Dissertation

submitted to the

Combined Faculties for the Natural Sciences and for Mathematics of the

Ruperto-Carola University of Heidelberg, Germany

for the degree of

Doctor of Natural Sciences

presented by

Felix Frauhammer, Master of Science

born in Ludwigsburg, Germany

Oral examination: 13th July 2021

Cell type classification for multi-sample multi-condition comparisons in single-cell RNA sequencing data

Referees: Prof. Dr. Benedikt Brors
Prof. Dr. Karsten Rippe

Abstract

Multicellular organisms require specialized cell types in order to function. While a widely accepted definition does not exist, cell types are regarded as groups of cells with similar properties, such as RNA expression, protein abundance and epigenetic modification.

Single-cell RNA sequencing (scRNAseq) is a recent breakthrough for exploring cell types, providing expression estimates for all genes in thousands of individual cells. Using data-driven algorithms, such as unsupervised clustering, scRNAseq has discovered new cell types and created large reference data sets, next to other exploratory achievements. More recently, scRNAseq was applied to patient cohorts that include different groups, for example disease and healthy or disease subtypes. These multi-sample multi-condition data sets enable statistical inferences between groups, such as differential expression testing. In contrast to projects exploring unknown tissues or species, patient cohorts often study known cell types defined by specific marker genes.

Here, I present Pooled Count Poisson Classification (PCPC), a novel cell type classification approach designed for inference with multi-sample multi-condition scRNAseq data sets. PCPC implements a statistical model that allows researchers to distinguish cells according to marker-based cell type definitions, enabling reproducible and comparable analysis between data sets and technologies (e.g. scRNAseq and flow cytometry). Specifically, PCPC pools marker gene counts across related cells to overcome technical noise, and compares them to a user-defined threshold using the Poisson model.

In this work, I apply PCPC to three different data sets to demonstrate its utility. The first application shows it is able to annotate all lineages in data from human cord blood mononuclear cells (CBMCs), with a single marker gene per cell type.

The second application shows PCPC is able to discriminate fine cell type subsets, using data from a human tumor of mucosa-associated lymphoid tissue (MALT). Many cell types in the MALT tumor microenvironment, and T cell subsets in particular, are transcriptionally related, making their classification difficult. In spite of this challenging complexity, PCPC can even use lowly expressed marker genes, such as FOXP3 marking CD3E$^+$CD4$^+$FOXP3$^+$ regulatory T (T$_{\text{reg}}$) cells. Furthermore, I find T$_{\text{reg}}$ cells isolated from the MALT tumor can further be subdivided into CCR7$^+$ and ICOS$^+$ subsets, indicating a mixture of naive-like and activated T$_{\text{reg}}$ cells. In comparison to unsupervised clustering and the marker-based tool *Garnett*, classification with PCPC has more flexibility and fewer misclassifications, respectively. Thus, PCPC removes obstacles in studying complex tissues with scRNAseq, such as the microenvironment in human tumors.

Furthermore, I demonstrate a multi-sample multi-condition comparison using data from a patient cohort of aggressive and indolent lymphoma subtypes. PCPC is applied to classify CD3E$^+$CD8B$^+$ cytotoxic T cells, followed by differential expression testing between the aggressive and indolent subtypes. This uncovers significantly lower LGALS1 expression in indolent tumors, further implicating this gene in tumor aggressiveness and T cell inhibition.

Currently, PCPC requires data generated with unique molecular identifiers (UMI), as well as substantial manual work. Due to its ability to resolve complex tissues with few marker genes, PCPC may bring clarity to transcriptomic cell type definitions and prove useful for multi-sample multi-condition comparisons in scRNAseq data.

Zusammenfassung

Mehrzellige Organismen bestehen aus verschiedenen, spezialisierten Zelltypen. Eine allgemein akzeptierte Definition dieser Zelltypen existiert nicht. Sie werden aber als Gruppen von Zellen mit ähnlichen Eigenschaften verstanden, wie zum Beispiel RNA-Expression, Zusammensetzung der Eiweißmoleküle oder epigenetische Modifikationen.

