [245], we found a minor population of CD8 T cells (*CD8B*; CD4$^+$ T cell sub-cluster (CD4Tc) #8; 7.54% of all CD4$^+$ T cells) 'remaining' within the tentative CD4$^+$ T cell cluster (Fig. 3A,B). The CD4$^+$ T cells broadly separated into naïve-like (SELL, CCR7; CD4Tc #5,11,1,2) and memory-like (CD44; CD4Tc #9,4,0,3,6,7) clusters based on marker gene expression (Fig. 3B). Memory cells further separated into subsets with mostly effector memory-like (CD69; CD4Tc #3,0,4) and central memory-like (CD27; CD4Tc #7,6,9) phenotype. We also identified a cluster of likely Treg identity (FOXP3, CTLA4, CD4Tc #10, Fig. 3B) located at the intersect between naïve and memory cells. Notably, this cluster expressed individual markers of T cell exhaustion (TIGIT) [246] previously associated with loss of suppressive capacity of Tregs in the tumour micro-environment [247].

We next used VISION (previously named FastProject [248]) to identify transcriptional signatures rather than individual marker genes to better interpret the CD4$^+$ T cell sub-clustering. Transcriptional signatures identified a transcriptional gradient ranging from naive to memory T cell state. This was in line with previous findings in rodents [249], and potentially indicated that CD4$^+$ T cells generally form transcriptional gradients rather than distinct subclusters also explaining the poor applicability of clustering approaches alone for this cell type.

We next sought to identify compartment- and disease-specific changes among CD4$^+$ T cell sub-clusters. We found that several memory-type clusters (CD4Tc #3,4,0,9) were more abundant in CSF compared to blood while naïve clusters (CD4Tc #1,11,2) and exhausted Tregs (CD4Tc #10) were less frequent using t-test based statistics (**Figure3.8(c)**) in accordance with previous studies [238, 239]. Disease-associated changes in blood were limited to a reduction of a single memory-like cluster (CD4Tc #4) in MS compared to control (**Figure3.8(d)**). Transcriptional changes in blood and CSF did not encompass any of the key T helper cell lineage transcripts (e.g. *TBX21, GATA3, RORC*). In CSF, a CD4$^+$ T cell sub-cluster (2,240 cells) of memory cells was significantly more abundant in MS vs. control (CD4Tc #0; (**Figure3.8**)). This cluster expressed multiple genes associated with cytotoxic function (*GZMB, PRF1, CCL5*) despite similar levels of CD4$^+$ T cell marker genes (*CD4, IL7R*), low doublet probability (predicted doublet t-test p value 0.68), and absence of CD8 or NK cell markers (*CD8B, NKG7*; **Figure3.8(e)**) in this population. This gene signature showed considerable similarity with a recently described population of cytotoxic CD4$^+$ T cells [57] that is enriched within the CD4$^+$ T cells effector memory recently activated (TEMRA)

compartment. To independently confirm this, we quantified $CD4^+CD45RA^+CD27^-$ TEMRA cells and $CD4^+CD25^{high}CD127^{low}$ Tregs by flow cytometry in the CSF of newly recruited donors (data not shown). Both populations were significantly more abundant in MS than in controls). This indicates that cytotoxic $CD4^+$ T cells and Tregs [250] expanded in the CSF in MS.

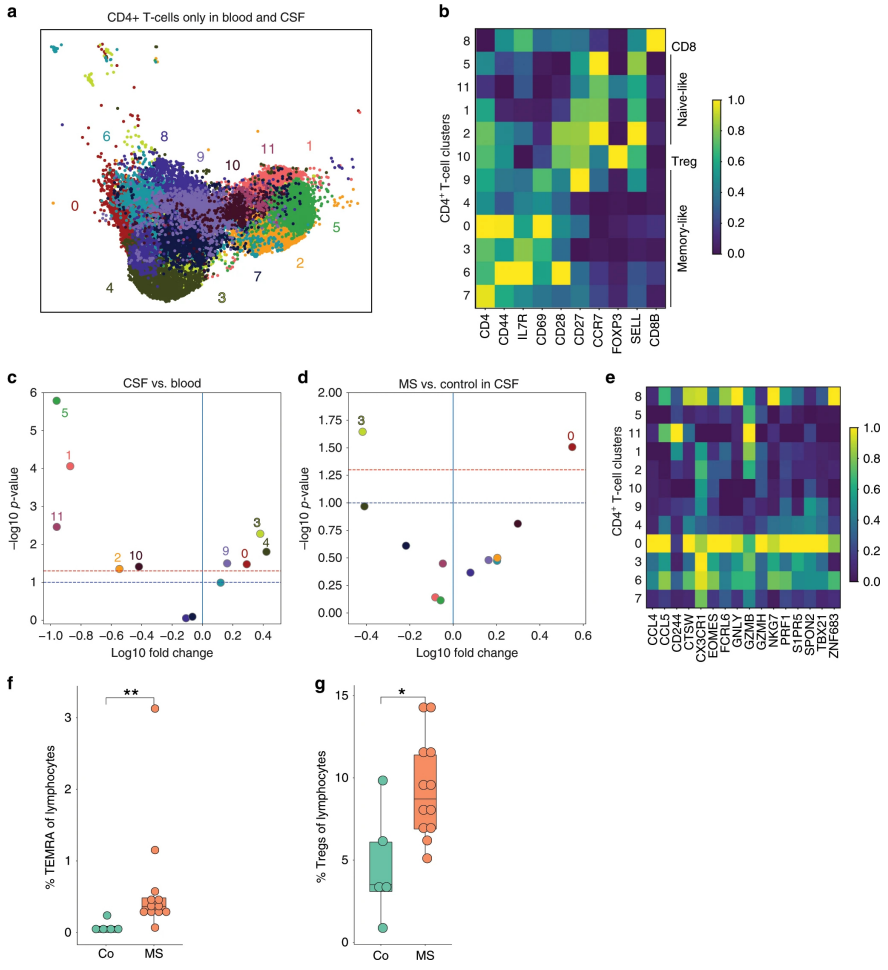
Cell set enrichment analysis (CSEA) identifies cluster-independent transcriptional changes.

Although the clustering analysis was informative about the general cell states, it was not readily able to identify a stratification of the cells into specific T helper cell subsets. We therefore developed a novel procedure –cell set enrichment analysis (CSEA) –which reuses the GSEA test for working on ranked lists of cells rather than genes (**Methods, Figure3.9**). Available bulk expression data are used to identify gene signature sets characterizing immune cell populations (top left). These gene sets are used for either (i) gene set enrichment analysis (GSEA) of our scRNA-seq differential expression results (top middle) or (ii) single-cell VISION signature scores, input to both VISION consistency testing and cell set enrichment analysis (CSEA) testing (bottom). See **Methods** section for details. The red line is the enrichment score (ES) trajectory of the signature gene set, while the blue lines are the ES trajectories of all randomized genesets. A more extreme positive value in the left-most part of the figure indicates enrichment of cells expressing the signature set, compared to cells expressing the randomized genesets. In this procedure, the cells are first ordered by a transcriptional phenotype of interest (e.g., summed expression of genes in a pathway). The statistical test can then detect cases in which a subset of cells from one group (e.g., MS) exhibit unusually high or low values of that transcriptional phenotype compared to cells from the second group (e.g., control). We used this analysis with signature scores obtained from the VISION pipeline based on signatures obtained from databases and literature curation (**Methods**) to specifically analyze $CD4^+$ T cells from CSF and blood.

Our CSEA testing procedure returned lists of cell sets significantly (**Methods**) enriched in MS and expressing a certain gene signature. The cell sets that were enriched in MS when compared to controls expressed signatures of T helper cell type 1 (Th1) [54] and T follicular helper (TFH) cells [251] (**Figure3.9bc**). We found that the TFH signature was enriched in the CSF ($P = 0.002$) but not in the blood ($P = 0.889$). Th1 cells are significantly enriched in both blood ($P=0.012$) and CSF ($p=0.0$). The leading edge size reflects the number of cells driving the high enrichment score (ES). In all cases the leading edge is small (< 600 cells) indicating that a subset of cells is driving the enrichment. We also generated a random geneset that is matched to the original signature set in both number of genes and the average expression of each gene (**Methods**). The enrichment score (ES) of the signature set is higher than that of the random genesets. Similar results were also obtained with more loose

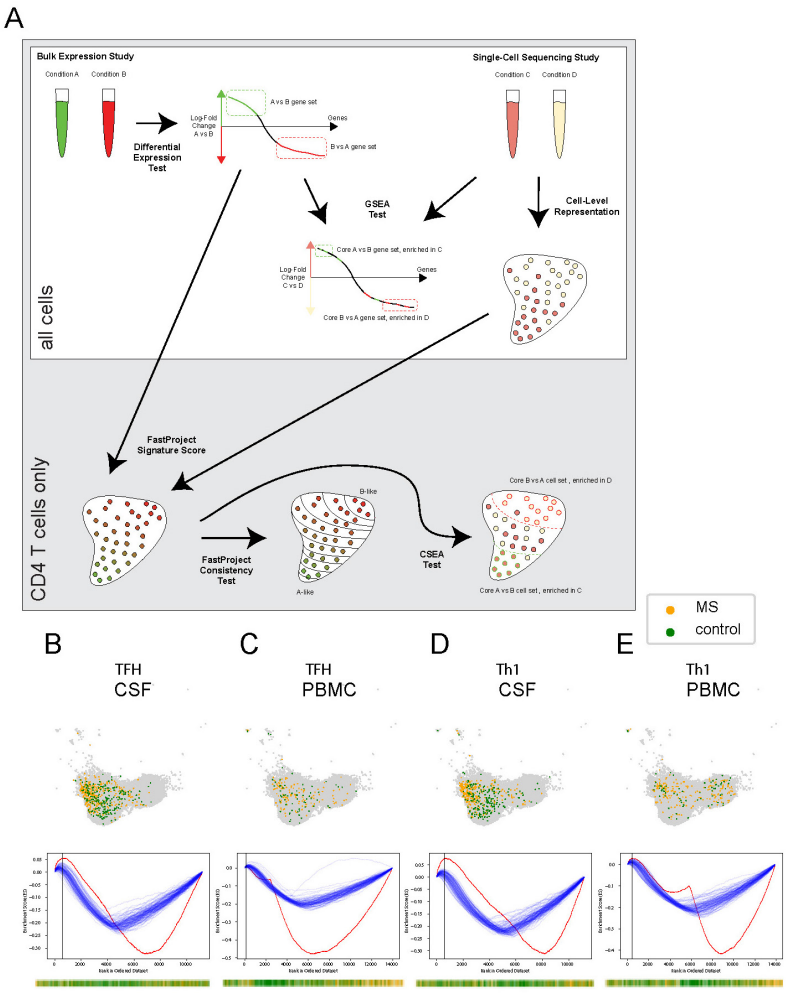
average expression matching (results not shown). Thus, $CD4^+$ T cells expressing a $Th1^-$ and TFH-like signature were enriched in MS in the CSF, but were spread across sub-clusters. Our novel analytical approach could therefore decouple clustering of cells from disease-state or differentiation-state enrichment of cells, providing a new framework for interpreting complex scRNA-seq datasets. Interestingly, TFH cells are required for B cell maturation [252]. This lead us to hypothesize that TFH might be functionally related with the MS-specific B cell-expansion in the CSF.

Figure 3.8: Cytotoxic-like population of CD4 T cells is induced in the CSF in MS



(*a*) UMAP plot showing sub-clustering of all CD4$^+$ T cells combined from blood (13,933 cells) and CSF (11,172 cells). Sub-clusters are numbered 0-11. (*b*) Heatmap depicting per cluster average expression of selected T cell subset marker genes. (*c*) Volcano plot showing differences of CD4$^+$ T cell cluster abundance in CSF compared to blood as fold change (log10) against p-value (-log10) based on Student's t-test. (*d*) Volcano plot showing differences of CD4$^+$ T cell cluster abundance in MS compared to control within CSF based on Student's t-test. (*e*) Heatmap showing average gene expression of selected cytotoxicity markers derived from [57]. (*f*) The proportion of TEMRA cells (CD45RA+CD27-) among live lymphocytes in the CSF of control (co; n = 5) and MS (n = 12) patients was quantified by flow cytometry. (*g*) The proportion of Treg cells (CD25highCD127low) among live lymphocytes in the CSF of donors as in panel F was quantified by flow cytometry. * $P < 0.05$, ** $P < 0.01$.

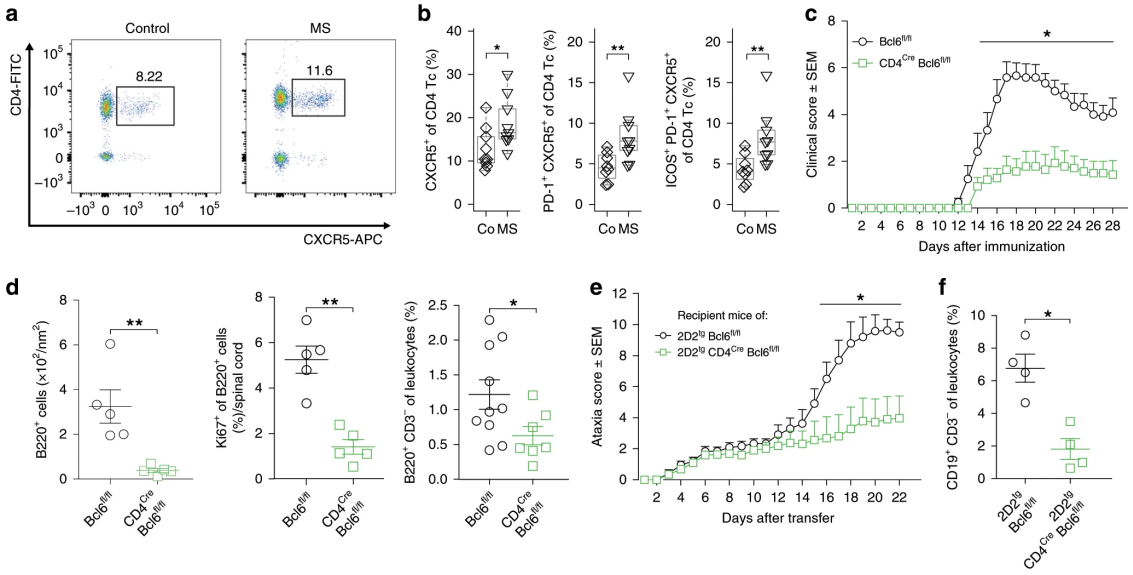
Figure 3.9: Cell set enrichment analysis (CSEA) identifies cluster-independent transcriptional changes



(a) Scheme of GSEA/VISION/CSEA Analysis. ($b-e$) Two selected CSEA analysis using TFH marker gene set (B,C) and Th1 marker gene set (D,E) in the CD4+ T cell analysis. TFH cells are significantly enriched in MS cells in CSF (b), but not in blood (c), while Th1 cells are enriched in both (D,E)Top row shows UMAP plots highlighting the leading edge cell sets from MS (orange) and control (green) samples. Cells depicted in grey are not part of the leading edge cell set of the respective signatures. Bottom row shows the enrichment score (y-axis) as a function of the rank of the cells by their Vision signature score (x-axis). The red line indicate the position of the leading edge. The 1D density plot shows the Vision signature score data with all MS cells in orange and control cells in green with the x-axis being the rank of the cells by their Vision signature score.

B cell-helping T follicular helper cells expand in the CSF in multiple sclerosis and exacerbate corresponding animal models

Figure 3.10: TFH cells expand in the CSF in MS and promote MS animal models



T follicular helper (TFH) cells expand in the CSF in MS and promote MS animal models (*a*) Representative flow cytometry dot plot of CSF cells from a control and MS patient after gating on live CD3+ cells. (*b*) Proportion of CXCR5+ (left), of PD-1$^+$CXCR5$^+$ (middle), and of ICOS$^+$PD-1$^+$CXCR5$^+$ (right) cells among live CD3$^+$CD4$^+$ T cells in CSF of control (co; n = 9) and MS (n = 9) patients quantified by flow cytometry. (*c*) Active EAE was induced in Bcl6fl/fl (wildtype, circles, n = 6) and CD4CreBcl6fl/fl (squares, n = 7) mice using MOG35-55 peptide (**Methods**). Mice were monitored daily for clinical EAE signs. One representative of three independent experiments is shown. (*d*) At day 28 after EAE induction, the density of CD3-B220+ leukocytes was quantified in spinal cord paraffin cross-sections by histology (left). The proportion of Ki67+ among B220+ cells was quantified (middle). The proportion of B220+ cells was quantified by flow cytometry at peak of EAE (right). (*e*) Naive CD4$^+$ T cells were sorted from Bcl6fl/fl2D2tg mice (circle) and CD4CreBcl6fl/fl2D2tg mice (squares), differentiated in vitro (**Methods**), and intravenously injected into wildtype recipient mice (n = 6-8 per group) at 5x106 cells per mouse. Recipients were monitored for signs of EAE. One representative out of five independent experiments is shown. (*f*) At day 28 after transfer, the proportion of CD3-CD19+ leukocytes in brain and spinal cord was quantified by flow cytometry. * $P < 0.05$, ** $P < 0.01$, *** $P < 0.005$.

We therefore next tested whether TFH cells are in fact altered in the CSF in MS. We identified CD3$^+$CD4$^+$CXCR5$^+$ TFH cells in the CSF by flow cytometry and found a significantly increased proportion of TFH cells in MS patients (**Figure3.10ab**) in accordance with previous studies in the blood [26, 253] and CSF [254]. Activated PD-1+ and PD-1+ICOS+ TFH

cells were also increased in the CSF (**Figure3.10**) while the alternative $CD4^{+}CXCR5^{-}PD\text{-}1^{+}$ subset [255] was unchanged (data not shown). The percentage of PD-1+ TFH cells in CSF positively correlated with the proportion of CSF plasma cells quantified by flow cytometry ($r = 0.70$, $p < 0.05$). Next, we performed bulk population RNA-seq from sorted TFH cells from the CSF of MS patients ($n = 7$) vs. controls ($n = 6$) to better characterize this cell type. Surprisingly, no genes reached the significance threshold for differential expression. This indicated that CSF-resident TFH cells increase in abundance, but do not considerably alter their phenotype in MS. We then performed GSEA and found an enrichment of gene-sets (not individual genes) associated with T cell memory and pathogenicity in MS-derived TFH cells ($P < 0.01$, Bonferroni correction). Genes recurring in these enriched gene-sets were associated with cytotoxicity (e.g. *GZMA, GZMK, CASP3, CASP4*) and co-inhibitory function (e.g. *KLRG1, TIGIT, CTLA4*). Although statistically less stringent, this approach indicated that pathogenic TFH cells may expand in the CSF in MS patients.