Die Einzelzell-RNA-Sequenzierung (scRNAseq, von *single-cell RNA sequencing*) markierte einen Durchbruch in der Erforschung von Zelltypen. Sie misst die Expression aller Gene in Tausenden von Zellen. Durch die Anwendung datengesteuerter Algorithmen, wie unüberwachtem Clustering, hat scRNAseq die Entdeckung neuer Zelltypen und die Erschaffung großer Referenzdatensätze ermöglicht. In neueren Studien wurde scRNAseq auf Patientenkohorten angewandt, welche verschiedene Gruppen umfassen (z.B. kranke und gesunde Probanden oder verschiedene Krankheitsbilder). Durch den Vergleich verschiedener Gruppen mit jeweils mehreren Patienten ermöglichen diese Datensätze statistische Rückschlüsse (Inferenz), etwa die Ermittlung differenzieller Genexpression. Im Gegensatz zu Projekten, die sich mit unbekannten Geweben oder Spezies befassen, untersuchen Studien mit Patientenkohorten häufig bereits bekannte Zelltypen, die durch spezifische Gene, die Zelltypmarker, definiert werden.

In dieser Arbeit stelle ich *Pooled Count Poisson Classification* (PCPC) vor, eine neue Methode der Zelltypklassifizierung speziell für die Inferenz in scRNAseq-Daten mit mehreren Patienten und Gruppen. PCPC basiert auf einem statistischen Modell, welches es Wissenschaftlern ermöglicht, Zellen nach Zelltypmarkern zu unterscheiden. Dies ermöglicht wiederum eine reproduzierbare und vergleichbare Analyse zwischen verschiedensten Datensätzen und Technologien (wie z.B. scRNAseq und Durchflusszytometrie). PCPC bündelt die Zellmarkerexpression verwandter Zellen, um technisches Rauschen zu vermindern. Die gebündelte Expression wird dann unter Verwendung des Poisson-Modells mit einem benutzerdefinierten Schwellenwert verglichen.

Ich wende PCPC in dieser Arbeit auf drei Datensätzen an, um zu zeigen, wie die Zellklassifizierung funktioniert. Zunächst nutze ich die Methode, um alle Zelltyplinien in einem Datensatz von mononukleären Zellen des menschlichen Nabelschnurbluts (CBMCs, von *cord blood mononuclear cells*) zu unterscheiden, mit nur einem einzigen Marker pro Zelltyp.

Des Weiteren kann PCPC feine Zellsubtypen in den Daten eines menschlichen Tumors des Mukosa-assoziierten Lymphgewebes (MALT) auftrennen. Viele Zelltypen in der MALT-Tumormikroumgebung (und insbesondere T-Zell-Untergruppen) sind transkriptionell ähnlich, was deren Klassifizierung erschwert. Trotz dieser anspruchsvollen Komplexität kann PCPC sogar schwach

exprimierte Zelltypmarker verwenden, wie z.B. FOXP3, welches CD3E$^+$CD4$^+$ FOXP3$^+$ regulatorische T (T$_{\text{reg}}$)-Zellen markiert. Außerdem können T$_{\text{reg}}$-Zellen aus dem MALT-Tumor weiter unterteilt werden, in CCR7$^+$ und in ICOS$^+$ Untergruppen. Dies deutet auf eine gemischte Population aus naiven und aktivierten T$_{\text{reg}}$-Zellen im MALT-Tumor hin. Im Vergleich zu unüberwachtem Clustering und dem markergestützten Programm *Garnett* zeigt sich, dass PCPC mehr Flexibilität bei geringerer Fehlklassifizierung bietet. Somit ermöglicht es eine genauere Untersuchung komplexer Gewebe mit scRNAseq, wie z.B. der Mikroumgebung menschlicher Krebsgeschwüre.

Ich nutze außerdem eine Patientenkohorte mit aggressiven und indolenten Lymphomen, um diese beiden Gruppen exemplarisch mit Inferenzstatistik zu vergleichen. Für diese Demonstration werden CD3E$^+$CD8B$^+$ zytotoxische T-Zellen mit PCPC klassifiziert, um sie danach auf differenzielle Expression zwischen aggressiven und indolenten Subtypen zu untersuchen. Hierbei entdecke ich eine signifikant verminderte Expression von LGALS1 in den indolenten Lymphomen. Dies liefert weitere Hinweise auf eine bereits diskutierte Verbindung zwischen LGALS1 mit Tumoraggressivität und T-Zellinhibierung.

Derzeit unterstützt PCPC nur Daten, welche mit eindeutigen molekularen Identifikatoren (UMI, von *unique molecular identifiers*) erzeugt wurden. Außerdem muss der Wissenschaftler die Gene und ihre Schwellenwerte manuell auswählen. Aufgrund seiner Fähigkeit, komplexe Gewebe mit wenigen Zelltypmarkern aufzutrennen, ermöglicht PCPC aber eindeutigere Zelltypdeklarationen und hat das Potenzial, die Vergleiche mehrerer Patienten und Gruppen in scRNAseq-Daten zu vereinfachen.

How to read this thesis

- **Figure axis.** I omit axis annotations in UMAP embeddings, since this single measure simplifies most figures in this thesis. If you are unfamiliar with UMAP and feature plots, I highly recommend reading the 'UMAP' paragraph in section 1.4 and all of section 2.2.

- **Citation style**. The ampersand ("&") is used to concatenate sources that have and have not undergone peer review (bioRxiv preprints, blog posts and websites). Example: [1] & [2, preprint].

- **Cellpypes**. I encourage the reader to look at Figure 5.1 in section 5.2 ahead of time. It describes the cellpypes R package that I am currently developing and which will simplify the classification approach described in this thesis. The R package is unfinished and thus placed in the outlook (section 5.1).

- ***I* and *we*** are used with purpose in this work. *I* is used whenever a thought or analysis can be attributed solely to myself, and *we* when other people contributed (most notably of course, my direct supervisor Simon Anders).

- **'Data'** is plural in the original Latin, but I follow the convention proposed by the Guardian [3] and use singular verbs ('Little data is available'), since they feel more natural than plural verbs ('Little data are available').

- **Footnotes** give further explanations or anecdotal evidence that may be of interest but is not backed by thorough analysis or literature sources.

Contents

List of Figures

Chapter 1

Introduction

1.1 Single-cell RNA sequencing (scRNAseq)

Ribonucleic acid (RNA), and in particular messenger RNA (mRNA), is used by cells to function according to the information stored in their genomes. mRNA molecules encode the genetic information necessary to synthesize proteins, and their abundance is therefore indicative of cell function. The field of transcriptomics is the scientific discipline that studies mRNA abundances of all genes in different cell types, tissues, species and conditions (such as healthy / diseased, wild type / mutant, etc.). Until recently, microarrays and RNA sequencing were the dominant techniques in transcriptomics. Both microarrays and RNA sequencing profile gene expression in 'bulk', i.e. after pooling thousands of cells. In contrast, modern protocols are efficient enough to measure gene expression in individual cells [4–6]. This technology is known as single-cell RNA sequencing (scRNAseq), and has revolutionized the field of transcriptomics. Instead of exploring 'bulk' transcriptomes after sorting cells experimentally, scRNAseq allows to sort cells *in silico* (i.e. during data analysis) by their transcriptional similarities. This makes scRNAseq a powerful technique to investigate cell-cell heterogeneity with unsupervised (data-driven) algorithms. For example, the continuous transcriptional changes during differentiation processes can be explored, as well as different cell states in diverse biological settings (different organs, conditions, species, etc.). While other single-cell 'omics' protocols are being developed (measuring the proteome and/or epigenome instead of transcriptome, or even multiple of these omics modalities – see [7] for an overview), scRNAseq remains dominant in the 'single-cell field', in parts due to commercial adaptions (c.f. section 1.2). Next to the proven potency of scRNAseq data, its high dimensionality and technical noise are challenging to work with, requiring rigorous statistical algorithms. For this reason, the single-cell field is as much a computational discipline as it is an experimental one.