We then tested whether TFH cells in fact promote neuro-inflammation to a functionally relevant extent using common animal models of MS. We generated mice with T cell-restricted deficiency of *Bcl6*⁻ the lineage-defining transcription factor of TFH cells [252]. Such $CD4^{Cre}Bcl6^{fl/fl}$ mice lack TFH cells and fail to mount antigen-specific B cell responses [252], while the differentiation of other T helper cell lineages(**Figure3.10(a)**) and the composition of the peripheral immune compartment after immunization were unchanged (**Figure3.10(b)**) as previously described [256].

We induced active EAE using myelin oligodendrocyte glycoprotein $(MOG)_{35\text{-}55}$ peptide in these mice and EAE severity was significantly reduced in $CD4^{Cre}Bcl6^{fl/fl}$ mice compared to Cre-negative littermates (**Figure3.10(c)**). Accordingly, the number of inflammatory lesions and infiltrated area in the spinal cord of $CD4^{Cre}Bcl6^{fl/fl}$ mice were lower than in controls. We tested how the absence of TFH cells influenced B cells in the CNS and found a lower proportion of B cells ($B220^{+}CD3^{-}$) infiltrating the CNS in $CD4^{Cre}Bcl6^{fl/fl}$ mice by flow cytometry and in the spinal cord by histology (**Figure3.10(d)**).

Pan-T cell deficiency of *Bcl6* in $CD4^{Cre}Bcl6^{fl/fl}$ mice will affect the priming phase of EAE and target both TFH cells and T follicular regulatory (TFR) cells [257]. To make a contribution of these potential confounders less likely, we next generated $2D2^{tg}CD4^{Cre}Bcl6^{fl/fl}$ mice expressing a T cell receptor transgene recognizing MOG [258] to enable immunization-independent adoptive transfer EAE induction. After transfer of interleukin (IL)-17 producing myelin-reactive T helper cells into wildtype hosts (**Methods**), $2D2^{tg}CD4^{Cre}Bcl6^{fl/fl}$ control T cells induced considerably more severe EAE than *Bcl6*-deficient $2D2^{tg}CD4^{Cre}Bcl6^{fl/fl}$ donor cells (**Figure3.10(e)**). This was despite comparable pre-transfer polarization of donor T cells. Control recipients also showed a higher proportion of B cells in the CNS than recipients of **Bcl6**-deficient T cells. Taken together, our data indicated that TFH cells locally drive B cell responses in the CNS and promote MS-like autoimmunity.

3.3 Methods

Patient recruiting and inclusion

A total of 54 control patients and 60 MS patients were screened for eligibility. After screening, a total of 39 treatment-naive patients with MS or clinically isolated syndrome (CIS) receiving a lumbar puncture (LP) for diagnostic purposes, were prospectively recruited. The control group consisted of 27 patients diagnosed with idiopathic intracranial hypertension (IIH). Patients were recruited in four consecutive cohorts. Cohort 1: single-cell RNA-seq of unsorted CSF cells (named scRNAseq; 6 IIH vs. 6 MS patients). Cohort 2: CSF cell flow cytometry only using a general flow cytometry staining panel (named flow only; 7 IIH vs. 12 MS patients; gating in **Figure3.2**), cohort 3: flow sorted $CD3^+CD4^+CXCR5^+$ TFH cells from CSF for bulk RNA-seq (named TFH RNAseq; 9 IIH vs. 9 MS patients), cohort 4: CSF cell flow cytometry using a staining panel designed to quantify $CD4^+$ TEMRA cells and Treg cells (named validation; 5 IIH vs. 12 MS patients; gating in (**Figure3.2**). All patients were of Caucasian ethnicity and gave written informed consent. The study was performed in accordance with the declaration of Helsinki and approved by the ethics committee of the Westfälische Wilhelms University Münster under reference number 2015-522-f-S.

Generation of single-cell libraries and sequencing

Single-cell suspensions were loaded onto the Chromium Single Cell Controller using the Chromium Single Cell 3' Library & Gel Bead Kit v2 (both from 10X Genomics) chemistry following the manufacturer's instructions. Sample processing and library preparation was performed according to manufacturer instructions using AMPure beads (Beckman Coulter). Sequencing was carried out on a local Illumina Nextseq 500 using the High-Out 75 cycle kit with a 26-8-0-57 read setup.

Preprocessing of sequencing data

Processing of sequencing data was performed with the cellranger pipeline v2.0.2 (10X Genomics). Raw bcl files were de-multiplexed using cellranger mkfastq. Subsequent read alignments and transcript counting was done individually for each sample using cellranger count with standard parameters. Cellranger aggr was employed, to ensure that all samples had the same number of confidently mapped reads per cell. The cellranger computations were carried out at the High Performance Computing Facility of the Westfälische Wilhems-University (WWU) Münster.

Single-Cell Sample Filtering

Initial exploratory data analysis identified one MS sample and one IIH sample whose clustering did not overlap with any of the other samples (data not shown). This suggested strong batch effects. Both samples were excluded from further analysis, leaving 4 control- and 4 MS-derived samples from CSF and 5 control and 5 MS-derived samples from PBMC. Nine barcode-level quality control (QC) metrics were computed for the unfiltered 10x Cell Ranger output: (1) number of unique molecular identifiers (UMIs), (2) number of reads, (3) mean reads per UMI, (4) standard deviation of reads per UMI, (5) percent of reads confidently mapped to the gene, (6) percent of reads mapped to the genome but not a gene, (7) percent of reads unmapped, (8) percent of UMIs corrected by the Cell Ranger pipeline, and (9) the number of cell barcodes corrected by the Cell Ranger pipeline. These metrics were used for filtering and normalization. We applied the gene and sample filtering using a scheme previously described [259]. This involved four steps: Define common genes based on UMI counts: Genes with nu or more UMIs in at least 25% of barcodes, where nu is the upper-quartile of the non-zero elements of the UMI matrix.

Filter samples based on QC metrics. Remove samples with low numbers of reads, low proportions of mapped reads, or low numbers of detected common genes. The threshold for each measure is defined data-adaptively: A sample may fail any criterion if the associated metric under-performs by zcut standard deviations from the mean metric value or by zcut median absolute deviations from the median metric value. Here we have used zcut = 2. This function is implemented in scone::metric_sample_filter (see below).

Remove barcodes from donors with fewer than 100 barcodes following sample filtering. These donors have contributed too few high-quality samples to reliably estimate donor-specific effects. Only seven cells were removed in this step.

Filter genes based on UMI counts: Genes with nu or more UMIs in at least ns barcodes, where nu is the upper-quartile of the non-zero elements of the sample-filtered UMI matrix. We have set ns = 5 to accommodate markers of rare populations. This sub-step ensures that included genes are detected in a sufficient number of samples after sample filtering. For the $CD4^+$-only analysis this step was applied again after the data matrix was subset to include only $CD4^+$ clusters.

Single-Cell Analysis

Harmonization

We utilized a Bayesian variational inference model scVI [43] to infer a shared latent space of dimension 10 for all single cells from different tissue, condition and batches. Visualizations were generated using UMAP to further reduce the latent space to two dimensions. scVI is a deep generative model that learns a probabilistic representation of the transcriptional states

of single-cells conditional on the sequencing batches, thus no explicit library size and batch correction is needed.

Level 1 Clustering Analysis

After sample filtering, we performed louvain clustering on the scVI latent space as implemented in `https://github.com/taynaud/python-louvain`. We first constructed a k-nearest-neighbor graph from the scVI latent space, and then used the louvain.find_partition function with the ModularityVertexPartition method to recover a total of 25 clusters. Three of these clusters correspond to CD4 T cells and were tentatively combined into a single cluster for further analysis resulting in 22 first level clusters. From this, we removed one red blood cell (RBC) cluster (2,333 cells; *HBA1, HBA2, HBB*), three clusters with high doublet probability (see below) and one blood-derived cluster with low quality (mitochondrial genes, no canonical marker genes) (361 cells) for further analysis.

Doublet Detection

We computed a doublet score for each single cell using the function scrub_doublets in the Scrublet package [260] with all default parameters. We then removed all clusters with greater that 20% of cells labeled as doublets (1,186, 290 and 105 cells) including one cluster of lower quality cells, one cluster expression Monocyte marker genes, and one cluster expression B cell marker genes.

Level 2 Clustering Analysis

For cells that were classified as a single cluster but two distinct clusters were visible on UMAP visualization (Monocytes, B cells and mDC cells), we performed further clustering on the scVI latent space using Spectral Clustering from the scikit-learn package SpectralClustering with number of cluster set to 2 and affinity matrix computed using k-nearest-neighbor with k=15 . The clusters we visually identified on UMAP were confirmed to be the same as the results of Spectral Clustering. With further validation using signature genes, we included the second level clusters into the main analysis. Monocyte cluster separated into Mono1 and Mono2, B cell clusters separated into clusters B1 and B2, and mDC1 separated into mDC1 and mDC2.

T cell clustering analysis

For all CD4 T cells (excluding regulatory T cells), we performed Louvain clustering on the scVI latent space, excluding all other cells. With the same parameters as the Level 1 clustering analysis. We partitioned the CD4 T cells into a total of 12 clusters.

Systematic comparison with published microglia and CSF datasets

We obtained the key marker genes of myeloid lineage cell clusters from recent publications [234, 261, 262, 216] and plotted their expression onto our combined dataset. We extracted the combined oligodendrocyte markers from a study performing single nuclei RNA-sequencing of frozen brain parenchyma [261] and selected genes that are also highly variable in our dataset (genes *APOE, CD74, HLA-DRA, PTPRC, C3*). We extracted markers of five myeloid clusters from a CSF-based study [216] (genes *C1QB, C1QC, APOE, C1QA, LYVE1, SEPP1, FCGBP, APOC1, C3, A2M, MSR1, EPB41L2, MARCO, RNASE1, F13A1*). We also obtained microglia markers (*TMEM119, CCL4, P2RY13, EGR2, CX3CR1, CCL2, SLC2A5, EGR3, CD83*) and markers of MS-associated microglia markers (*CTSD, CD74, SPP1, APOC1, HLA-DRA, PADI2, GPNMB, HLA-DRB1, ANXA2, HLA-DPB1, CPM, LGALS1, LYZ, LIPA, APOE, MAFB*) [234]. We extracted marker genes from a rodent study (*NLRC5, IL12RB1, PSMB9, TAP1, TAP2, IFIH1, IRF7, ZBP1*) [262]. We then plotted the combined expression level of the respective gene signatures into our combined blood and CSF dataset.

VISION Analysis

We passed raw and normalized UMI data to the VISION pipeline (https://github.com/YosefLab/VISION) [248]. Mean expression per gene symbol was calculated prior to the analysis in order to make the features relatable to general gene signatures. The goal of FastProject analysis –on which VISION is based –is to uncover biologically meaningful gene signatures that vary coherently across single-cell neighbourhoods [248]. These signatures can help assign meaning to the dominant expression differences between clusters. In addition to raw data, we passed QC, donor, status, and Seurat cluster covariates for exploratory analysis and visualization. VISION quantifies the extent to which cell signature values cluster across the cell manifold by using "consistency testing."VISION scores the extent to which neighbouring cells (similar expression profiled) are predictive of a cell's signature value using autocorrelation (Giri's C) statistics, comparing against random permutations in order to assign statistical significance with respect to a uniform null model. We also included the Seurat t-SNE as a precomputed projection. Our signature set includes: Human cell cycle genes described before [25], representing sets of genes marking G1/S, S, G2/M, M, and M/G1 phases. The MSigDB C7 immunological signature collection [263]. TH signatures compiled previously [54]. NetPath database signatures [264]. Curated T cell signatures [249]. Curated TFH [251] signature sets. Curated Temra signature [57] Housekeeping genes were referenced from the same source as the SCONE negative controls above [265].

Comparing gene expression and cluster composition

Differential Composition Analysis

For both the initial and the CD4$^+$-only clustering, we used t-test and beta-binomial generalized linear model in package aod::betabin [266] to test the difference in cluster abundances (cell counts) between MS donors and control donors. We used both methods because when cell types are rare, the estimated proportions of a cell type in each donor might be over-dispersed. The two methods show consistent results and thus we show the differential composition analysis from the beta binomial distribution comparison. For the beta-binomial regression model unless indicated in the figure legends, we set the count of the cell type of interest and the total count of cells of each donor to be the response variable and the state of the donor (MS or control) or the tissue of origin (CSF or blood) to be the independent variable. We tested for Pearson's correlation between the frequency of each B cell cluster and cluster 0 in CD4 T cells. We adjusted the P value threshold to $0.05/15 = 0.0033$, since we tested for significant correlation using three B cell subsets in 5 different sample partitions (all samples, CSF only, blood only, MS only and control only). The abundance of cluster 0 in CD4 T cells is not signifiantly correlated to B cell subset abundances in any of these comparisons.

Differential Expression Analysis

We used three different tests for the discovery of differentially expressed genes between two groups of cells. First we computed Bayes Factor using the imputed counts from scVI. Bayes Factor is a generalization of the p-value and is computed as

$$logP(x_a x_b)P(x_b > x_a)$$

where x_a is the gene expression of the gene of interest in group a. and x_b in group b. We use the generative model of scVI to obtain the batch-corrected mean of the negative binomial distribution of transcript counts. Second we used the library-size corrected UMI counts for Mann-Whitney U test. At last we followed the methods of the best performing method in a single-cell specific DE method assessment paper [267] and we used EdgeR [235] with cellular detection rate and batch id as covariates.

Gene Set Enrichment Analysis (GSEA)

After deriving lists of differentially expressed (DE) genes, we applied GSEA tests [268] to all cluster specific DE gene lists DE between CSF and blood. We used the enrichr function in gseapy v0.9.12 to find overlap between the DE genes and function genesets. We used signed significance scores based on the Adjusted P-value provided by the enrichr function. Sets considered in this analysis include all MSigDB C7 signature sets and all curated T cell

signature sets described previously [249] with 10 or more genes quantified in the present study; "UP" and "DN" signature subsets were tested separately.

Cell Set Enrichment Analysis (CSEA)

For the CD4$^+$ T cells analysis we developed a novel adaptation of the GSEA method, applying the technique to cell sets: CSEA (**Figure3.9**). CSEA is a hypothesis testing method for simultaneously uncovering enrichments and identifying subsets of cell sets of importance. In this procedure, a collection of cells is first ordered by a transcriptional phenotype of interest (e.g., sum expression of genes in a pathway). The resulting statistical test is sensitive to cases in which only a subset of cells from one group (e.g., MS) exhibit unusually high values of the transcriptional phenotype. The input to this method is a list of N cells, rank-ordered by some input signal. Our analysis uses VISION signature scores, reflecting known axes of biological variation. VISION signature scores –based on FastProject signature scores [248] –are computed by first centering and scaling each normalized log expression cell profile. Following scaling, the sum of gene expression values in the negative signature subset are subtracted from the sum of gene expression values in the positive signature subset. Signatures are normalized to the total number of genes in the set. For example, a signature set that describes a dichotomy between naïve and memory T cells may be used to score individual cells, indicating that some cells have higher expression of genes characterizing the naïve state and lower expression of genes characterizing the memory state. Using the notation previously described [268] we will use r_j to denote the cell j's signature score; indices have been sorted so that $r_j > r_j + 1$. The test involves considering all cells up to a specific position, i. A "hit"score is defined as the signature score optionally exponentiated by parameter $p(|r_j|p)$ for members of cell set S, divided by the sum over all set members in the list. A "miss" score is similarly calculated for non-members of S, but without weighing by signature score magnitudes.

The CSEA enrichment score (ES) is defined as the maximum of the difference between the running cumulative sum of hit scores and miss score with respect to index i. When $p = 0$, the ES reduces to a one-sided KS test statistic for differential signature analysis between cell sets. When $p = 1$, the cells in S are weighted by their signature score, normalized by the sum of the score over all the cells in S. We apply the same permutation scheme as described for GSEA above. For $p > 0$, CSEA cannot be seen as a simple differential signature test: CSEA tests for enrichment of a cell set at the high tail of the signature score distribution, but additionally weighs the set elements according to their signature value. This reduces the effects of low-magnitude cells in S, whereas all cells not in S are treated the same no matter the magnitude of their signature score. CSEA tests if high magnitude (positive or negative) cells are enriched at a specific tail, applying permutation tests to account for the additional variability induced by the magnitude weights. The set of indices up to where the objective score reaches its maximum also holds significance –in GSEA [268] referred to as

the "leading-edge" of the enrichment test. The intersection of the set S and the leading-edge is the leading-edge subset, representing an important core subset of cells driving an enrichment. For each VISION signature, we treated the computed signature scores as cell signature scores r_j. The sets under consideration were the mutually exclusive sets of MS and control cells. The goal of this approach is to identify core sets of cells that drive each biological condition's enrichment for high signature values.

To screen a set of gene signatures, we computed the Vision signature score for 64 gene signatures related to CD4 T cell states, cell cycle, interlukin expression and T cell subsets. To determine the P-value of the CSEA enrichment score (ES) for a geneset, we shuffle the disease state labels for cells 100 times and compute the probability that the maximum ES computed with the true labels is greater than the ES computed with the shuffled labels. We also generated a random geneset that is matched to the original signature set in both number of genes and the average expression of each gene. This is done by finding the top 20 gene that has the closest mean expression to each gene in the original signature set, and then randomly sampling one of them. We then corrected for multiple testing using the Benjamini-Hochberg procedure to generate the corrected P-values. We filtered the result based on three criteria: the corrected P-value of the true signature set is smaller than 0.05, the corrected P-value of the control signature set is greater than 0.05, and that the leading edge is smaller than 1000. This results in 2 significant signatures, TFH and Th1. We then validated this result by computing the ES of 1000 matched control genesets for each signature set. We then report the P-value as the probability that the true signature set's maximum ES being greater than the maximum ES in the matched random genesets. We also computed the ES for enrichment in control and found that the enrichment score is significantly larger than the control set but the leading edge is much larger than for enrichment in CSF.

We also tested the performance of our model on varying match levels of the randomized geneset to the original signature set. The match levels do not affect the results significantly, showing that our conclusion is not driven by the gene-matching procedure itself. However when randomized genesets are selected completely randomly, the ES become extremely variable, showing that some degree of matching is required.