1.2 From cell suspension to count matrix

Every scRNAseq experiments starts with a cell suspension, i.e. solid tissues have to be digested enzymatically to dissociate cells from one another. In this thesis, all analysis starts from the raw counts in the form of a gene-by-cell expression count matrix. To go from single-cell suspension to this count matrix involves the following steps:

1. Cell capture and cell lysis. To increase protocol efficiency, each cell is captured in a small reaction volume, for example water droplets in oil immersion [6], micro-wells [8, 9] or microtiter plates [5, 10–14].

2. Reverse transcription (RT), i.e. conversion of mRNA molecules to complementary DNA (cDNA). It is generally accepted that RT is the limiting step in scRNAseq protocols [1]. The success rate in 10x scRNAseq has been estimated to be between 6 and 32%, depending on which version protocol is used [13, 15, 16]. This means that only a third or less of the mRNA molecules are successfully converted to cDNA, resulting in sparse data (c.f. section 1.3).

3. Sequencing library. The cDNA is fragmented and tagged with sequencing adapters (typically from the commercial 'TruSeq' kit), which is done in a single step ('tagmentation') in scRNAseq protocols to increase efficiency [17].

4. Read (pseudo)alignment and quantification. The raw sequencing files are processed to count matrices with thousands of columns and rows, which contain the measured transcript count for all genes and cells. This step is often done with 10x CellRanger [18], which internally uses alignment with the often used tool 'STAR' [19]. Several alternatives to CellRanger exist, such as dropEst [20], as well as pseudo-alignment tailored to scRNAseq data [21]. These methods, as well as CellRanger, incorporate mechanisms to correct for sequencing errors, especially in barcodes identifying cells and molecules (UMIs, see below).

Protocols to generate scRNAseq data fall into two categories, depending on whether they use unique molecular identifiers (UMIs) (Figure 1.1, explained below). The protocols with UMIs are most relevant to this thesis, and include DropSeq [6] and its commercialization 10x Gemcode [22], next to others [8, 9, 13, 14, 23]. Non-UMI protocols are SMARTseq2 [5] and others [4, 10, 11, 24]. The major difference for data analysis is that data generated with or without UMIs require different noise models (introduced in section 1.3). This is because non-UMI protocols make it impossible to remove amplification bias introduced during polymerase chain reaction (PCR). By contrast, UMI protocols can identify PCR duplicates, as illustrated in Figure 1.1. The scRNAseq protocol reagents destroy the cell, releasing RNA molecules from

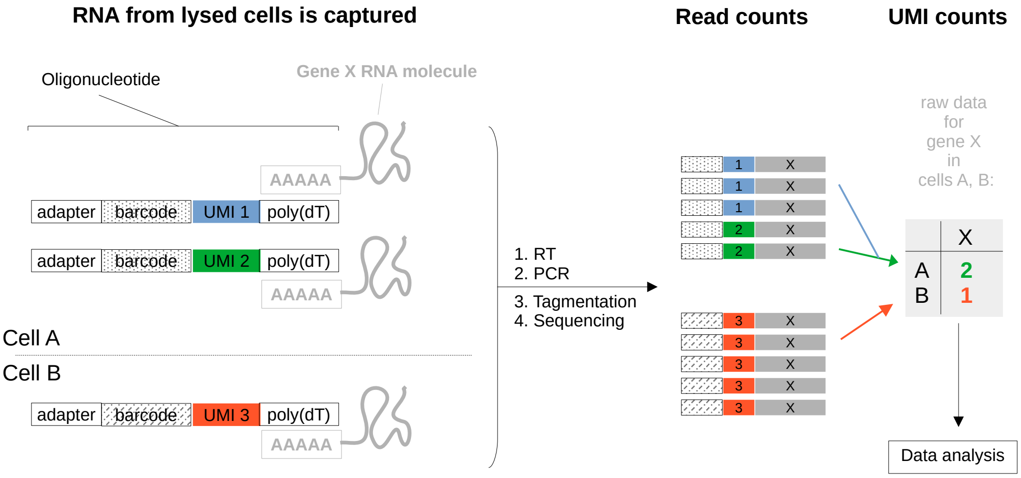


Figure 1.1: Unique molecular identifiers (UMI) count single molecules. The read count in cell B is misleadingly high due to PCR duplicates; UMI counts avoid this amplification bias. For simplicity the schematic shows only two cells and a single gene. More details are in the main text. RT: reverse transcription, PCR: polymerase chain reaction, poly(dT): stretch of around 30 deoxythymine bases, AAAAA: polyadenylation (of unknown length), RNA: ribonucleic acid.

the lysed cells. The single-stranded oligonucleotides have a deoxythymine stretch (poly(dT)) that binds polyadenylated RNA, enriching for mRNA (ribosomal RNA is much more abundant but lacks polyadenylation). This links the mRNA molecule to a sequencing adapter and a cell-specific DNA sequence ('adapter' and 'barcode' in Figure 1.1, left), and importantly to the molecule-specific UMI [1]. After library preparation and sequencing, the read counts contain PCR duplicates. In the example of Figure 1.1, Cell B has misleadingly high read counts (UMI 3, red), masking expression differences to cell A. This happens because PCR is a non-linear process, meaning a single mRNA molecule gives rise to different numbers of sequencing reads for each cell and gene. The final UMI counts (Figure 1.1, right) do not suffer from PCR bias, because PCR duplicates can be identified and collapsed into a single UMI count. This has implications for data analysis, as introduced below (section 1.3).

I note that all data used throughout this thesis was generated with UMI counts, as discussed in section 4.5. Until recently, UMI protocols had the disadvantage that they only counted the ends of mRNA molecules, lacking whole-transcript coverage. Of note, the recently published SMARTseq3 protocol [12] combines whole-transcript coverage with UMI counts, enabling the removal of PCR duplicates as well as investigating isoform usage and allele specificity.

[1] To understand oligonucleotide synthesis, I recommend the DropSeq paper [6]. In general, 10x Genomics offers clear figures on their website as well.

1.3 Statistical models for scRNAseq counts

The low transcript capture rate mentioned above, combined with the small mRNA content of a single cell, result in sparse data. One illustrative observation is that scRNAseq data has the same characteristics as bulk RNA samples after diluting it to the same, low concentration [25, Fig. 1 therein]. Kharchenko *et al.* [26] thoroughly described the high amount of zeros in the data (around 80%) and an initially puzzling gap between zeros and non-zero counts, making the data appear almost bimodal, if not binary [2]. In these early years, when only non-UMI protocols were available, the term 'drop-out' was coined and is still used today, but with multiple meanings. In this work, I refer to 'drop-outs' as zero sequencing counts when instead a non-zero count had been expected. For example, all T cells are likely to have had CD3E mRNA molecules at time of lysis, so observed zeros likely represent 'drop-out' events rather than indicating actual absence of mRNA molecules. Important terminology in this context is that for cells with zero sequencing counts, we can only say that the respective gene is not *detected* in the cell. It is important to avoid claiming the gene was not *expressed*, as in no mRNA molecules were present. This is because in scRNAseq data, zero sequencing counts can represent 'true' zeros (gene was not expressed and thus not detected) or drop-outs (gene was expressed, but not detected).

The state-of-the-art of modeling scRNAseq data today is to use the Poisson distribution and its derivatives (allowing for over-dispersion, i.e. larger variances than the Poisson) to directly model the raw sequencing counts [3]. The consensus to model non-UMI data is to use the zero-inflated negative binomial (ZINB) distribution [27–31]. This model represents counts as a mixture of a negative binomial distribution for detected values and a drop-out component to model additional zeros in the data. This drop-out component was implemented as a low-rate Poisson random variable by Kharchenko *et al.* [26], while the more recent tool ZINB-WaVE uses the dirac function, modeling a binary 'decision' between zero and non-zero count [27].