Bulk RNA-Seq of sorted TFH cells

TFH cells were sorted from the CSF of 9 MS donors and 9 IIH donors using a BD FACS Aria III cell sorter using an 85 µm nozzle and the drop delay was determined using BD Accudrop beads. Sorting was performed using sort precision mode "purity"for live $CD3^+CD4^+CXCR5^+$ cells. Antibodies against CD3 (UCHT1), CD4 (OKT4), CXCR5 (J252D4), PD-1 (EH12.2H7) and ICOS (C398.4A) were from Biolegend. Cells were sorted directly into 1.5 ml reaction tubes containing 100 µl RNA Lysis Buffer (Zymo Research). After sorting, tubes were vortexed, briefly centrifuged and frozen at -80 °C until RNA isolation. Data were analyzed using

FlowJo software v10.4.1 (Tree Star, Inc.). Samples for bulk RNA-sequencing were prepared using a modified version of the SmartSeq2 protocol [269]. Briefly, unquantified purified RNA was used as input. Reaction volumes were scaled up and the number of PCR cycles during cDNA amplification adjusted accounting for the higher number of input cells compared to the original protocol [269]. Library Preparation was done by the Next UltraII FS DNA Library Prep Kit (New England Biolabs) using 1-3 ng of cDNA as input. Sequencing was carried out on a NextSeq500 using the High-Out 75 cycle kit (Illumina).

Bulk expression quantification

RNA-seq reads were aligned to the RefSeq hg38 transcriptome (GRCh38.2) using Bowtie2 [270]. The resulting transcriptome alignments were processed using the RNA-Seq by Expectation Maximization (RSEM) toolkit to estimate expected counts over RefSeq transcripts [271]. Several genes were quantified multiple times due to alternative isoforms unrelated by RefSeq annotation. Before expression data normalization, the gene entry with maximum counts was selected to represent the gene in further analysis.

Sample and gene filtering were similar to the scRNA-seq filtering method above, enforcing ($>$ 107k reads, $>$ 10% read alignment (forced), $>$ 93.3% common genes detected; corresponding to zcut = 20). Out of 18 initial samples (9 control vs. 9 MS), 5 total samples (3 control vs. 2 MS) were removed after QC. Setting ns = 1, we analysed 11,383 genes below. For each sample, we computed transcriptome alignment and quality metrics using FastQC (Babraham Bioinformatics), Picard tools (Broad Institute), and custom scripts. Computed metrics included: (1) number of reads; (2) number of aligned reads; (3) percentage of aligned reads; (4) number of duplicate reads; (5) primer sequence contamination; (6) average insert size; (7) variance of insert size; (8) sequence complexity; (9) percentage of unique reads; (10) ribosomal read fraction; (11) coding read fraction; (12) UTR read fraction; (13) intronic read fraction; (14) intergenic read fraction; (15) mRNA read fraction; (16) median coefficient of variation of coverage; (17) mean 5' coverage bias; (18) mean 3' coverage bias; and (19) mean 5' to 3' coverage bias.

Data were normalized using SCONE. 569 positive controls were derived from MSigDB C7 entries annotated to include TFH cell types, including the most frequently included gene symbols in those entries. Negative controls for RUVg and evaluation were derived from the housekeeping gene list. Control lists were sampled down to 186 genes per list so as to match mean expression of genes in each list. The study group included two batches with 4/3 and 3/3 MS/IIH samples respectively. Biological condition was used only for evaluation. SCONE recommended TMM scaling and adjustment for 2 factors of RUVg and batch condition.

We performed PCA on the scaled log-transformed normalized data for visualization. DE between MMS and IIH donors was performed with limma-voom, using RUVg factors and

batch in the model to adjust for unwanted variation. Per-gene DE significance scores were computed from log-transformed P-values. No single gene reached significance after correction for multiple hypothesis testing. The 42 most frequent core members of the significant enrichments (Bonferroni adjusted P-value less than 0.01) –genes driving 7 or more of these enrichments –were selected and their normalized log values were correlated against eachother and represented in a sorted heatmap using pheatmap defaults.

3.4 Discussion

In this study, we constructed the first unbiased comparative single-cell map of blood and CSF cells. We identified a compartment-specific leukocyte transcriptome and composition including an unknown enrichment of mDC1 and Tregs in the CSF. Monocytes in the CSF were especially distinct and partly resembled CNS border-associated macrophages. These findings emphasized the unique immune microenvironment of the CSF. We used MS to test how a paradigmatic autoimmune disease would affect leukocytes in a compartment-specific fashion. Surprisingly, we found that MS preferentially increased transcriptional diversity in blood, while it increased cell type diversity in CSF thus providing evidence for compartmentalized mechanisms driving human autoimmunity in the brain. In MS-derived CSF, we found an expansion of cytotoxic-phenotype $CD4^+$ T cells [57] that could be involved in local MS pathology. We also found that clustering-based methods alone poorly capture disease-associated changes within $CD4^+$ T cells and developed CSEA as a new cluster-independent analytical approach to address this. This lead us to investigate TFH cells and these cells in fact expanded in MS within the CSF and promoted B cell accumulation and disease severity in MS-like animal models. Our study thereby provides a signature case for reverse translation from unbiased single cell transcriptomics in humans to disease mechanisms in rodents.

Our unbiased approach considerably extended the available flow cytometry-based characterization of CSF leukocytes [212, 226]. Notably, mDC1 cells abundant in the CSF expressed markers of cross-presenting capacity (*XCR1, WDFY4*; [227]) while NK2 cells in the CSF expressed the corresponding ligands (*XCL1, XCL2*) indicating that cell types equipped for cross-presentation and anti-viral defence circulate the CSF. We also replicated the known activated/memory phenotype [239, 214] of CSF-resident T cells and identified a distinct pattern of adhesion molecule expression in CSF leukocytes (**Figure3.8**). Such a repository of compartment-specific gene expression signatures could allow specifically targeting CSF cells in the future (e.g. *CCL3* in CSF myloid cells (**Figure3.5(c)**). This also allowed us to provide a human confirmation beyond a previous single case study in HIV [216] of the rodent border-associated macrophage cell phenotype [232, 233]. Our findings thus lended further support to a species-independent ‘peri-CNS immune system’ involved in local autoimmunity and anti-pathogen defense.

A plethora of studies have analyzed mechanisms of neuro-inflammation [272, 273] albeit often equating rodent models with human MS. Unlike our CSF-focussed study, purely human transcriptional studies often relied on easily accessible peripheral blood mononuclear cells [274] sometimes even using unsorted cells [275]. Some transcriptional studies of blood cells focussed on T cells [276], different treatments [277, 278, 279], or myelin antigen-specific T cells using pre-defined gene-sets [274]. However, whether blood leukocytes actually constitute a suitable surrogate of disease mechanisms in MS remains unknown. A single available transcriptomic study of unsorted bulk CSF cells in MS returned signs of local B cell expansion [215]. Invasive lumbar punctures are rarely justifiable in healthy individuals, which limits access to optimal controls in any CSF-based study [280]. Others have used somatoform disorders with the inherent risk of misdiagnosis [226]. We chose IIH controls, because they require large volume CSF removal, are well matched with MS patients with regard to sex and age, and because basic CSF parameters and B cells are unchanged in IIH [281, 226]. Some of the complex cellular changes we observed in MS vs. IIH may still be biased by this specific choice of controls. We also preferentially recruited untreated MS patients in (first) relapse to limit clinical complexity. The phenotype of CSF cells in remission or under MS treatments may be considerably different. The specificity of CSF cell changes in MS vs. other inflammatory CNS diseases such as neuromyelitis optica spectrum diseases remains unknown. The transcriptomics cohort of our study is also clearly under-powered (and is not designed) to address the known intra-disease heterogeneity of MS [282]. Our study, however, provides an essential reference point for future studies with this focus.

Specific T helper (Th) cell lineages have long been associated with MS-like pathology in rodents, while evidence in human MS is more ambiguous [283, 284]. Notably, blood T cells in our dataset showed some induction of Th17 cell-related signalling (*IL6R*) although most core Th17 transcriptional modules were not differentially expressed [285]. In contrast, CSF cells showed signs of Th1 cell-related signalling on the individual gene level (e.g. *IL12RB1, IL18RAP, IRF8*), by GSEA, and when using CSEA (**Figure3.9**). We also found an expansion of $CD4^+$ T cells with cytotoxic phenotype (CD4 Tc cluster #0) in MS vs. control patients in the CSF, but not in blood (**Figure3.6**). One of the marker genes in this cluster was EOMES (**Figure3.6(e)**) and notably EOMES is also a genetic risk locus for RRMS [286]. Previously, $EOMES^+CD4^+$ T cells were shown to increase in the blood of patients with secondary progressive (SP), albeit not relapsing-remitting (RR)MS and in late-stage EAE [287]. However, the previous study was underpowered to detect MS vs. control differences in the CSF (5 total samples). Another set of studies defined cytotoxic $CD4^+$ T cells by the lack of CD28 expression and these cells expanded in EAE [288] and in the blood of RRMS patients [289]. A quantification of such cells in the CSF is unavailable. Another recent study used CytOF to quantify 35 predefined chemokine and cytokine markers in blood cells from MS patients [290]. A population of $GM^-CSF^+CXCR4^+$ T helper cells expanded in the peripheral blood of RRMS patients and was enriched in the CSF compared to the blood as expected for a memory population [290]. But again, no MS vs. control comparison in the CSF was provided. This highlights the unique CSF vs. blood design of our study. It

remains to be tested to what extent *GM-CSF$^+$CXCR4$^+$* T helper cells represent a population with cytotoxic capacity and may overlap with our CD4 Tc cluster #0. Neither *CSF2* (encoding GM-CSF) nor *CXCR4* were detected in our dataset. In summary, although cell type definitions vary considerably between studies, CD4$^+$ T cells with cytotoxic potential may locally contribute to MS pathogenesis.

We also found that TFH cells enhanced B cell enrichment in the CNS in EAE and correlated with B lineage cell abundance in the CSF. We used a genetically more rigorous approach than a previous study [291] and our application of adoptive transfer EAE makes a contribution of TFR cells [257] unlikely, because effector cells do not convert to Tregs in EAE [292]. We and others [254] speculate that a pathological interaction between TFH cells and B cells in the CSF may locally drive CNS autoimmune reactions. In fact, B cell clones have long been known to, at least partially, expand in the CSF in MS [220, 293] together with migration from the periphery [218, 241]. Previous studies support both an influx of B cells that have matured (i.e. class-switched) in the periphery and a local maturation of B cells in the CSF [218, 241, 294, 295]. Our approach is in accordance with these studies and is unlikely to return false positives as it is unbiased and corrected for multiple-hypothesis testing. The relevance of B cells in MS is also supported by the efficacy of B cell-depleting therapies [221]. It will be exceptionally interesting to extend our single cell study design to MS patients receiving B cell-depleting treatments or in later disease stages. Our study provides an essential reference point for such future studies of human CSF and will likely facilitate understanding of diverse neurological diseases such as Parkinson's and Alzheimer's disease in the future.

3.5 Acknowledgements

This manuscript is written in collaboration with David Schafflick, Chenling A Xu, Maike Hartlehnert, Michael Cole, Andreas Schulte-Mecklenbeck, Tobias Lautwein, Jolien Wolbert, Michael Heming, Sven G Meuth, Tanja Kuhlmann, Catharina C Gross, Heinz Wiendl, Nir Yosef and Gerd Meyer Zu Horste. D.S., M.Hartlehnert, and T.L. performed experiments, M.C., M.Heming., C.A.X. and N.Y. performed computational analyses, A.S.-M. and C.G. processed CSF samples, T.K. performed histology, J.W. performed data and manuscript editing, S.G.M. and H.W. cosupervised the study, N.Y. and G.M.z.H. conceived and supervised the study, and wrote the manuscript. All authors critically revised the manuscript.

We thank Claudia Kemming, Anna-Lena Börsch, Maik Höfer, Gabriele Berens, and Kirsten Weiss for technical assistance. We thank Arpita Singhal for help in developing the CSEA pipeline. G.M.z.H. was supported by grants from the Deutsche Forschungsgemeinschaft (DFG, ME4050/4-1, ME4050/8-1), from the Gemeinnützige Hertie Stiftung, from the Innovative Medical Research (IMF) program of the University Münster, and from the Ministerium für Innovation, Wissenschaft und Forschung (MIWF) des Landes Nordrhein-

Westfalen. This project was funded in part by the DFG Sonderforschungsbereich Transregio 128 of the DFG (to S.G.M, A09 to H.W. and C.C.G., Z02 to H.W. and T.K.).

Chapter 4

Automated and Crowd-Sourced Annotation Cell Types using Tabula Sapiens

4.1 Introduction

Cell type annotation is a crucial task in scRNAseq analysis because it determines the quality of all downstream analysis including marker discovery and cell type abundance comparison. It is also highly time consuming and requires domain specific knowledge as well as familiarity with scRNAseq data. As scRNAseq becomes an increasingly standard lab technique, being able to generate automatic annotation will make accurate analysis a lot more accessible to the scientific community. We developed a probabilistic method scANVI to perform label transfer in scRNAseq datasets in Chapter 2. However there are several major challenges of automatic annotations. 1. Automatic annotation algorithms rely on either annotated reference datasets [14, 153, 65], or curated marker lists of known cell types [68]. If there is a new cell type unique to a newly sequenced dataset, the automatic annotation algorithms will not be able to generate accurate predictions. This issue is especially pronounced for rare cell types. Therefore, expert knowledge input is constantly needed for cell type annotation despite being very time consuming. 2. There is no gold standard ground truth in the cell type annotation task. Biology is complex and cell states might change due to environmental changes across different experiments. Most currently existing cell type annotations are done on the cluster level, and when cell states are varying continuously, even human experts might disagree on what a cell is. The lack of ground truth makes it difficult to pick one 'best method' across different applications. 3. cell types are not isolated from each other and are related by similar functions and origins. Thus there are many different levels of granularity in which cell types can be annotated. This often makes it difficult to generalize annotations across different datasets. Therefore, it is crucial for automatic cell type annotation pipelines to 1. be easily accessible and facilitate the process of manual annotation; 2. highlight the

disagreement between different methods and 3. focus on building an ontology and meaning of cell type relationships rather than having a single label per cell.

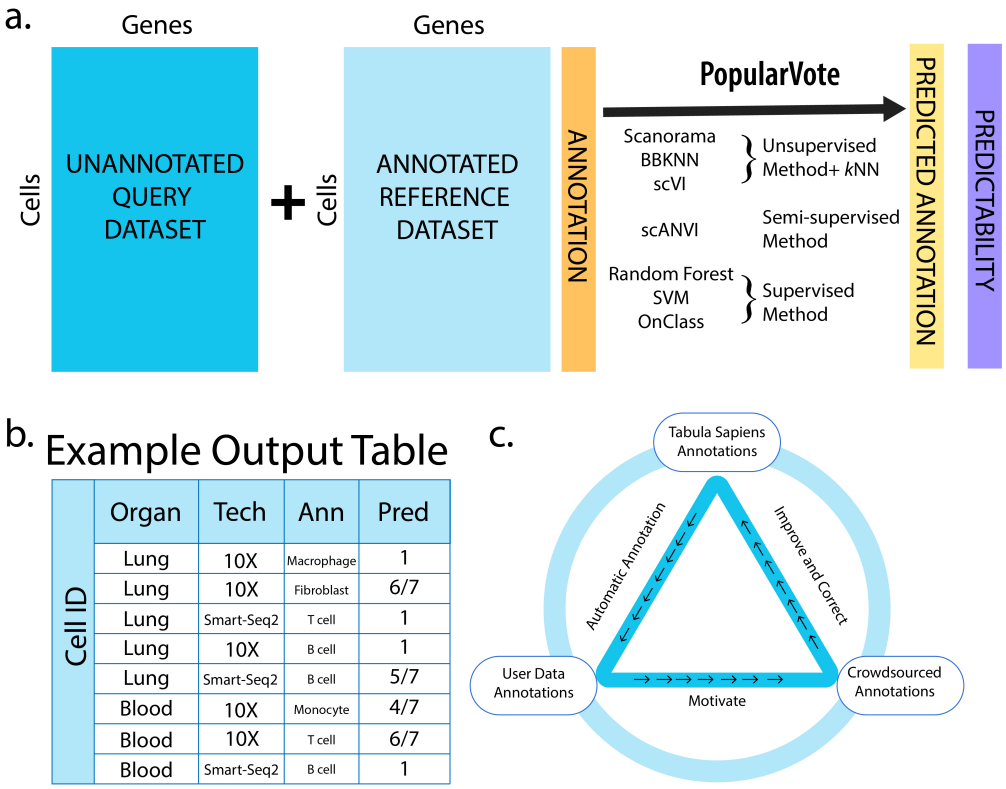
A good algorithm however is not sufficient for automatic annotation. Training data for the algorithm is just as important. A good reference dataset to train the automatic annotation algorithms should contain as many cell types as possible in a variety of environments. Tabula Sapiens is a highly collaborative effort to build a first-draft human cell atlas at a single cell level. It is built to be a public dataset that can be easily browsed from a hosted online portal for exploration of cell types and gene expression profiles in multiple human organs. The objective is to sequence two million cells from 25 organs from 8 donors and the dataset currently consists of 14 organs and 156,559 cells. Many organ expert groups have collaborated with the Chan-Zuckerberg Biohub to generate this dataset (manuscript in preparation, data portal [296]). The comprehensiveness of Tabula Sapiens in tissue and in cell type means that it is an ideal reference dataset when annotating new datasets. Tabula Sapiens also includes cells sampled from different organs and individual and sequenced using different technologies, allowing the annotation algorithms to control for environmental and technical variation.

Using Tabula Sapiens as a reference dataset, we designed an automatic annotation pipeline that takes in unannotated count matrices from scRNAseq experiments, transfer labels from an annotated dataset and generates predicted annotation with a predictability score indicating the confidence of the prediction (**Figure4.1**). We name our method PopularVote because we use a total of 7 automatic annotation methods to compute the majority vote prediction as well as a predictability score based on the agreement of different algorithms. The annotation methods we use include random forest (RF), support vector machine (SVM)), scANVI [117], onClass [297], and k nearest neighbours (kNN) after batch-correction using single cell harmonization methods (scVI [298], BBKNN [299], Scanorama [150]. These methods encompass supervised methods that are trained only on labelled data (RF, SVM, OnClass), unsupervised harmonization methods trained with data without label information (BBKNN, Scanorama, scVI) and a semi-supervised method trained with both labelled and unlabelled data(scANVI). OnClass is the only method currently available that can use the full ontology information for cell type annotation in single cells. BBKNN, Scanorama and OnClass are graph-based methods, scVI and scANVI are Bayesian Neural Network methods, SVM uses a linear model, and Random Forest uses a decision tree approach. The unsuperviwsed methods are coupled with kNN for generating label predictions. OnClass is the only method that uses ontology in its training procedure and is therefore able to predict cell types that does not exist in the reference dataset. PopularVote also produces a number of diagnostic plots to facilitate user evaluation of annotation results, and is evaluated based on cell ontology terms.

While this chapter mainly focused on the task of developing an automatic cell type annotation method, we also participated in the Tabula Sapiens project in the annotation of

the scRNAseq data and designing mechanisms in which the scientific community can access this data. PopularVote had played an important role in this process. We used PopularVote to generate initial annotations to guide manual annotations. Once the organs had been manually annotated, we used it to check annotation consistency and fill in annotations for cells that the tissue experts cannot confidently identify. Besides using it internally, we also built a public-facing interface for PopularVote so that biologists can use the Tabula Sapiens as a reference dataset for annotating their own dataset. At last, it is important for Tabula Sapiens as a resource to update as our knowledge of human cells are updated. Thus we build a mechanism for biologists to suggest new annotations to our data from the Tabula Sapiens Portal. The Crowd Sourcing annotations will be processed and Incorporated into the official versioned annotations. In summary, PopularVote have been used in many annotation tasks in the Tabula Sapiens project, and we hope that it will also become a resource for other projects.

Figure 4.1: PopularVote and Annotation Tasks



We show a diagram of how the automatic annotation pipeline facilitates the three components of annotation in the Tabula Sapiens project. (*a*) Diagram of the input and output of PopularVote. (*b*) Example output table. (*c*) There are three main annotation tasks in the Tabula Sapiens Project: Reference Annotation, User Data Annotation and Crowd-Sourcing Annotation.

4.2 Results

Design of Automatic Annotation Pipeline

The automatic annotation pipeline consists of two main component: the reference dataset and the annotation algorithms. We named our annotation pipeline PopularVote because we generate the final prediction using the majority vote of a number of common algorithms using Tabula Sapiens as reference data. Our pipeline can generate predicted labels for each cell in an un-annotated scRNAseq dataset as well as a predictability score. Predictability is defined as the fraction of automatic annotation algorithms which agree with the majority vote prediction. The predictability score indicates how much the user can trust the automatically generated labels.

A comprehensive and trust-worthy reference dataset is the basis of any prediction pipeline. We decided to use Tabula Sapiens as our reference dataset because it contains a large number of cells from multiple organs and therefore provides ample training data for computational models for cell type annotation. It also contains cells sequenced with two different technologies and multiple donors, allowing annotation algorithms to learn about batch effect so that technical variation is not confounded with cell type specific variation. This dataset is curated by a centralized bioinformatics team for data quality control but is annotated by a team of biologist with domain-specific knowledge. In addition all of the annotations are verified with cross validation. The high quality data and labels of Tabula Sapiens is what enables the annotation algorithms to generate reliable predictions. Another notable feature of the Tabula Sapiens dataset is that all cells are annotated within the OBO Foundry candidate ontology Cell Ontology (CL) [1] vocabulary. In addition to using the existing OBO cell ontology information, we contributed to the OBO cell ontology endeavor by suggesting additional terms including alveolar fibroblast, capillary aerocyte, immature enterocyte and mature enterocyte to be added. These are cell types that have been known in the literature that we also found in our dataset, but could not find the appropriate terms to annotate them within cell ontology. Other single cell functional genomics databases such as the Human Cell Atlas [10] and the Brain Initiative [142] also use the OBO cell ontology naming conventions.

For the annotation algorithms we picked a mix of standard machine learning methods (random forest (RF), support vector machine (SVM)), single cell specific annotation methods (scANVI, onClass), and k nearest neighbours (kNN) after batch-correction using single cell harmonization methods (scVI [298], BBKNN [299], Scanorama [150]). We chose these methods because they are shown to have good prediction accuracy in [71] and/or good harmonization performances [178]. Users can specify a subset of these methods if they find some of the methods to be not applicable to their dataset. The main advantage of this pipeline is that it is entirely implemented using the Jupyter Notebook and Docker container framework so the users do not need to set up the software environment. The accessibility allows the user to not be limited by a single method in a platform that they are familiar with. Com-

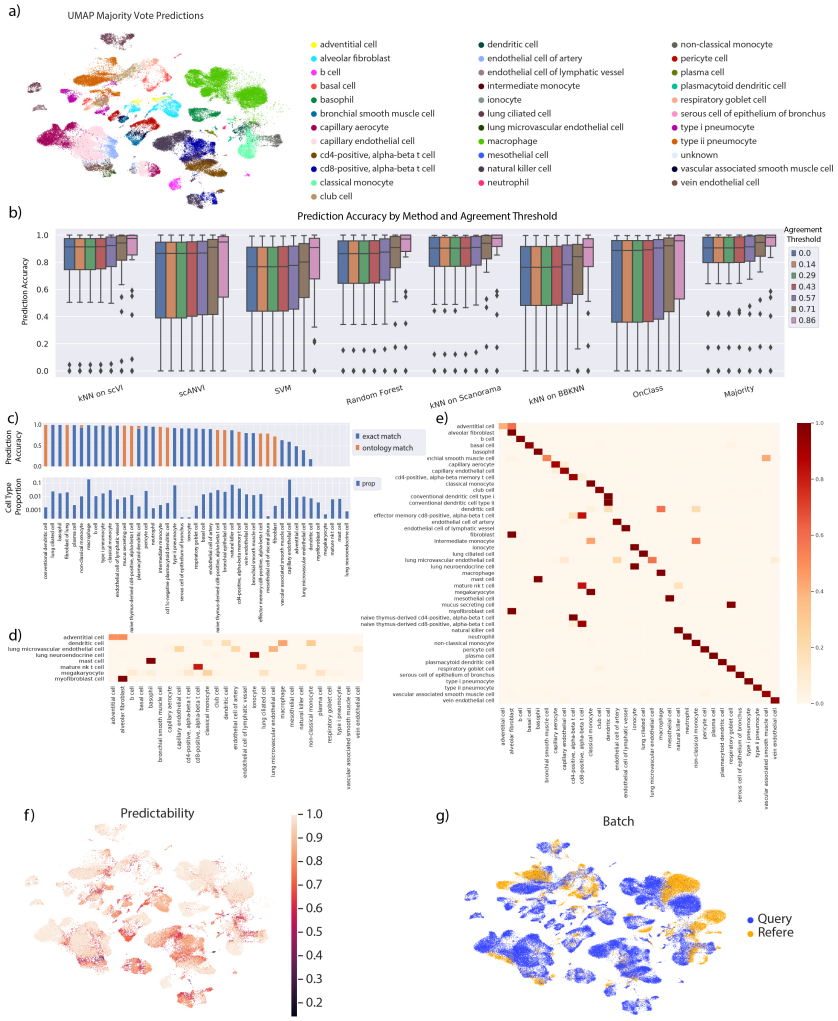
paring a variety of methods also allows the user to gain intuition about the agreement and disagreement of the methods, which we measure by our predictability statistic.

Benchmark

We evaluate the performance of PopularVote on an annotated lung dataset [300]. The Lung Atlas is annotated carefully to a high level of granularity and contains a large number of high quality cell sequences, making it suitable for the evaluation task. PopularVote achieves high accuracy on the lung benchmark dataset as shown in **Figure4.2**. We visualize the majority prediction of lung cell types in **Figure4.2(a)** and demonstrate that the predicted cell types correspond well to the cluster structure of the query dataset. PopularVote not only generates cell type prediction but also a predictability score based on the agreement of the different prediction algorithms. **Figure4.2(b)** shows the per cell type prediction accuracy of each prediction method of 7 annotation methods as well as the majority vote prediction. The error bar indicates how variable the accuracy is depending on the cell type. kNN on either scVI or Scanorama harmonized latent space and the majority vote prediction give the best average prediction accuracy. We also show that predictability is an important feature to determine the trust-worthiness of cell type predictions, as the prediction accuracy increases for every method when we limit the accuracy computation to only cells with high predictability score (**Figure4.2(b)**). We note that the variation of prediction accuracy is much higher between different cell types compared to among different methods.

To evaluate the quality of our predictions, we compute both the exact match and ontology match accuracy (blue and orange in the stacked bar plot in **Figure4.2(c)**, see Methods) of each method and show the results for the majority vote prediction as an example in **Figure4.2(c,d,e)**. Ontology match is a measure of accuracy that takes cell ontology into account. Intuitively, a prediction algorithm that predicts one cell type as another similar cell type has better performance than one that predicts it as something unrelated. Cell ontology encodes similarity in a tree structure, and we define a ontology match for a cell type as all of its offspring cell types and all of the cell types on its path to the root of the ontology. This measure is especially useful if a cell type label only exist in the query and not in the reference. For example mucus secreting cell and naïve CD4 T cells are both labels unique to the query dataset but their ontology match accuracy is close to 1. This is because they are predicted as a child term (lung goblet cell) or a parent term (CD4 T cell) that does exist in the reference dataset.

Figure 4.2: Lung Benchmark Results



(a) UMAP projection of the scVI latent space, colored by the majority vote prediction. (b) Per cell type prediction accuracy of different prediction algorithms and increasing predictability filtering threshold. The error bar represent variation in accuracy among different cell types (25 and 75 percentile). (c) Barplot of prediction accuracy (both exact match and hierarchical accuracy and of cell type frequency. (d) Confusion matrix heatmap for cell types with lowest prediction accuracy. Rows represent ground truth cell types and columns the majority vote predictions. (e) Confusing matrix heatmap where rows represent ground truth cell types and columns are the majority vote predictions. (f) UMAP projection of the scVI latent space, colored by predictability score varying from $0-1$. (g) UMAP projection of the scVI latent space, colored by batch identity of each cell.

A few of the poorly predicted cell types include lung neuroendocrine cells, mature natural killer T cells, megakarocytes, mast cells and myofibroblasts. All of these cell types are unique to the query dataset and not sampled or labeled in the Tabula Sapiens lung reference data. However these cell types are often mis-annotated as the most similar cell types that are present in the reference. These cell types are related to the true cell type but do not constitute a ontology match because they are not found along a single lineage on the ontology tree. For example mature natural killer T cell are annotated as CD8 T cell and natural killer cell, and dendritic cells are annotated as other myeloid immune cells such as macrophage and non-classical monocyte. These disagreemnts are highlighted in **Figure4.2(d)**. Details of all predictions can be observed in the prediction heatmap (**Figure4.2(e)**) where rows correspond to ground truth labels in the dataset and columns correspond to the predicted labels and the commonly mis-annotated cell types are highlighted in **Figure4.2(d)**. By comparing automatic annotation and manual annotations, the users of PopularVote can focus their efforts on easily confounded cell types.

Finally we show the predictability score (**Figure4.2(f)**) on the UMAP projection of query dataset and the batch mixing (**Figure4.2(g)**) of the UMAP projection of both datasets. We notice a higher predictability score in regions of the UMAP where the query and reference dataset is well mixed.

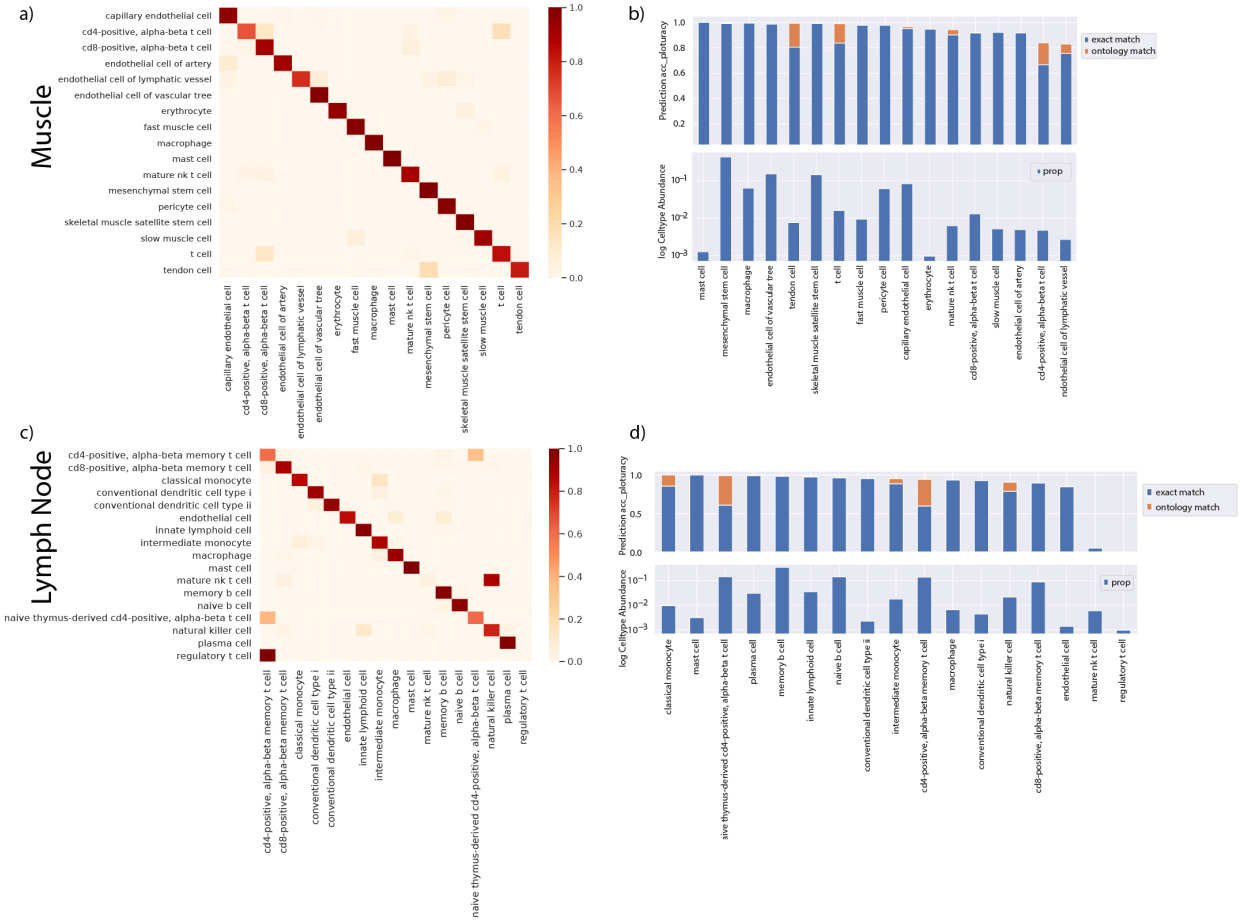
Using PopularVote to Evaluate the Consistency of Manual Annotations

First we use PopularVote in the annotation of Tabula Sapiens reference data. Tabula Sapiens is annotated by a large number of experts and will in the future be annotated collectively through our crowd-sourcing platform, we would like to perform quality control (QC) on the manual annotations. We realized that PopularVote can be useful for automating the QC process. Briefly, we hide 20% of the cell type labels from a manually annotated dataset, and use the other 80% of the labelled cells to generate labels for the first 20% of cells. We then repeat this procedure 5 times to generate a predicted label for every cell in the dataset. We can then compare the prediction with the original manual labels of those cells. If a cell type is difficult to classify, we would expect the automatic annotation algorithms to make mistakes more often. In this process we also generate a predictability score for each cell. If the cells annotated as the same cell type are transcriptionally distinct from all other cell types, then the classification algorithms should be able to easily classify those cells and agree with each other. In other words, when the annotation is consistent with the data, our cross validation analysis should return high predictability scores and high accuracy. It should be noted that consistency is not the same as accuracy: if one cell type is substituted as another one, the consistency score will remain the same. The purpose of the cross validation study is to bring attention to cell types that are potentially mis-annotated. One caveat is that for

cell types that are functionally distinct but have high transcriptional similarity, the cross validation consistency might be low even when the original labels are accurate. This will require manual examination to distinguish from true mis-classification errors. However we can argue that in this case, since the cell types are indeed easy to confound, the PopularVote cross validation does achieve what it is designed to do. We applied PopularVote to all organs in Tabula Sapiens Pilot 1 and 2 and generated predictability score for all cells by running a 5-fold cross validation.

In **Figure4.3** we show the output of the cross validation pipeline for Muscle and Lymph Node. The majority vote prediction in muscle agree with the manual annotation, showing that the manual annotation is highly self-consistent (**Figure4.3(a,b)**). In Lymph Node the same is true for most cell types except for the T cell subsets. In **Figure4.3(c)** we observe a confusion between CD4 memory T cell and CD4 naïve T cell, mature NK T cell and natural killer cell, CD4 memory T cell and regulatory T cells. CD4 T cells is known to not have clear clustering boundaries from transcriptional data, and the cross validation results indicates that the boundaries proposed in the manual annotation does not correspond well to the transcriptional similarity in the data. This analysis is especially useful as the crowd-sourced data from the Tabula Sapiens Portal accumulates. It is not feasible to manually examine each manual annotation input, and this analysis will give a first pass idea of the quality of the manual annotation as well as highlight cell types that might have been mis-annotated.

Figure 4.3: Using PopularVote for Manual Annotation Consistency Check



(*a*) Confusion matrix of manual annotation (rows) and majority vote prediction by cross validation (columns) in Muscle. (*b*) Cell type prediction accuracy and abundance bar-plot in Muscle. (*c*) Confusion matrix of manual annotation (rows) and majority vote prediction by cross validation (columns) in Lymph Node. (*d*) Cell type prediction accuracy and abundance bar-plot in Muscle.

Using Popular Vote in the Wild to Annotate New Samples

After we demonstrated the use of PopularVote in Tabula Sapiens, we decided to make it available for general public for their annotation tasks. PopularVote can be accessed through both Google Colab Notebooks and Docker Containers. Both Colab and Docker share the same backend code, but have different advantages. Google Colab requires less computational set up and can make use of the Google cloud GPU for users who are not familar with programming and deep learning, or do not have local access to GPUs. The use of GPUs speeds up PopularVote significantly for training scVI, scANVI and OnClass. However Google Colab has limited RAM and needs to be relaunched when disconnected, thus is not suitable for large datasets. Docker containers require local or cloud computational resource set up from the user, but also comes with all required packages pre-installed. It is more suitable for users with large datasets. Both implementations can be customized easily by editing the PopularVote backend Python code.

PopularVote outputs the predictions of 7 prediction algorithms (scVI+kNN, BBKNN+kNN, Scanoram+kNN, SVM, RF, OnClass and scANVI), the aggregate majority vote prediction and the predictability score. The users can explore these results in an automatically generated data object. It can be used for downtream analysis in Scanpy [35] or visualized in CellxGene [301]. CellxGene allows user to interactively visualize gene expression, meta data associated with each cell on a 2D latent space of choice. We also generate evaluation figures focused on the overall and per cell-type agreement and batch mixing in **Figure4.4**.
Unlike in the benchmark example, new datasets do not come with any manual annotation, but we can use other unsupervised measures to evaluate the performance of annotation. Namely the quality of data harmonization can be used to highlight mismatch between the query and reference dataset, and the agreement between multiple prediction algorithms can be used to highlight cell types that are difficult to predict. These quality measures are automatically generated at the end of the prediction pipeline. They can help users identify both outlier methods and outlier cell types. We use the lung atlas dataset as an example for how these measures can be useful.