In contrast to non-UMI counts, data generated with UMIs are not zero-inflated [32]. In other words, the number of observed zeros is not larger than expected from the small amount of mRNA input material. Control experiments without any biological variation show good agreement with the

[2] This gap is generally accepted now to be due to amplification bias during library preparation (non-UMI protocols only): Polymerase chain reaction is non-linear, i.e. the amplification success can differ by a few orders of magnitude between cells and between genes. Directly after reverse transcription, the difference between zero and non-zero values might not be large, but amplification increases this difference. Therefore, the same gene is not detected at all in some cells, while high numbers of sequencing counts are observed in other cells.

[3] Some algorithms model normalized counts instead of the raw counts. In this scenario, the statistical distribution is unknown and has to be approximated, e.g. with the log-normal distribution.

Poisson model [32] & [33, preprint], with the exception of highly expressed genes [33, preprint] (the authors speculate this is due to efficiency noise, i.e. droplet-to-droplet differences in capturing efficiency). The Poisson model therefore fits well in many settings [1, 32, 34] & [33, preprint]. In the presence of strong biological variation (e.g. for heterogeneous cell types, or modeling expression across multiple cell types), UMI counts are usually modeled as negative binomial random variable [1, 35, 36]. This is because the negative binomial distribution is overdispersed, i.e. allows for more variance than the Poisson distribution. This is achieved with a Gamma-Poisson hierarchy: Conceptually, expression strengths sampled from a Gamma distribution are used as Poisson rates to model how sequencing counts were generated [4]. I note that also for UMI data, some methods use a log-likelihood ratio test to determine whether ZINB is necessary or if the negative binomial model suffices [36]. Thus, UMI data is modeled by Poisson, negative binomial or ZINB, depending on how much biological variation is to be expected.

In summary, the state-of-the-art model for non-UMI data is to use a zero-inflated negative binomial distribution (ZINB) [27], while standard negative binomial or even Poisson are used for UMI data.

1.4 Frequent analysis steps

Here, I introduce important terms and analysis procedures, such as 'total UMI', 'feature plot', normalization, PCA and UMAP. Once a count matrix has been obtained (see section 1.2), the analysis steps listed here are applied prevalently to scRNAseq data. They are shared by many commonly used analysis tools, such as Seurat [37], Scanpy [38] or monocle2 [39].

1. **Normalization.** The number of sequencing counts typically varies between cells by multiple orders of magnitude. To make expression rates comparable between cells, the raw counts have to be normalized to the total number of counts. To this end, the counts are often divided by each cell's sum of all (UMI) counts, which in this work I will refer to as totals, total UMI or library size. In particular, I do refer to the totals as 'size factors', to distinguish them from numbers centered around 1. Other approaches to normalization include using the geometric mean [40, Garnett] computed after masking all zeros, or regressing the dependency of observed counts and a cell's totals [41, scTransform]. Following size-factor normalization, variance-stabilizing

[4] For this reason, the negative binomial distribution is also known as Gamma-Poisson distribution. I note that in non-sequencing techniques that obtain continuous read-outs instead of discrete sequencing counts, gene expression is modeled with a log-normal instead of the Gamma distribution. With the correct parametrization, however, both distributions are similar in shape and the Gamma distribution is analytically more tractable.

transformations are applied. This aims at giving the same 'influence' on later processing steps to all genes, and typical choices are logarithmic ('log-normalization' [42, Seurat vignette]) or square root-based transformations such as Freeman-Tukey [33, preprint].

2. **Filtering cells.** This step is often termed quality control (QC). Most typically, cells with extremely low or high totals are excluded, assuming they are empty beads, cell-cell doublets or unwanted outlier cells. Also, cells with high mitochondrial mRNA content are excluded, assuming most cytosolic mRNA was leaked after cell damage.