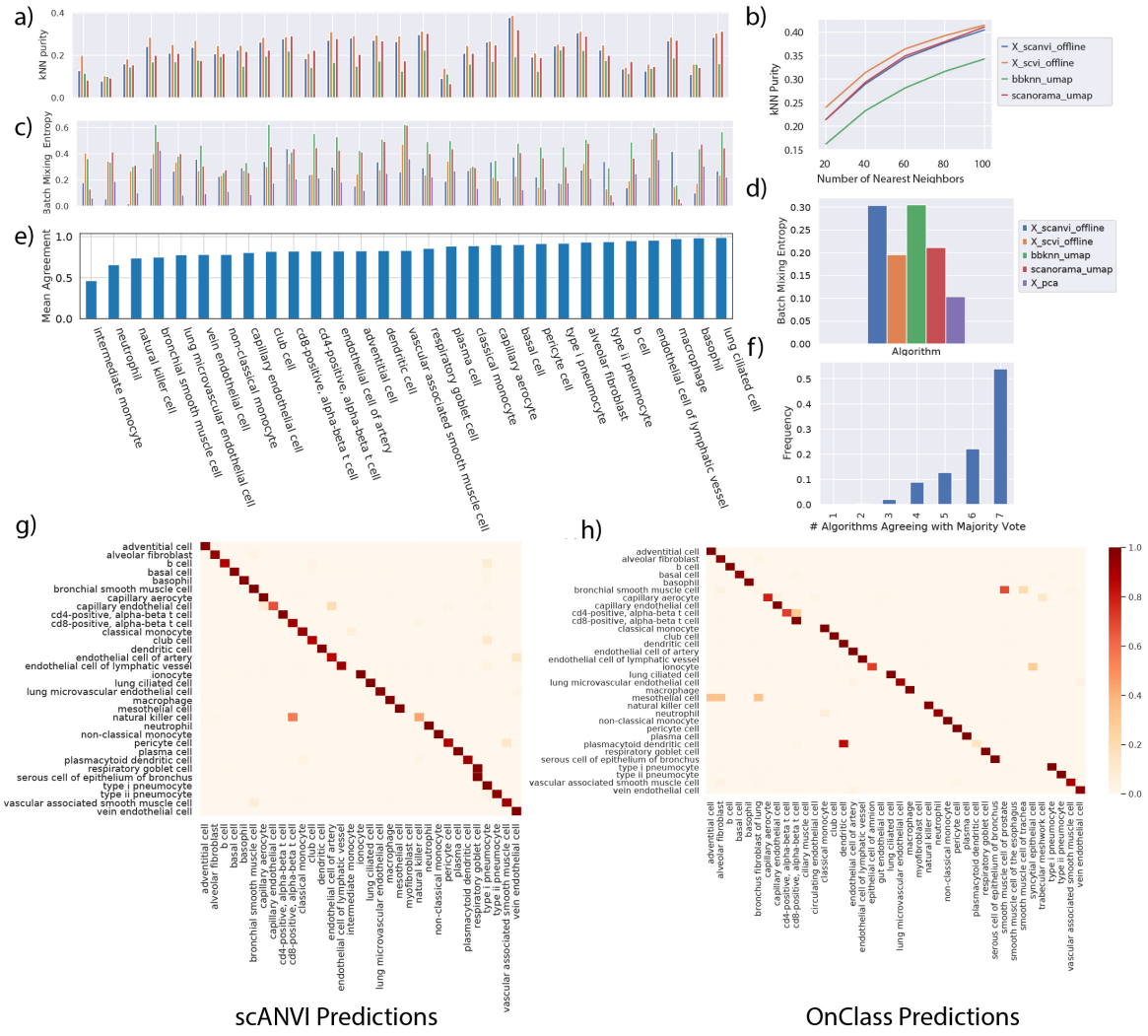
First, we explore how to evaluate the prediction performance using the predictability score in the absence of ground truth. The agreement between individual algorithms and the majority vote prediction can be used as basis to exclude outlier prediction methods or to highlight difficult to predict cell types. We show the heatmap comparing the individual prediction algorithms (scANVI and OnClass) compared to the majority vote predictions (**Figure4.4(g,h)**). scANVI has high concordance with the majority vote prediction except in natural killer cell (predicted as CD8-positive, alpha-beta T cell), and respiratory goblet cells (predicted as respiratory goblet cell). These disagreement fall within cell types that are transcriptionally and functionally similar. OnClass on the other hand, does not have these two disagreement with the majority vote predictions. One distinct feature of OnClass is its ability to predict cell types that do not exist in the original reference. However these

predictions needs more scrutiny than cell types that are manually annotated as can be shown in (**Figure4.4(h)**). Bronchial smooth muscle cells are predicted as several other types of smooth muscle cell from other organs, and these errors can be easily caught by looking at the confusion matrix heatmap. Tissue experts can gain insights from the disagreement to identify cell types that are easily confounded. This can facilitate further improvement of the cell type annotations.

In **Figure4.4(e)**, we show the percentage of cells that agree with majority vote prediction, sorted from low to high. Users can identify cell types that have high chances of disagreement between algorithms, such as intermediate monocyte in the lung benchmark dataset. From the heatmap in **Figure4.2(e)** we know that cells annotated as intermediate monocyte are predicted as classical monocyte in around half of the time. Intermediate monocytes develops from classical monocytes and are transcriptionally similar. This indicates that agreement is also a good indicator of mis-annotations when ground truth labels are not available. We can also evaluate the overall performance in a dataset by looking at the distribution of algorithm agreement over all the cells (**Figure4.4(f)**). We see that there are very few cells where none of the algorithm agree with each other, and most cells are agreed upon by all seven algorithms (**Figure4.4**).

Secondly accuracy of the annotation depends on harmonization. If the datasets do not share the same cell types, or the harmonization methods are not sufficient to correct batch effect, the annotation algorithms will also have difficulty transferring labels between different datasets. We can use the harmonization quality as a basis to exclude methods that over or under-correct batch effects. We use kNN Purity and Batch Mixing Entropy (see Chapter 2) to evaluate unsupervised data harmonization performance.These two metrics measure two opposing yet necessary aspects of data harmonization: kNN purity measures the conservation of nearest neighbor structure before and after harmonization, while batch mixing entropy measure how well cells from both datasets mixes in local neighborhood. In the ideal case, cells that are biologically similar from both query and reference dataset will cluster together in the harmonized latent space, resulting in both high kNN Purity and high Batch Mixing Entropy. However batch effect could be under or over corrected, and there could be composition or cell state differences between the two datasets.

Figure 4.4: Additional User Report Figures



(*a*) Per Cell type *k*NN purity bar plot colored by each algorithm. (*b*) Average *k*NN purity line plot with respect to the number of nearest neighbors, colored by each algorithm. (*c*) Per Cell type Batch Mixing Entropy bar-plot colored by each algorithm. (*d*) Average Batch Mixing Entropy bar-plot with respect to the number of nearest neighbors, colored by each algorithm. (*e*) Average per cell type predictability score. (*f*) Distribution of predictability score across all cells. (*g*) Confusion matrix comparing the majority vote predictions (rows) to the scANVI predictions (columns). (*h*) Confusion matrix comparing the majority vote predictions (rows) to the OnClass predictions (columns)

We look at the variation in kNN Purity and Batch Mixing Entropy by different cell types and harmonization methods to highlight cases when our pipeline might fail. In our example dataset, all cell types have reasonably high kNN Purity values (the expected value at random is around 0.066) (**Figure4.4(a)**). The per cell type Batch Mixing Entropy is variable across different methods (**Figure4.4(c)**). BBKNN has the highest Batch Mixing Entropy score over the entire dataset (**Figure4.4(d)**) but has the lowest score in kNN Purity, suggesting that it has the tendency to over-correct batch effect. scANVI is not the best performing method in either kNN Purity or Batch Mixing Entropy but achieve a good balance. Therefore we conclude that the harmonization methods have different trade-offs but all have reasonable performances, and do not exclude any method in this example.

We can also examine the harmonization results on a per cell type basis. Cell types with low Batch Mixing Entropy means that the cells from the query and reference dataset are separated in the latent space. Some cell types are separated before harmonization (neutrophil and natural killer cell) but all harmonization algorithms increase the batch mixing entropy, indicating that the technical differences between the query and reference dataset in these cell types are accounted for. Cell types such as the macrophage have low batch mixing entropy except in scANVI latent space which was trained with cell type labels, indicating that although the cells of this cell type share similar transcriptional profile there might be significant biological differences. We manually examined the marker expression of the macrophage cells from both the query and reference dataset, and found that they both express essential macrophage markers. Although in this specific case the degree of mixing does not affect prediction accuracy or predictability, the ability of PopularVote to highlight such populations is still useful.

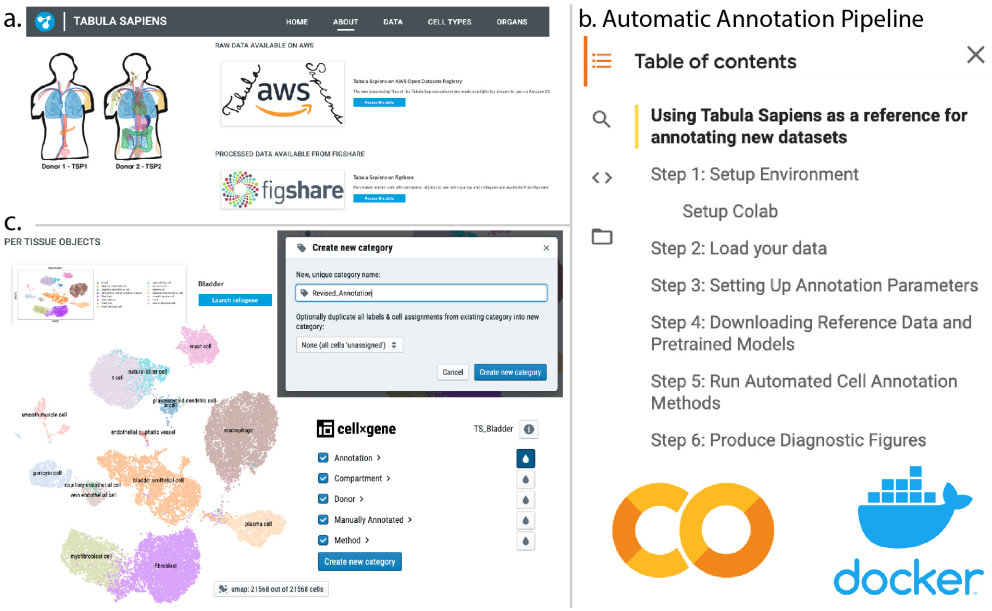
Tabula Sapiens Portal Software of PopularVote

Tabula Sapiens data is hosted on a public data portal to facilitate access to this data for the scientific community with or without prior programming experiences. **Figure4.5** shows the many ways the Tabula Sapiens Dataset can be accessed. Users can download either the raw data through Amazon Web Services (AWS) or the processed data from Figshare (**Figure4.5(a)**). The final data release on the data portal contains in the data layers a normalized and logged gene count matrix, a raw count matrix, and another count matrix after we used decontX [37] to decontaminate the original count matrix for ambient RNA. The data object contains the UMAP projection of the scVI latent space for visualization. For each cell, the data object contains its organ and anatomical information, the technology the cell was sequence with, its annotation and wheter the annotation was derived manually by a tissue expert, or automatically using PopularVote's majority vote prediction. Users can query their own data against the Tabula Sapiens reference using PopularVote without painstakingly setting up computational environment using either Google CoLab or Docker Containers (**Figure4.5(b)**). The annotation pipeline is well annotated and is intended to require minimal programming experiences. However, the annotation pipeline is hosted on

Github and open source so that power users can easily change parameters of the algorithms used in PopularVote or add new methods. Users who are not interested in processing the data themselves can directly browse the dataset using a hosted session of CellxGene (**Figure4.5(d)**).

For future development, we would like to allow users to not only access the data that we have generated but also to revise our annotations. Currently, users can create a new annotation category tied to their user ID on the hosted sessions and view both their own annotations and existing meta data by coloring cells by one of the meta-data columns on CellxGene. The annotations is then saved and collected for crowd-sourced annotations. There are certainly challenges associated with crowd-source annotations: How do we identify high quality annotations and low quality annotations? How do we incorporate new information while keeping our data tractable? How do we motivate and credit the contributors? Part of the solution is that we can then use PopularVote to scrutinize the crowd-sourced annotation as mentioned in the "Using PopularVote to Evaluate the Consistency of Manual Annotations" section. We can prioritize crowd-sourced annotations that have high consistency score, are mostly similar to our original annotation but add new information such as a new cell type, or higher resolution in the cell ontology tree. The Tabula Sapiens annotation can be updated regularly with versioned history with all changes that are accepted during a certain time period, to keep the changes tractable.

Figure 4.5: Tabula Sapiens Portal and Crowd-Sourcing



We show the web interfaces for accessing the Tabula Sapiens data using multiple approaches. (*a*) Tabula Sapiens data can be downloaded from the online portal through Amazon Web Services(AWS) or Figshare. (*b*) Querying Tabula Sapiens using unannotated data with Google Colab and Docker Containers. (*c*) CellxGene sessions are also hosted online and can be accessed from the Tabula Sapiens Portal.

4.3 Methods

Preprocessing and Manual Annotation of Tabula Sapiens Data

Each organ is sequenced by both 10X and Smart-seq2 technology and could have data from multiple donors. We first harmonize the multiple batches of data to generate a harmonized visualization of the cells using scVI and shared in a data object compatible with both Scanpy and CellxGene [301]. CellxGene is a data visualization tool that allows user to interactively explore any scRNAseq dataset in the Scanpy standard format. The data object contains three main component: gene count data, cell-wise metadata, and gene-wise metadata. CellxGene allow the user to color cells by any cell metadata such as donor, technology and cell type annotations. The user can also select cells based on any meta data features, or using a lasso tool. The tissue experts on the Tabula Sapiens project visualize the data and marker gene expression on CellxGene to generate manual annotation. We then generate a data object containing the new annotation, perform marker discoveries and check the consistency of annotations with current cell ontology terms. In this process we noticed several data quality issues such as ambient RNA and missing cell types due to over-strict filtering scheme. After correcting for such issues we updated the data object and performed multiple rounds manual annotation. and generated a final object containing all of the organs.

Crowd Sourcing Annotation Pipeline

We created a pathway for generating crowd-sourced annotation using a web-hosted version of CellxGene. Users can go on the Tabula Sapiens Portal and visualize the data on CellxGene, create an annotation category. CellxGene automatically generate a .csv file that then can be checked for consistency and incorperate into the next version of official reference annotation. The three task of annoations are connected through the use of a common annotation pipeline (PopularVote). In addition, the user data annotation function will motivate users to look closely at the reference dataset, motivating them to add new terms and correct annotation errors through the crowd-sourcing platform, which will eventually improve the reference annotation.

Implementation of PopularVote

All of the algorithms used in PopularVote are from published sources and here we explain the hyper-parameter choices and the exact implementation we used. For all classic machine learning algorithms we use the implementations from scikit-learn [302] release 0.22.2.post1 with the following parameters: KNeighborsClassifier(n_neighbors=15, weights="uniform"), LinearSVC(max_iter=1000),RandomForestClassifier. We use the Scanpy [35] version >= 1.6 implementation for BBKNN, and Scanorama version >= 1.7. For scVI and scANVI we use

the scvi-tools release 0.91.1. Onclass was still in development when PopularVote was developed and we used the 1c4c3f332ae3effceae46c1bb82a6104e92901cd commit for our pipeline. One challenge in applying machine learning algorithms on scRNAseq dataset is the unbalanced nature of the training data. Unbalanced training data has been a topic of research in the machine learning community and the three options for alleviating its effect on the classifier are 1. cost-sensitive learning 2. down-sampling and 3. over-sampling [303]. Unbalanced training set is a common issue in most scRNAseq dataset especially because there are many rare cell types that carry out important biological function. We choose to use down-sampling of our data and set an upper limit of number of labelled cells to be 100. If a cell type contains more than 100 cells then only 100 of them remained labelled in the training set. The down-sampling increases the average per-cell-type prediction accuracy as well as reduces training time without compromising the accuracy of the common cell types because most of the cell types that have lower prediction accuracy are the rare cell types (**Figure4.2**).

Ontology match Accuracy

In order to take cell type ontology into account while computing accuracy, we defined a new statistic called ontology match accuracy. We base our ontology on the OBO Foundry candidate ontology Cell Ontology (CL) [1], and included modifications based on organ expert feedback. Each cell type is represented by a class with a unique id, and are related to its parents or children term by a directional edge. This measure is particularly useful when different datasets are annotated at different levels of granularity. For example, a cell can be annotated as T cell if the annotator is not familiar with different T cell subsets and states, but a more T cell-focused researcher might annotate it as naive CD4 T cell. Neither annotation is wrong but an exact match algorithm will count the two versions of annotations as mismatched. Cell Ontology allows us to define for each cell type a set of acceptable matches: all of its offspring and ancestors. When comparing ground truth and predictions, or different versions of annotations, we compute a hierarchical accuracy that is either equal or greater than the simple exact match accuracy because it better reflects the biological correspondence between different terms. A extension of this measure would be to assign different weights to matches at different distances on the tree, but since the OBO Cell Ontology does not provide edge length for the connections, and we have noticed a bias of having more levels in the tree structure in more studied systems we have decided to evaluate annotation consistency based on a combination of exact and ontology match, reasoning that the more detailed weighted accuracy will be upper and lower bounded by these two measures of accuracy.

4.4 Conclusion

We present in this chapter an automatic annotation pipeline PopularVote that was devised as part of the Tabula Sapiens project. It was essential for the annotation of the large amount

of data in this project, and will serve as a community resource to facilitate the querying of the reference dataset. PopularVote provides a robust automatic annotation framework using multiple algorithms. It also provides an estimate of automatic annotation confidence by computing the agreement, highlighting cells that might have been mis-annotated for expert revision. The accuracy of any automatic annotation pipeline depends on the quality of the reference dataset and Tabula Sapiens is a comprehensive dataset that is ideal for serving this purpose. We show with a published dataset that the predictions and diagnostic figures of PopularVote is accurate and can help biologists identify mis-annotations. The PopularVote scheme can be easily expanded to include more algorithms, and can also incorporate other wisdom of the crowd aggregation methods other than the majority vote approach. We demonstrate the use of PopularVote through Google Colab and Docker Containers. We also set up a data portal for the scientific community to generate crowd-sourced annotation for Tabula Sapiens data. This will facilitate feedback in order to improve the Tabula Sapiens annotation to reflect the most up-to-date understanding of cell types. PopularVote will be used in this process to evaluate the consistency of the crowd-sourced annotations. In conclusion, we show the use of PopularVote in multiple application scenarios in the Tabula Sapiens project, and how it could serve as a useful tool for the general public.

4.5 Acknowledgement

This chapter is written in collaboration with Angela Pisco, Galen Xing, Aaron McGeever, supervised by Nir Yosef. AP and NY and I conceived of the project. AP performed the initial sequencing processing. GX, AM and I implemented the method. I performed the analysis and wrote the manuscript. We'd like to thank the rest of the Tabula Sapiens Consortium for generating the data, performing the manual annotation, and help in understanding the data.

Conclusion

In my dissertation I showcase a number of tools (SymSim in Chapter 1, scVI and scANVI in Chapter 2, PopularVote in Chapter 4) that my co-authors and I had developed for the analysis of scRNAseq data. SymSim is a simulation method that can be used for benchmarking inference methods. scVI is a Bayesian model that accounts for statistical uncertainty in scRNAseq data using Variational Inference, and can be applied to many tasks such as dimensionality reduction, data harmonization and differential gene expression. scANVI is a semi-supervised extension of scVI that uses the same framework for the cell type annotation task. PopularVote is an automatic cell type annotation pipeline that uses wisdom of the crowd to generate cell type annotation and predictability estimate for unannotated scRNAseq datasets. These methods are applied in experimental studies. We used data harmonization to enable to comparison of immune cells across tissues and donors to understand the cellular changes associated with Multiple Sclerosis. We also used data harmonization and automatic cell type annotation in Tabula Sapiens to build a cell atlas for the human body. scRNAseq is a rapidly growing technology and these methods have shown to be useful in real world applications. In the near future, the computational biology community will be creating an easily accessible ecosystem of single cell analysis methods. scVI and scANVI are already part of the package scVI tools that also include other statistical methods. PopularVote is hosted on Google CoLab and Docker Container along with the Tabula Sapiens data for the automatic annotation task. The data in this thesis including the Multiple Sclerosis immune profile and the Tabula Sapiens atlas has also been shared with the public, and has already been used by other studies or annotation pipelines. The open code and data I hope will contribute to future scRNAseq studies to speed up the discovery process.

Bibliography

[1] Alexander D Diehl et al. "The Cell Ontology 2016: enhanced content, modularization, and ontology interoperability". In: *Journal of biomedical semantics* 7.1 (2016), pp. 1–10.

[2] Fuchou Tang et al. "mRNA-Seq whole-transcriptome analysis of a single cell". In: *Nature methods* 6.5 (2009), pp. 377–382.

[3] Martin Bengtsson et al. "Quantification of mRNA in single cells and modelling of RT-qPCR induced noise". In: *BMC molecular biology* 9.1 (2008), pp. 1–11.

[4] Arjun Raj et al. "Stochastic mRNA synthesis in mammalian cells". en. In: *PLoS Biol.* 4.10 (Oct. 2006), e309.

[5] Kazuki Kurimoto et al. "An improved single-cell cDNA amplification method for efficient high-density oligonucleotide microarray analysis". In: *Nucleic acids research* 34.5 (2006), e42–e42.

[6] Aleksandra A Kolodziejczyk et al. "The technology and biology of single-cell RNA sequencing". In: *Molecular cell* 58.4 (2015), pp. 610–620.

[7] Saiful Islam et al. "Quantitative single-cell RNA-seq with unique molecular identifiers". en. In: *Nat. Methods* 11.2 (Feb. 2014), pp. 163–166.

[8] Tiago Lubiana and Helder I Nakaya. "Towards a pragmatic definition of cell type". In: *Authorea Preprints* (2021).

[9] Eran A Mukamel and John Ngai. "Perspectives on defining cell types in the brain". In: *Current opinion in neurobiology* 56 (2019), pp. 61–68.

[10] Aviv Regev et al. "The human cell atlas". In: *eLife* 6 (Dec. 2017), e27041.

[11] Conrad Hal Waddington. *The strategy of the genes*. Routledge, 2014.

[12] Slava Epelman, Kory J Lavine, and Gwendalyn J Randolph. "Origin and functions of tissue macrophages". In: *Immunity* 41.1 (2014), pp. 21–35.

[13] Allon Wagner et al. "In silico modeling of metabolic state in single Th17 cells reveals novel regulators of inflammation and autoimmunity". In: *bioRxiv* (2020).

[14] Chenling Xu et al. "Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models". In: *Molecular systems biology* 17.1 (2021), e9620.

[15] Samantha A Morris. "The evolving concept of cell identity in the single cell era". In: *Development* 146.12 (2019).

[16] Katherine M McKinnon. "Flow cytometry: an overview". In: *Current protocols in immunology* 120.1 (2018), pp. 5–1.

[17] Karthik A Jagadeesh et al. "Identifying disease-critical cell types and cellular processes across the human body by integration of single-cell profiles and human genetics". In: *bioRxiv* (2021).

[18] Quy H Nguyen et al. "Experimental considerations for single-cell RNA sequencing approaches". In: *Frontiers in cell and developmental biology* 6 (2018), p. 108.

[19] Malte Luecken and Fabian J Theis. "Current best practices in single-cell RNA-seq analysis: a tutorial". In: (2019).

[20] Daniel Ramsköld et al. "Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells". In: *Nature biotechnology* 30.8 (2012), pp. 777–782.

[21] Alex K Shalek et al. "Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells". In: *Nature* 498.7453 (2013), pp. 236–240.

[22] Yurong Xin et al. "Use of the Fluidigm C1 platform for RNA sequencing of single mouse pancreatic islet cells". In: *Proc Natl Acad Sci* 113.12 (2016), pp. 3293–3298.

[23] Grace X Y Zheng et al. "Massively parallel digital transcriptional profiling of single cells". en. In: *Nat. Commun.* 8 (Jan. 2017), p. 14049.

[24] Allon M Klein et al. "Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells". In: *Cell* 161.5 (2015), pp. 1187–1201.

[25] E Z Macosko et al. "Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets". eng. In: *Cell* 161 (May 2015), pp. 1202–1214.

[26] X Fan et al. "Circulating CCR7+ICOS+ Memory T Follicular Helper Cells in Patients with Multiple Sclerosis". eng. In: *PLoS One* 10 (2015), e0134523.

[27] Rashel V Grindberg et al. "RNA-sequencing from single nuclei". In: *Proceedings of the National Academy of Sciences* 110.49 (2013), pp. 19802–19807.

[28] Naomi Habib et al. "Massively parallel single-nucleus RNA-seq with DroNc-seq". In: *Nature methods* 14.10 (2017), pp. 955–958.

[29] Saiful Islam et al. "Quantitative single-cell RNA-seq with unique molecular identifiers". In: *Nature Methods* 11.2 (2014), pp. 163–166. ISSN: 1548-7091. DOI: 10.1038/nmeth.2772.

[30] Simone Picelli et al. "Smart-seq2 for sensitive full-length transcriptome profiling in single cells". In: *Nature Methods* 10.11 (2013), pp. 1096–1098. ISSN: 1548-7091. DOI: 10.1038/nmeth.2639.

[31] Tamar Hashimshony et al. "CEL-Seq: single-cell RNA-Seq by multiplexed linear amplification". en. In: *Cell Rep.* 2.3 (Sept. 2012), pp. 666–673.

[32] Itay Tirosh et al. "Single-cell RNA-seq supports a developmental hierarchy in human oligodendroglioma". en. In: *Nature* 539.7628 (Nov. 2016), pp. 309–313.

[33] Andrew Butler et al. "Integrating single-cell transcriptomic data across different conditions, technologies, and species". In: *Nat Biotechnol* 36 (Apr. 2018), p. 411.

[34] Tim Stuart et al. "Comprehensive Integration of Single-Cell Data". In: *Cell* 177.7 (2019), pp. 1888–1902.

[35] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. "SCANPY: large-scale single-cell gene expression data analysis". In: *Genome biology* 19.1 (2018), pp. 1–5.

[36] Adam Gayoso et al. "scvi-tools: a library for deep probabilistic analysis of single-cell omics data". In: *bioRxiv* (2021).

[37] Shiyi Yang et al. "Decontamination of ambient RNA in single-cell RNA-seq with DecontX". In: *Genome biology* 21.1 (2020), pp. 1–15.

[38] Matthew D Young and Sam Behjati. "SoupX removes ambient RNA contamination from droplet-based single-cell RNA sequencing data". In: *GigaScience* 9.12 (2020), giaa151.

[39] Christopher S McGinnis, Lyndsay M Murrow, and Zev J Gartner. "DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors". In: *Cell systems* 8.4 (2019), pp. 329–337.

[40] Samuel L Wolock, Romain Lopez, and Allon M Klein. "Scrublet: computational identification of cell doublets in single-cell transcriptomic data". In: *Cell systems* 8.4 (2019), pp. 281–291.

[41] Erica AK DePasquale et al. "DoubletDecon: deconvoluting doublets from single-cell RNA-sequencing data". In: *Cell reports* 29.6 (2019), pp. 1718–1727.

[42] Jonathan A Griffiths et al. "Detection and removal of barcode swapping in single-cell RNA-seq data". In: *Nature communications* 9.1 (2018), pp. 1–6.

[43] Romain Lopez et al. "Deep Generative Modeling for Single-cell Transcriptomics". In: *Nat Methods* 15.12 (2018), pp. 1053–1058.

[44] Rahul Satija et al. "Spatial reconstruction of single-cell gene expression data". In: *Nature biotechnology* 33.5 (2015), pp. 495–502.

[45] Grace XY Zheng et al. "Massively parallel digital transcriptional profiling of single cells". In: *Nat Commun* 8 (2017), p. 14049.

[46] Tim Stuart et al. "Comprehensive integration of single-cell data". In: *Cell* 177.7 (2019), pp. 1888–1902.

[47] Angela Oliveira Pisco et al. "A single cell transcriptomic atlas characterizes aging tissues in the mouse". In: *BioRxiv* (2019), p. 661728.

[48] Amit Zeisel et al. "Molecular Architecture of the Mouse Nervous System". In: 174.4 (2018). ISSN: 0092-8674. DOI: 10.1016/j.cell.2018.06.021.

[49] Evan Der et al. “Single cell RNA sequencing to dissect the molecular heterogeneity in lupus nephritis”. en. In: *JCI Insight* 2.9 (May 2017).

[50] Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. “Challenges in unsupervised clustering of single-cell RNA-seq data”. In: *Nature Reviews Genetics* 20.5 (2019), pp. 273–282.

[51] Valentine Svensson. “Droplet scRNA-seq is not zero-inflated”. In: *bioRxiv* (2019), p. 582064.

[52] Aaron T L. Lun, Karsten Bach, and John C Marioni. “Pooling across cells to normalize single-cell RNA sequencing data with many zero counts”. In: *Genome Biol.* 17 (2016), p. 75.

[53] David DeTomaso et al. “Functional interpretation of single cell similarity maps”. In: *Nat Commun* 10.1 (2019), pp. 1–11.

[54] Tapio Lönnberg et al. “Single-cell RNA-seq and computational analysis using temporal mixture modelling resolves Th1/Tfh fate bifurcation in malaria”. en. In: *Sci Immunol* 2.9 (Mar. 2017).

[55] Xiaojie Qiu et al. “Reversed graph embedding resolves complex single-cell trajectories”. In: *Nature methods* 14.10 (2017), p. 979.

[56] Wouter Saelens et al. “A comparison of single-cell trajectory inference methods”. In: *Nature biotechnology* 37.5 (2019), pp. 547–554.

[57] Veena S Patil et al. “Precursors of human CD4 cytotoxic T lymphocytes identified by single-cell transcriptome analysis”. en. In: *Sci Immunol* 3.19 (Jan. 2018).

[58] David Schafflick et al. In: *Nat Commun* 11.1 (2020), pp. 1–14.

[59] Kelly Street et al. “Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics”. In: *BMC genomics* 19.1 (2018), pp. 1–16.

[60] Gioele La Manno et al. “RNA velocity of single cells”. In: *Nature* 560.7719 (2018), pp. 494–498.

[61] Wouter Saelens et al. “A comparison of single-cell trajectory inference methods”. In: *Nature biotechnology* 37.5 (2019), pp. 547–554.

[62] Tamim Abdelaal et al. “A comparison of automatic cell identification methods for single-cell RNA sequencing data”. In: *Genome biology* 20.1 (2019), pp. 1–19.

[63] J Javier Diaz-Mejia et al. “Evaluation of methods to assign cell type labels to cell clusters from single-cell RNA-sequencing data”. In: *F1000Research* 8 (2019).

[64] Qianhui Huang et al. “Evaluation of Cell Type Annotation R Packages on Single-cell RNA-seq Data”. In: *Genomics, Proteomics & Bioinformatics* (2020).

[65] Vladimir Yu Kiselev, Andrew Yiu, and Martin Hemberg. “scmap: projection of single-cell RNA-seq data across data sets”. In: *Nat Methods* 15 (Apr. 2018), p. 359.

[66] Yuqi Tan and Patrick Cahan. "SingleCellNet: a computational tool to classify single cell RNA-Seq data across platforms and across species". In: *Cell systems* 9.2 (2019), pp. 207–213.

[67] Xinxin Zhang et al. "CellMarker: a manually curated resource of cell markers in human and mouse". In: *Nucleic acids research* 47.D1 (2019), pp. D721–D728.

[68] Allen W Zhang et al. "Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling". In: *Nature methods* 16.10 (2019), pp. 1007–1015.

[69] Aleksandr Ianevski, Anil K Giri, and Tero Aittokallio. "Fully-automated cell-type identification with specific markers extracted from single-cell transcriptomic data". In: *bioRxiv* (2019), p. 812131.

[70] Hannah A Pliner, Jay Shendure, and Cole Trapnell. "Supervised classification enables rapid annotation of cell atlases". In: *Nature methods* 16.10 (2019), pp. 983–986.

[71] Tamim Abdelaal et al. "A comparison of automatic cell identification methods for single-cell RNA sequencing data". In: *Genome biology* 20.1 (2019), pp. 1–19.

[72] Francisco Avila Cobos et al. "Benchmarking of cell type deconvolution pipelines for transcriptomics data". In: *Nature communications* 11.1 (2020), pp. 1–14.

[73] Shaked Afik, Gabriel Raulet, and Nir Yosef. "Reconstructing B-cell receptor sequences from short-read single-cell RNA sequencing with BRAPeS". In: *Life science alliance* 2.4 (2019).

[74] Shaked Afik et al. "Targeted reconstruction of T cell receptor sequence from single cell RNA-seq links CDR3 length to T cell differentiation state". In: *Nucleic acids research* 45.16 (2017), e148–e148.

[75] Huijuan Feng et al. "Complexity and graded regulation of neuronal cell-type–specific alternative splicing revealed by single-cell RNA sequencing". In: *Proceedings of the National Academy of Sciences* 118.10 (2021).

[76] Orit Rozenblatt-Rosen et al. "Building a high-quality Human Cell Atlas". In: *Nature Biotechnology* 39.2 (2021), pp. 149–153. ISSN: 1087-0156. DOI: 10.1038/s41587-020-00812-4.

[77] Mark D Wilkinson et al. "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific data* 3.1 (2016), pp. 1–9.

[78] Bo Li et al. "Cumulus provides cloud-based data analysis for large-scale single-cell and single-nucleus RNA-seq". In: *Nature Methods* 17.8 (2020), pp. 793–798.

[79] HuBMAP Consortium et al. "The human body at cellular resolution: the NIH Human Biomolecular Atlas Program". In: *Nature* 574.7777 (2019), p. 187.

[80] Yuhan Hao et al. "Integrated analysis of multimodal single-cell data". In: *bioRxiv* (2020).

[81] Irene Papatheodorou et al. “Expression Atlas update: from tissues to single cells”. In: *Nucleic acids research* 48.D1 (2020), pp. D77–D83.

[82] Shuai He et al. “Single-cell transcriptome profiling of an adult human cell atlas of 15 major organs”. In: *Genome biology* 21.1 (2020), pp. 1–34.

[83] Tabula Muris Consortium et al. “Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris”. In: *Nature* 562.7727 (2018), pp. 367–372.

[84] Meghan C Mott, Joshua A Gordon, and Walter J Koroshetz. “The NIH BRAIN Initiative: Advancing neurotechnologies, integrating disciplines”. In: *PLoS biology* 16.11 (2018), e3000066.

[85] Johannes C Melms et al. “A molecular single-cell lung atlas of lethal COVID-19”. In: *Nature* (2021), pp. 1–9.

[86] Jing Jiang et al. “scREAD: A single-cell RNA-Seq database for Alzheimer’s Disease”. In: *Iscience* 23.11 (2020), p. 101769.

[87] Xiuwei Zhang, Chenling Xu, and Nir Yosef. “SymSim: simulating multi-faceted variability in single cell RNA sequencing”. In: *Nat Commun* 10.1 (2019), pp. 1–16.

[88] Eric Lubeck et al. “Single-cell in situ RNA profiling by sequential hybridization”. In: *Nature methods* 11.4 (2014), p. 360.

[89] Jeffrey R Moffitt et al. “High-throughput single-cell gene-expression profiling with multiplexed error-robust fluorescence in situ hybridization”. In: *Proceedings of the National Academy of Sciences* 113.39 (2016), pp. 11046–11051.

[90] Marlon Stoeckius et al. “Simultaneous epitope and transcriptome measurement in single cells”. In: *Nat Methods* 14.9 (2017), p. 865.

[91] Ansuman T Satpathy et al. “Massively parallel single-cell chromatin landscapes of human immune cell development and intratumoral T cell exhaustion”. In: *Nature biotechnology* 37.8 (2019), pp. 925–936.

[92] Caleb A Lareau et al. “Droplet-based combinatorial indexing for massive-scale single-cell chromatin accessibility”. In: *Nature Biotechnology* 37.8 (2019), pp. 916–924.

[93] Rongxin Fang et al. “Comprehensive analysis of single cell ATAC-seq data with SnapATAC”. In: *Nature communications* 12.1 (2021), pp. 1–15.

[94] Stephen J Clark et al. “scNMT-seq enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells”. In: *Nature communications* 9.1 (2018), pp. 1–9.

[95] Junyue Cao et al. “Joint profiling of chromatin accessibility and gene expression in thousands of single cells”. In: *Science* 361.6409 (2018), pp. 1380–1385.

[96] El-Ad David Amir et al. “viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia”. en. In: *Nat. Biotechnol.* 31.6 (June 2013), pp. 545–552.

[97] Allon Wagner, Aviv Regev, and Nir Yosef. "Revealing the vectors of cellular identity with single-cell genomics". en. In: *Nat. Biotechnol.* 34.11 (Nov. 2016), pp. 1145–1160.

[98] Emma Pierson and Christopher Yau. "ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis". en. In: *Genome Biol.* 16 (Nov. 2015), p. 241.

[99] Rhonda Bacher and Christina Kendziorski. "Design and computational analysis of single-cell RNA-sequencing experiments". en. In: *Genome Biol.* 17 (Apr. 2016), p. 63.