3. **Filtering genes.** This step is also known as feature selection, and aims at identifying genes with a variance exceeding the measurement noise. Such 'highly-variable genes' contain biological information on top of the technical noise [25]. Multiple methods have been formally proposed [1, 25, 43, 44], and many scRNAseq analysis tools use their own implementations. For example, monocle2 [39] and Seurat [37] both select overdispersed genes according their own methods. I note that Seurat [37] selects highly-variable genes after log-normalization, which has been criticized [1], next to general concerns towards log-normalization [45].

4. **Principal component analysis (PCA)** computes linear combinations of the supplied features (genes), such that the first components (new features) maximize the variance amongst the data points (cells). While PCA formally rotates the coordinate system, preserving all information in the data, discarding later components with lower variance effectively reduces technical noise [46] & [33, preprint]. This is because biological variation is expected to accumulate in the first few components with the highest variance, since biological variation is often larger than technical noise and on top correlated among genes. Typically, the first 20 to 50 components are retained in practice, and the exact number is often chosen by the user after exploration [42, Seurat vignette]. As input for PCA, it is common to use gene expression values after normalization, transformation, centering and scaling. Alternatively, GLM-PCA [1] is a statistical generalization so that PCA can directly operate on raw UMI counts.

5. **Nearest neighbors.** For each cell, the k-nearest neighbors (kNN) are identified from pairwise cell-cell distances. This is an important step, since kNN are used for most analysis end-points, such as clusters and UMAP embeddings (see below). Mathematically, kNN information can be represented in a nearest neighbor graph, where each cell is a node that is connected to its kNN with edges. This representation is used in UMAP, clustering (see below) and also the classification tool presented in this thesis (see chapter 3). In scRNAseq, kNN are typically found using the Euclidean distance computed on the PCA embeddings.

6. **UMAP and tSNE embeddings.** scRNAseq data is finally visualized with non-linear embedding methods (non-linear meaning that the transformation can not be represented in mathematical matrix notation). Instead, UMAP [47] and tSNE [48] use nearest-neighbor graphs (introduced in section 1.7.2) to create low-dimensional embeddings (2D, most typically) where cells with similar transcriptomes are placed into close proximity of each other. I note that distances within these embeddings are arbitrary and controlled by UMAP's spread and min_dist parameters. In particular, cell-cell distances in UMAP embeddings are not quantitative due to the non-linear nature of the algorithm. An early comparison between UMAP and tSNE suggested UMAP was faster and better at preserving global structure for scRNAseq data [49]. However, tSNE is able to preserve global structure when parameterized correctly [50] and runs faster than UMAP since the introduction of FItSNE [50, 51]. One common application of UMAP or tSNE embedding are 'feature plots'. Coined by the Satija lab [42, Seurat vignette], the term 'feature plot' denotes embeddings colored by the expression strength of a single gene. This thesis contains many such feature plots, as they are crucial when assigning cell type labels to cells.