[100] Oliver Stegle, Sarah A Teichmann, and John C Marioni. "Computational and analytical challenges in single-cell transcriptomics". en. In: *Nat. Rev. Genet.* 16.3 (Mar. 2015), pp. 133–145.

[101] Aaron T L Lun and John C Marioni. "Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data". en. In: *Biostatistics* 18.3 (July 2017), pp. 451–464.

[102] Catalina A Vallejos, John C Marioni, and Sylvia Richardson. "BASiCS: Bayesian Analysis of Single-Cell Sequencing Data". en. In: *PLoS Comput. Biol.* 11.6 (June 2015), e1004333.

[103] Keegan D Korthauer et al. "A statistical approach for identifying differential distributions in single-cell RNA-seq experiments". en. In: *Genome Biol.* 17.1 (Oct. 2016), p. 222.

[104] Christoph Ziegenhain et al. "Comparative analysis of single-cell RNA sequencing methods". In: *Molecular cell* 65.4 (2017), pp. 631–643.

[105] Luke Zappia, Belinda Phipson, and Alicia Oshlack. "Splatter: simulation of single-cell RNA sequencing data". en. In: *Genome Biol.* 18.1 (Sept. 2017), p. 174.

[106] J Peccoud and B Ycart. "Markovian Modeling of Gene-Product Synthesis". In: *Theor. Popul. Biol.* 48.2 (1995), pp. 222–234.

[107] B Munsky, G Neuert, and A van Oudenaarden. "Using Gene Expression Noise to Understand Gene Regulation". In: *Science* 336.6078 (2012), pp. 183–187.

[108] Daniel R Larson. "What do expression dynamics tell us about the mechanism of transcription?" en. In: *Curr. Opin. Genet. Dev.* 21.5 (Oct. 2011), pp. 591–599.

[109] Jong Kyoung Kim and John C Marioni. "Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data". In: *Genome Biol.* 14 (2013), R7.

[110] Simone Picelli et al. "Smart-seq2 for sensitive full-length transcriptome profiling in single cells". en. In: *Nat. Methods* 10.11 (Nov. 2013), pp. 1096–1098.

[111] Amit Zeisel et al. "Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq". en. In: *Science* 347.6226 (Mar. 2015), pp. 1138–1142.

[112] Jellert T Gaublomme et al. "Single-Cell Genomics Unveils Critical Regulators of Th17 Cell Pathogenicity". en. In: *Cell* 163.6 (Dec. 2015), pp. 1400–1412.

[113] Olivia Padovan-Merhar et al. "Single mammalian cells compensate for differences in cellular volume and DNA copy number through independent global transcriptional mechanisms". en. In: *Mol. Cell* 58.2 (Apr. 2015), pp. 339–352.

[114] Keren Bahar Halpern et al. "Bursty Gene Expression in the Intact Mammalian Liver". In: *Mol. Cell* 58.1 (2015), pp. 147–156.

[115] Samuel O Skinner et al. "Single-cell analysis of transcription kinetics across the cell cycle". en. In: *Elife* 5 (Jan. 2016), e12175.

[116] Roy D Dar et al. "Transcriptional Bursting Explains the Noise-Versus-Mean Relationship in mRNA and Protein Levels". In: *PLoS One* 11.7 (2016), e0158298.

[117] Heng Xu et al. "Combining protein and mRNA quantification to decipher transcriptional regulation". In: *Nat. Methods* 12.8 (2015), pp. 739–742.

[118] David M Suter et al. "Mammalian genes are transcribed with widely different bursting kinetics". en. In: *Science* 332.6028 (Apr. 2011), pp. 472–474.

[119] Caroline R Bartman et al. "Enhancer Regulation of Transcriptional Bursting Parameters Revealed by Forced Chromatin Looping". en. In: *Mol. Cell* 62.2 (Apr. 2016), pp. 237–247.

[120] Michal Rabani et al. "A Massively Parallel Reporter Assay of 3' UTR Sequences Identifies In Vivo Rules for mRNA Degradation". In: *Mol. Cell* 68.6 (2017), 1083–1094.e5.

[121] Joseph Felsenstein and Joseph Felenstein. *Inferring phylogenies*. Vol. 2. Sinauer associates Sunderland, MA, 2004.

[122] Catalina A Vallejos et al. "Normalizing single-cell RNA sequencing data: challenges and opportunities". en. In: *Nat. Methods* 14.6 (June 2017), pp. 565–571.

[123] Diego Adhemar Jaitin et al. "Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types". en. In: *Science* 343.6172 (Feb. 2014), pp. 776–779.

[124] Gozde Kar et al. "Flipping between Polycomb repressed and active transcriptional states introduces noise in gene expression". en. In: *Nat. Commun.* 8.1 (June 2017), p. 36.

[125] Shaked Afik et al. "Targeted reconstruction of T cell receptor sequence from single cell RNA-seq links CDR3 length to T cell differentiation state". In: *Nucleic Acids Res.* (July 2017).

[126] Belinda Phipson, Luke Zappia, and Alicia Oshlack. "Gene length and detection bias in single cell RNA sequencing protocols". en. In: *F1000Res.* 6 (Apr. 2017), p. 595.

[127] Bo Wang et al. "Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning". en. In: *Nat. Methods* 14.4 (Apr. 2017), pp. 414–416.

[128] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data". en. In: *Bioinformatics* 26.1 (Jan. 2010), pp. 139–140.

[129] Michael I Love, Wolfgang Huber, and Simon Anders. "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2". en. In: *Genome Biol.* 15.12 (2014), p. 550.

[130] Michael I Love, Simon Anders, and Wolfgang Huber. *Analyzing RNA-seq data with DESeq2*. `http://bioconductor.org/packages/devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html`. Accessed: 2018-6-29. June 2018.

[131] Charlotte Soneson and Mark D Robinson. "Bias, robustness and scalability in single-cell differential expression analysis". en. In: *Nat. Methods* 15.4 (Feb. 2018), p. 255.

[132] Cole Trapnell et al. "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells". In: *Nature biotechnology* 32.4 (2014), p. 381.

[133] *How Many Cells*. `https://satijalab.org/howmanycells`. Accessed: 2018-5-19.

[134] Jong Kyoung Kim et al. "Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression". In: *Nat. Commun.* 6.1 (2015).

[135] Amos Tanay and Aviv Regev. "Scaling single-cell genomics from phenomenology to mechanism". In: *Nature* 541.7637 (2017), pp. 331–338.

[136] Evan Macosko et al. "Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets". In: *Cell* 161.5 (2017), pp. 1202–1214.

[137] Stefan Semrau et al. "Dynamics of lineage commitment revealed by single-cell transcriptomics of differentiating embryonic stem cells". In: *Nat Commun* 8.1 (2017), p. 1096.

[138] Jellert T Gaublomme et al. "Single-Cell Genomics Unveils Critical Regulators of Th17 Cell Pathogenicity". In: *Cell* 163.6 (2015), pp. 1400–1412.

[139] Anoop P Patel et al. "Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma". In: *Science* 344.6190 (June 2014), pp. 1396–1401.

[140] Philipp Angerer et al. "Single cells make big data: New challenges and opportunities in transcriptomics". In: *Curr Opin Syst Biol* 4 (2017), pp. 85–91.

[141] Nicholas Schaum et al. "Single-cell transcriptomic characterization of 20 organs and tissues from individual mice creates a Tabula Muris". In: *Nature* 562.7727 (2018), p. 367.

[142] *BRAIN Initiative Cell Census Network (BICCN)*. https://biccn.org/data/. 2018.

[143] Lu Wen and Fuchou Tang. "Boosting the power of single-cell analysis". In: *Nat Biotechnol* 36 (May 2018), p. 408.

[144] Laleh Haghverdi et al. "Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors". In: *Nat Biotechnol* 36.5 (2018), pp. 421–427.

[145] Allon Wagner, Aviv Regev, and Nir Yosef. "Revealing the vectors of cellular identity with single-cell genomics". In: *Nat Biotechnol* 34.11 (2016), pp. 1145–1160.

[146] W Evan Johnson, Cheng Li, and Ariel Rabinovic. "Adjusting batch effects in microarray expression data using empirical Bayes methods". In: *Biostatistics* 8.1 (2007), pp. 118–127.

[147] Matthew E. Ritchie et al. "limma powers differential expression analyses for RNA-sequencing and microarray studies". In: *Nucleic Acids Res* 43.7 (Apr. 2015), e47–e47. ISSN: 0305-1048.

[148] Davide Risso et al. "A general and flexible method for signal extraction from single-cell RNA-seq data". In: *Nat Commun* 9 (2018), p. 284.

[149] Joshua D Welch et al. "Single-Cell Multi-omic Integration Compares and Contrasts Features of Brain Cell Identity". In: *Cell* 177.7 (2019), pp. 1873–1887.

[150] Brian Hie, Bryan Bryson, and Bonnie Berger. "Efficient integration of heterogeneous single-cell transcriptomes using Scanorama". In: *Nature biotechnology* 37.6 (2019), pp. 685–691.

[151] David DeTomaso and Nir Yosef. "FastProject: A tool for low-dimensional analysis of single-cell RNA-seq data". In: *Bioinformatics* 17.1 (Dec. 2016), p. 315.

[152] Caleb Weinreb et al. "Lineage tracing on transcriptional landscapes links state to fate during differentiation". In: *Science* 367.6479 (2020).

[153] Florian Wagner and Itai Yanai. "Moana: A robust and scalable cell type classification framework for single-cell RNA-Seq data". In: *bioRxiv* (2018). [PREPRINT]. eprint: 10.1101/456129.

[154] Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *Proc Int Conf Learning Representations*. 2014.

[155] Christos Louizos et al. "The Variational Fair Autoencoder". In: *Proc Int Conf Learning Representations*. 2016.

[156] Vegard Nygaard, Einar Andreas Rødland, and Eivind Hovig. "Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses". In: *Biostatistics* 17.1 (2016), pp. 29–39. ISSN: 1465-4644. DOI: 10.1093/biostatistics/kxv027.

[157] Michael B Cole et al. "Performance assessment and selection of normalization procedures for single-cell RNA-seq". In: *Cell Syst* 8.4 (2019), pp. 315–328.

[158] *10x Genomics*. https://support.10xgenomics.com/single-cell-gene-expression/datasets. 2017.

[159] Maayan Baron et al. "A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure". In: *Cell Syst* 3.4 (2016), pp. 346–360.

[160] Mauro J Muraro et al. "A single-cell transcriptome atlas of the human pancreas". In: *Cell Syst* 3.4 (2016), pp. 385–394.

[161] Hannah Hochgerner et al. "Conserved properties of dentate gyrus neurogenesis across postnatal development revealed by single-cell RNA sequencing". In: *Nat Neurosci* 21.2 (2018), p. 290.

[162] Betsabeh Khoramian Tusi et al. "Population snapshots predict early haematopoietic and erythroid hierarchies". In: *Nature* 555.7694 (Feb. 2018), pp. 54–60.

[163] Franziska Paul et al. "Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors". In: *Cell* 163.7 (2015), pp. 1663–1677. ISSN: 0092-8674. DOI: `https://doi.org/10.1016/j.cell.2015.11.013`.

[164] Arpiar Saunders et al. "Molecular diversity and specializations among the cells of the adult mouse brain". In: *Cell* 174.4 (2018), pp. 1015–1030.

[165] Simone Picelli et al. "Full-length RNA-seq from single cells using Smart-seq2". In: *Nat Protoc* 9.1 (2014), p. 171.

[166] Diego Adhemar Jaitin et al. "Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types". In: *Science* 343.6172 (2014), pp. 776–779.

[167] Tamar Hashimshony et al. "CEL-Seq2: sensitive highly-multiplexed single-cell RNA-Seq". In: *Genome Biol* 17.1 (2016), p. 77.

[168] Christoph Ziegenhain et al. "Comparative Analysis of Single-Cell RNA Sequencing Methods". In: *Mol Cell* 65.4 (Feb. 2017), pp. 631–643. ISSN: 1097-2765.

[169] Ilya Korsunsky et al. "Fast, sensitive and accurate integration of single-cell data with Harmony". In: *Nat Methods* 16.12 (2019), p. 1289.

[170] Jun-Yan Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *IEEE Int Conf Comp Vision*. 2017, pp. 2223–2232.

[171] Matthew Amodio and Smita Krishnaswamy. "MAGAN: Aligning Biological Manifolds". In: *Proc Int Conf Mach Learn*. 2018, pp. 215–223.

[172] Leland McInnes et al. "UMAP: Uniform Manifold Approximation and Projection". In: *J Open Source Softw* 3.29 (2018), p. 861.

[173] Sung-Ho Goh et al. "The human reticulocyte transcriptome". In: *Physiol Genomics* 30.2 (2007), pp. 172–178.

[174] *Transcription profiling by high throughput sequencing of murine hematopoietic progenitors and lineage cells*. https://www.ebi.ac.uk/gxa/experiments/E-MTAB-3079/Results. 2018.

[175] Wenjun Ju et al. "Defining cell-type specificity at the transcriptional level in human disease". In: *Genome Res* (2013), p. 155697.

[176] Irene Papatheodorou et al. "Expression Atlas: gene and protein expression across multiple studies and organisms". In: *Nucleic Acids Res* 46.1 (2017), pp. 246–251.

[177] Carol J Bult et al. "Mouse genome database (MGD)". In: *Nucleic acids research* 47.D1 (2019), pp. D801–D806.

[178] Malte D Luecken et al. "Benchmarking atlas-level data integration in single-cell genomics". In: *BioRxiv* (2020).

[179] Baochen Sun, Jiashi Feng, and Kate Saenko. "Return of frustratingly easy domain adaptation". In: *AAAI Conf on Artificial Intelligence*. 2016.

[180] Martin Ester et al. "A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *KDD*. 1996, pp. 226–231.

[181] Jacob H. Levine et al. "Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis". In: *Cell* 162.1 (July 2015), pp. 184–197.

[182] Helder I Nakaya et al. "Systems biology of vaccination for seasonal influenza in humans". In: *Nat Immunol* 12.8 (July 2011), pp. 786–795.

[183] Güllü Görgün et al. "Chronic lymphocytic leukemia cells induce changes in gene expression of CD4 and CD8 T cells". In: *J Clin Invest* 115.7 (2005), pp. 1797–805.

[184] Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. "edgeR: A bioconductor package for differential expression analysis of digital gene expression data". In: *Bioinformatics* 26.1 (Jan. 2010), pp. 139–140.

[185] Charlotte Soneson and Mark D Robinson. "Bias, robustness and scalability in single-cell differential expression analysis". In: *Nat Methods* 15.4 (2018), p. 255.

[186] Andrew Gelman and Jennifer Hill. *Data Analysis using Regression and Multilevel/Hierarchical Models*. Cambridge university press, 2006.

[187] Diederik P Kingma et al. "Semi-supervised Learning with Deep Generative Models". In: *Adv Neural Inf Process Syst*. 2014, pp. 3581–3589.

[188] Yee W Teh, David Newman, and Max Welling. "A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation". In: *Adv Neural Inf Process Syst*. 2007, pp. 1353–1360.

[189] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *Proc Int Conf Learning Representations*. 2015.

[190] Casper Kaae Sønderby et al. "Ladder variational autoencoders". In: *Adv Neural Inf Process Syst*. 2016, pp. 3738–3746.

[191] Sergey Ioffe and Christian Szegedy. "Batch Normalization: accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proc Int Conf Mach Learn*. 2015.

[192] Jacob Goldberger and Ehud Ben-Reuven. "Training deep neural-networks using a noise adaptation layer". In: *Proc Int Conf Learning Representations*. 2017.

[193] Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. "Hierarchical Multi-Label Classification Networks". In: *Proc Int Conf Mach Learn*. Vol. 80. 2018, pp. 5075–5084.

[194] Leonhard Held and Manuela Ott. "On p-Values and Bayes Factors". In: *Annu Rev Stat Appl* 5.1 (2018), pp. 393–419.

[195] Robert E Kass and Adrian E Raftery. "Bayes Factors". In: *J Am Stat Assoc* 90.430 (1995), pp. 773–795.

[196] Tallulah S Andrews and Martin Hemberg. "M3Drop: dropout-based feature selection for scRNASeq". In: *Bioinformatics* 35.16 (2019), pp. 2865–2867.

[197] Matthew Amodio et al. "Exploring single-cell data with deep multitasking neural networks". In: *Nat Meth* (2019), pp. 1–7.

[198] Jiarui Ding, Anne Condon, and Sohrab P. Shah. "Interpretable dimensionality reduction of single cell transcriptome data with deep generative models". In: *Nat Commun* 9.1 (Dec. 2018), p. 2002.

[199] Dongfang Wang and Jin Gu. "VASC: Dimension Reduction and Visualization of Single-cell RNA-seq Data by Deep Variational Autoencoder". In: *Genomics, Proteomics & Bioinformatics* 16.5 (2018), pp. 320–331.

[200] Gökcen Eraslan et al. "Single-cell RNA-seq denoising using a deep count autoencoder". In: *Nat Commun* 10.1 (2019), p. 390.

[201] Christopher Heje Grønbech et al. "scVAE: Variational auto-encoders for single-cell gene expression data". In: *Bioinformatics* (2020).

[202] Qiwen Hu and Casey S Greene. "Parameter tuning is a key part of dimensionality reduction via deep variational autoencoders for single cell RNA transcriptomics". In: *Pacific Symposium on Biocomputing*. Vol. 24. 2019, p. 362.

[203] Romain Lopez et al. "A joint model of unpaired data from scRNA-seq and spatial transcriptomics for imputing missing gene expression measurements". In: *ICML Workshop on Computational Biology*. 2019.

[204] A M Klein et al. "Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells". Eng. In: *Cell* 161 (May 2015), pp. 1187–1201.

[205] Tim Stuart and Rahul Satija. *Integrative single-cell analysis*. 2019.

[206] Elham Azizi et al. "Single-Cell Map of Diverse Immune Phenotypes in the Breast Tumor Microenvironment". en. In: *Cell* 174.5 (Aug. 2018), 1293–1308.e36.

[207] Michael J T Stubbington et al. "Single-cell transcriptomics to explore the immune system in health and disease". en. In: *Science* 358.6359 (Oct. 2017), pp. 58–63.

[208] Alex K Shalek and Mikael Benson. "Single-cell analyses to tailor treatments". en. In: *Sci. Transl. Med.* 9.408 (Sept. 2017), eaan4730.

[209] J J Iliff et al. "A paravascular pathway facilitates CSF flow through the brain parenchyma and the clearance of interstitial solutes, including amyloid beta". eng. In: *Sci. Transl. Med.* 4 (Aug. 2012), 147ra111.

[210] C Schlager et al. "Effector T-cell trafficking between the leptomeninges and the cerebrospinal fluid". eng. In: *Nature* 530 (Feb. 2016), pp. 349–353.

[211] B Engelhardt et al. "Vascular, glial, and lymphatic immune gateways of the central nervous system". eng. In: *Acta Neuropathol.* 132 (Sept. 2016), pp. 317–338.

[212] Sungpil Han et al. "Comprehensive immunophenotyping of cerebrospinal fluid cells in patients with neuroimmunological diseases". en. In: *J. Immunol.* 192.6 (Mar. 2014), pp. 2551–2563.

[213] P Kivisakk et al. "Human cerebrospinal fluid central memory CD4+ T cells: evidence for trafficking through choroid plexus and meninges via P-selectin". eng. In: *Proc. Natl. Acad. Sci. U. S. A.* 100 (July 2003), pp. 8389–8394.

[214] R M Ransohoff and B Engelhardt. "The anatomical and cellular basis of immune surveillance in the central nervous system". eng. In: *Nat. Rev. Immunol.* 12 (Sept. 2012), pp. 623–635.

[215] B Brynedal et al. "Gene expression profiling in multiple sclerosis: a disease of the central nervous system, but with relapses triggered in the periphery?" eng. In: *Neurobiol. Dis.* 37 (Mar. 2010), pp. 613–621.

[216] Shelli F Farhadian et al. "Single-cell RNA sequencing reveals microglia-like cells in cerebrospinal fluid during virologically suppressed HIV". eng. In: *JCI Insight* 3.18 (Sept. 2018).

[217] Alastair Compston and Alasdair Coles. "Multiple sclerosis". eng. In: *Lancet* 372.9648 (Oct. 2008), pp. 1502–1517.

[218] Erica L Eggers et al. "Clonal relationships of CSF B cells in treatment-naive multiple sclerosis patients". en. In: *JCI Insight* 2.22 (Nov. 2017).

[219] J Haas et al. "B cells undergo unique compartmentalized redistribution in multiple sclerosis". eng. In: *J. Autoimmun.* 37 (Dec. 2011), pp. 289–299.

[220] A Corcione et al. "Recapitulation of B cell differentiation in the central nervous system of patients with multiple sclerosis". eng. In: *Proc. Natl. Acad. Sci. U. S. A.* 101 (July 2004), pp. 11064–11069.

[221] Stephen L Hauser et al. "Ocrelizumab versus Interferon Beta-1a in Relapsing Multiple Sclerosis". en. In: *N. Engl. J. Med.* 376.3 (Jan. 2017), pp. 221–234.

[222] H W Kreth et al. "Immunohistochemical identification of T-lymphocytes in the central nervous system of patients with multiple sclerosis and subacute sclerosing panencephalitis". en. In: *J. Neuroimmunol.* 2.2 (Apr. 1982), pp. 177–183.

[223] Joana Machado-Santos et al. “The compartmentalized inflammatory response in the multiple sclerosis brain is composed of tissue-resident CD8+ T lymphocytes and B cells”. eng. In: *Brain* 141.7 (July 2018), pp. 2066–2082.

[224] M Rangachari and V K Kuchroo. “Using EAE to better understand principles of immune function and autoimmune pathology”. eng. In: *J. Autoimmun.* 45 (Sept. 2013), pp. 31–39.

[225] Ruth Dobson et al. “Cerebrospinal fluid oligoclonal bands in multiple sclerosis and clinically isolated syndromes: a meta-analysis of prevalence, prognosis and effect of latitude”. en. In: *J. Neurol. Neurosurg. Psychiatry* 84.8 (Aug. 2013), pp. 909–914.

[226] M C Kowarik et al. “Immune cell subtyping in the cerebrospinal fluid of patients with neurological diseases”. eng. In: *J. Neurol.* 261 (Jan. 2014), pp. 130–143.

[227] Derek J Theisen et al. “WDFY4 is required for cross-presentation in response to viral and tumor antigens”. en. In: *Science* 362.6415 (Nov. 2018), pp. 694–699.

[228] Anne Waschbisch et al. “Pivotal Role for CD16+ Monocytes in Immune Surveillance of the Central Nervous System”. en. In: *J. Immunol.* 196.4 (Feb. 2016), pp. 1558–1567.

[229] Kok Loon Wong et al. “Gene expression profiling reveals the defining features of the classical, intermediate, and nonclassical human monocyte subsets”. en. In: *Blood* 118.5 (Aug. 2011), e16–31.

[230] Tobias Goldmann et al. “Origin, fate and dynamics of macrophages at central nervous system interfaces”. en. In: *Nat. Immunol.* 17.7 (July 2016), pp. 797–805.

[231] Fabia Filipello et al. “The Microglial Innate Immune Receptor TREM2 Is Required for Synapse Elimination and Normal Brain Connectivity”. en. In: *Immunity* 48.5 (May 2018), 979–991.e8.

[232] Giuseppe Faraco et al. “Perivascular macrophages mediate the neurovascular and cognitive dysfunction associated with hypertension”. en. In: *J. Clin. Invest.* 126.12 (Dec. 2016), pp. 4674–4689.

[233] Marta Joana Costa Jordão et al. “Single-cell profiling identifies myeloid cell subsets with distinct fates during neuroinflammation”. en. In: *Science* 363.6425 (Jan. 2019).

[234] Takahiro Masuda et al. “Spatial and temporal heterogeneity of mouse and human microglia at single-cell resolution”. en. In: *Nature* 566.7744 (Feb. 2019), pp. 388–392.

[235] M D Robinson, D J McCarthy, and G K Smyth. *edgeR: a Bioconductor package for differential expression analysis of digital gene expression data.* 2010.

[236] Maren Lindner et al. “Fibroblast growth factor signalling in multiple sclerosis: inhibition of myelination and induction of pro-inflammatory environment by FGF9”. en. In: *Brain* 138.Pt 7 (July 2015), pp. 1875–1893.

[237] Martina Cesani et al. “Metallothioneins as dynamic markers for brain disease in lysosomal disorders”. en. In: *Ann. Neurol.* 75.1 (Jan. 2014), pp. 127–137.

[238] P Kivisakk et al. "T-cells in the cerebrospinal fluid express a similar repertoire of inflammatory chemokine receptors in the absence or presence of CNS inflammation: implications for CNS trafficking". eng. In: *Clin. Exp. Immunol.* 129 (Sept. 2002), pp. 510–518.

[239] T Schneider-Hohendorf et al. "VLA-4 blockade promotes differential routes into human CNS involving PSGL-1 rolling of T cells and MCAM-adhesion of TH17 cells". eng. In: *J. Exp. Med.* 211 (Aug. 2014), pp. 1833–1846.

[240] Naveed Ahmed Khan et al. "FimH-mediated Escherichia coli K1 invasion of human brain microvascular endothelial cells". en. In: *Cell. Microbiol.* 9.1 (Jan. 2007), pp. 169–178.

[241] H C von Budingen et al. "B cell exchange across the blood-brain barrier in multiple sclerosis". eng. In: *J. Clin. Invest.* 122 (Dec. 2012), pp. 4533–4543.

[242] S Cepok et al. "Short-lived plasma blasts are the main B cell effector subset during the course of multiple sclerosis". eng. In: *Brain* 128 (July 2005), pp. 1667–1676.

[243] Nancy S Longo et al. "Analysis of somatic hypermutation in X-linked hyper-IgM syndrome shows specific deficiencies in mutational targeting". en. In: *Blood* 113.16 (Apr. 2009), pp. 3706–3715.

[244] E Rodríguez-Martín et al. "Natural killer cell subsets in cerebrospinal fluid of patients with multiple sclerosis". en. In: *Clin. Exp. Immunol.* 180.2 (May 2015), pp. 243–249.

[245] Jesse M Zhang et al. "An interpretable framework for clustering single-cell RNA-Seq datasets". en. In: *BMC Bioinformatics* 19.1 (Mar. 2018), p. 93.

[246] Norio Chihara et al. *Induction and transcriptional regulation of the co-inhibitory gene module in T cells*. 2018.

[247] Daniel E Lowther et al. "PD-1 marks dysfunctional regulatory T cells in malignant gliomas". en. In: *JCI Insight* 1.5 (Apr. 2016).

[248] David DeTomaso and Nir Yosef. "FastProject: a tool for low-dimensional analysis of single-cell RNA-Seq data". en. In: *BMC Bioinformatics* 17.1 (Aug. 2016), p. 315.

[249] J T Gaublomme et al. "Single-Cell Genomics Unveils Critical Regulators of Th17 Cell Pathogenicity". Eng. In: *Cell* (Nov. 2015).

[250] U Feger et al. "Increased frequency of CD4+ CD25+ regulatory T cells in the cerebrospinal fluid but not in the blood of multiple sclerosis patients". eng. In: *Clin. Exp. Immunol.* 147 (Mar. 2007), pp. 412–418.

[251] X Liu et al. "Transcription factor achaete-scute homologue 2 initiates follicular T-helper-cell development". eng. In: *Nature* 507 (Mar. 2014), pp. 513–518.

[252] R I Nurieva et al. "Bcl6 mediates the development of T follicular helper cells". eng. In: *Science* 325 (Aug. 2009), pp. 1001–1005.

[253] J Romme Christensen et al. “Systemic inflammation in progressive multiple sclerosis involves follicular T-helper, Th17- and activated B-cells and correlates with progression”. eng. In: *PLoS One* 8 (2013), e57820.

[254] Yoshimi Enose-Akahata et al. “Immunophenotypic characterization of CSF B cells in virus-associated neuroinflammatory diseases”. en. In: *PLoS Pathog.* 14.4 (Apr. 2018), e1007042.

[255] Deepak A Rao et al. “Pathologically expanded peripheral T helper cell subset drives B cells in rheumatoid arthritis”. In: *Nature* 542.7639 (Feb. 2017), pp. 110–114.

[256] Kristin Hollister et al. “Insights into the role of Bcl6 in follicular Th cells using a new conditional mutant mouse model”. en. In: *J. Immunol.* 191.7 (Oct. 2013), pp. 3705–3711.

[257] Weiwei Fu et al. “Deficiency in T follicular regulatory cells promotes autoimmunity”. en. In: *J. Exp. Med.* 215.3 (Mar. 2018), pp. 815–825.

[258] E Bettelli et al. “Myelin oligodendrocyte glycoprotein-specific T cell receptor transgenic mice develop spontaneous autoimmune optic neuritis”. eng. In: *J. Exp. Med.* 197 (May 2003), pp. 1073–1081.

[259] Michael B Cole et al. “Performance Assessment and Selection of Normalization Procedures for Single-Cell RNA-seq”. en. May 2018.

[260] Samuel L Wolock, Romain Lopez, and Allon M Klein. *Scrublet: computational identification of cell doublets in single-cell transcriptomic data.*

[261] Sarah Jäkel et al. “Altered human oligodendrocyte heterogeneity in multiple sclerosis”. en. In: *Nature* 566.7745 (Feb. 2019), pp. 543–547.

[262] Ana Mendanha Falcão et al. “Disease-specific oligodendrocyte lineage cells arise in multiple sclerosis”. en. In: *Nat. Med.* 24.12 (Dec. 2018), pp. 1837–1844.

[263] Arthur Liberzon et al. “Molecular signatures database (MSigDB) 3.0”. en. In: *Bioinformatics* 27.12 (June 2011), pp. 1739–1740.

[264] Kumaran Kandasamy et al. “NetPath: a public resource of curated signal transduction pathways”. en. In: *Genome Biol.* 11.1 (Jan. 2010), R3.

[265] Eli Eisenberg and Erez Y Levanon. *Human housekeeping genes are compact.* 2003.

[266] Renaud Lesnoff and Matthieu Lancelot. “aod: Analysis of overdispersed data”. In: *R package version 1* (2012).

[267] Alessandra Dal Molin, Giacomo Baruzzo, and Barbara Di Camillo. “Single-Cell RNA-Sequencing: Assessment of Differential Expression Analysis Methods”. en. In: *Front. Genet.* 8 (May 2017), p. 62.

[268] A Subramanian et al. “Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles”. eng. In: *Proc. Natl. Acad. Sci. U. S. A.* 102 (Oct. 2005), pp. 15545–15550.

[269] S Picelli et al. "Full-length RNA-seq from single cells using Smart-seq2". eng. In: *Nat. Protoc.* 9 (Jan. 2014), pp. 171–181.

[270] B Langmead et al. *Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol10: R25*. 2009.

[271] Bo Li and Colin N Dewey. "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome". en. In: *BMC Bioinformatics* 12 (Aug. 2011), p. 323.

[272] T Korn and A Kallies. "T cell responses in the central nervous system". eng. In: *Nat. Rev. Immunol.* 17 (Mar. 2017), pp. 179–194.

[273] Burkhard Becher, Sabine Spath, and Joan Goverman. "Cytokine networks in neuroinflammation". en. In: *Nat. Rev. Immunol.* 17.1 (Jan. 2017), pp. 49–59.

[274] Y Cao et al. "Functional inflammatory profiles distinguish myelin-reactive T cells from patients with multiple sclerosis". eng. In: *Sci. Transl. Med.* 7 (May 2015), 287ra74.

[275] A Achiron et al. "Blood transcriptional signatures of multiple sclerosis: unique gene expression of disease activity". eng. In: *Ann. Neurol.* 55 (Mar. 2004), pp. 410–417.

[276] J Satoh et al. "T cell gene expression profiling identifies distinct subgroups of Japanese multiple sclerosis patients". eng. In: *J. Neuroimmunol.* 174 (May 2006), pp. 108–118.

[277] L Ottoboni et al. "An RNA profile identifies two subsets of multiple sclerosis patients differing in disease activity". eng. In: *Sci. Transl. Med.* 4 (Sept. 2012), 153ra131.

[278] S Srinivasan et al. "Dysregulation of MS risk genes and pathways at distinct stages of disease". eng. In: *Neurology(R) neuroimmunology & neuroinflammation* 4 (May 2017), e337.

[279] D Nickles et al. "Blood RNA profiling in a large cohort of multiple sclerosis patients and healthy controls". eng. In: *Hum. Mol. Genet.* 22 (Oct. 2013), pp. 4194–4205.

[280] Charlotte Teunissen et al. "Consensus definitions and application guidelines for control groups in cerebrospinal fluid biomarker studies in multiple sclerosis". en. In: *Mult. Scler.* 19.13 (Nov. 2013), pp. 1802–1809.

[281] D E Berezovsky et al. "Cerebrospinal fluid total protein in idiopathic intracranial hypertension". eng. In: *J. Neurol. Sci.* 381 (Oct. 2017), pp. 226–229.

[282] C Lucchinetti et al. "Heterogeneity of multiple sclerosis lesions: implications for the pathogenesis of demyelination". eng. In: *Ann. Neurol.* 47 (June 2000), pp. 707–717.

[283] Giordani Rodrigues Dos Passos et al. "Th17 Cells Pathways in Multiple Sclerosis and Neuromyelitis Optica Spectrum Disorders: Pathophysiological and Therapeutic Implications". en. In: *Mediators Inflamm.* 2016 (Jan. 2016), p. 5314541.

[284] T Korn et al. "IL-17 and Th17 Cells". eng. In: *Annu. Rev. Immunol.* 27 (2009), pp. 485–517.

[285] N Yosef et al. “Dynamic regulatory network controlling TH17 cell differentiation”. eng. In: *Nature* 496 (Apr. 2013), pp. 461–468.

[286] S Sawcer et al. “Genetic risk and a primary role for cell-mediated immune mechanisms in multiple sclerosis”. eng. In: *Nature* 476 (Aug. 2011), pp. 214–219.

[287] Ben J E Raveney et al. “Eomesodermin-expressing T-helper cells are essential for chronic neuroinflammation”. en. In: *Nat. Commun.* 6 (Oct. 2015), p. 8437.

[288] Marjan Vanheusden et al. “Cytomegalovirus infection exacerbates autoimmune mediated neuroinflammation”. en. In: *Sci. Rep.* 7.1 (Apr. 2017), p. 663.

[289] M Thewissen, V Somers, N Hellings, et al. “CD4+ CD28null T cells in autoimmune disease: pathogenic features and decreased susceptibility to immunoregulation”. In: *The journal of* (2007).

[290] Edoardo Galli et al. “GM-CSF and CXCR4 define a T helper cell signature in multiple sclerosis”. en. In: *Nat. Med.* 25.8 (Aug. 2019), pp. 1290–1300.

[291] Jun Guo et al. “T Follicular Helper-Like Cells Are Involved in the Pathogenesis of Experimental Autoimmune Encephalomyelitis”. eng. In: *Front. Immunol.* 9 (2018), p. 944.

[292] T Korn et al. “Myelin-specific regulatory T cells accumulate in the CNS but fail to control autoimmune inflammation”. eng. In: *Nat. Med.* 13 (Apr. 2007), pp. 423–431.

[293] A Lossius et al. “Selective intrathecal enrichment of G1m1-positive B cells in multiple sclerosis”. eng. In: *Annals of clinical and translational neurology* 4 (Oct. 2017), pp. 756–761.

[294] J N Stern et al. “B cells populating the multiple sclerosis brain mature in the draining cervical lymph nodes”. eng. In: *Sci. Transl. Med.* 6 (Aug. 2014), 248ra107.

[295] Arumugam Palanichamy et al. “Immunoglobulin class-switched B cells form an active immune axis between CNS and periphery in multiple sclerosis”. en. In: *Sci. Transl. Med.* 6.248 (Aug. 2014), 248ra106.

[296] *Tabula Sapiens Portal*. `https://tabula-sapiens-portal.ds.czbiohub.org/`. Accessed: 2021-04-28.

[297] Sheng Wang et al. “Unifying single-cell annotations based on the Cell Ontology”. In: *bioRxiv* (2020), p. 810234.

[298] Romain Lopez, Jeffrey Regier, Michael Cole, Michael Jordan, Nir Yosef. “A deep generative model for gene expression profiles from single-cell RNA sequencing”. In: *arXiv* (Sept. 2017).

[299] Krzysztof Polański et al. “BBKNN: fast batch alignment of single cell transcriptomes”. In: *Bioinformatics* 36.3 (2020), pp. 964–965.

[300] Kyle J Travaglini et al. “A molecular cell atlas of the human lung from single-cell RNA sequencing”. In: *Nature* 587.7835 (2020), pp. 619–625.

[301] Colin Megill et al. "cellxgene: a performant, scalable exploration platform for high dimensional sparse matrices". In: *bioRxiv* (2021). DOI: 10.1101/2021.04.05.438318. eprint: https://www.biorxiv.org/content/early/2021/04/06/2021.04.05.438318.full.pdf. URL: https://www.biorxiv.org/content/early/2021/04/06/2021.04.05.438318.

[302] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[303] Gary M Weiss, Kate McCarthy, and Bibi Zabar. "Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs?" In: *Dmin* 7.35-41 (2007), p. 24.