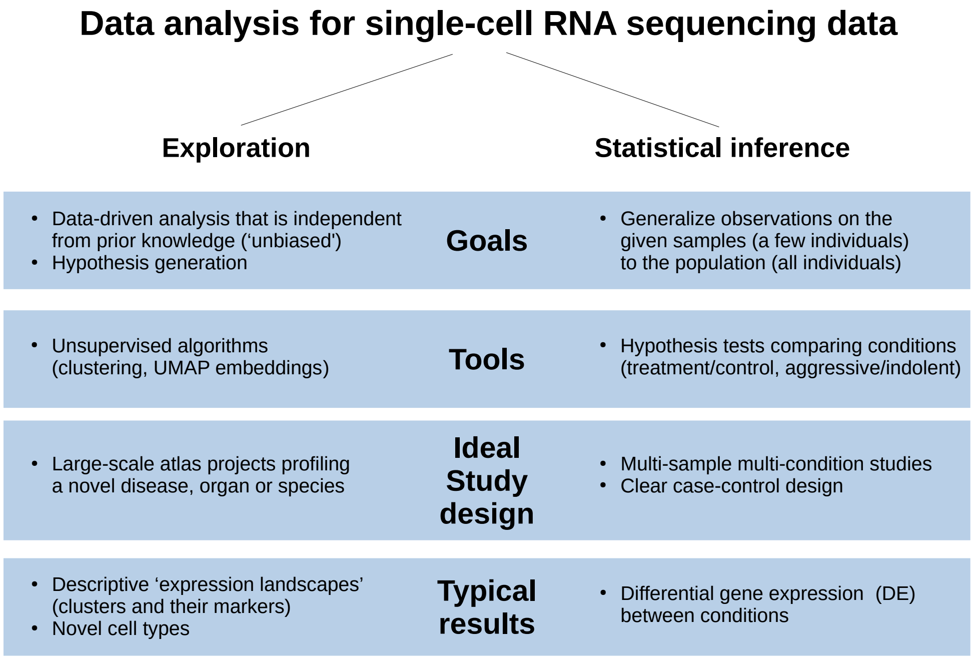
7. **Clustering.** Clustering cells by transcriptomic similarity is introduced thoroughly in section 1.7.2. In the presence of differentiation processes or similar transcriptomic gradients, pseudotime (computed e.g. with Monocle2 [39]) can be a more goal-oriented alternative to using discrete clusters.

1.5 Inference with scRNAseq data sets

Inference and exploration are two branches of statistics, and Figure 1.2 gives an overview over their role in scRNAseq data analysis. Exploring data involves visualization, clustering and other unsupervised algorithms. Inference aims at generalizing from the observed samples (patients, mice, cell culture dishes, *etc.*) to the wider population (all patients, mice or cell culture dishes). Put in simple terms, exploration generates hypotheses, while inference tests them [5]. The differences between inference and exploration are central to the classification approach I propose in section 3, so I introduce related terminology in the following.

scRNAseq is most notorious for its capability of unsupervised exploration. Indeed, several large atlas projects have already been completed to this end: Tabula muris [52], the mouse cell atlas [8], the mouse nervous system [53],

[5] Inference uses statistical hypothesis testing, for which a full introduction is beyond the scope of this work. Briefly, a hypothesis is compared to a simpler alternative, the null hypothesis, in terms of how likely both were to 'generate' the observed data.



Data analysis for single-cell RNA sequencing data

Exploration		Statistical inference
• Data-driven analysis that is independent from prior knowledge (‘unbiased’) • Hypothesis generation	**Goals**	• Generalize observations on the given samples (a few individuals) to the population (all individuals)
• Unsupervised algorithms (clustering, UMAP embeddings)	**Tools**	• Hypothesis tests comparing conditions (treatment/control, aggressive/indolent)
• Large-scale atlas projects profiling a novel disease, organ or species	**Ideal Study design**	• Multi-sample multi-condition studies • Clear case-control design
• Descriptive ‘expression landscapes’ (clusters and their markers) • Novel cell types	**Typical results**	• Differential gene expression (DE) between conditions

Figure 1.2: Exploration and inference with scRNAseq data. Their differences in goals, tools, study design and results call for separate cell type classification methods, as argued in this thesis. Multi-sample mutli-condition studies are also referred to as ‘cohort’. The bullet point in each category are non-exhaustive examples.

the malaria parasite [54], the brain of fruit flies [55], and the (yet to be finished) human cell atlas [56]. In a few cases, the major exploratory goal of discovering novel cell types has been achieved [57–59]. In recent years, a new trend has entered the field: Cohort studies started using scRNAseq for group comparisons, i.e. with inference instead of exploration as main goal. For example, Schirmer *et al.* [60] compared *post mortem* brain tissue samples of 21 patients with and without multiple sclerosis. The same research group did a similar comparison for autism spectrum disorder [61], with 31 patients (half of which with autism, the other half were control individuals). A similarly large cohort was recently reported for inflammatory bowel disease, obtaining scRNAseq data for 18 patients with ulcerative colitis and 12 control individuals. Other diseases were studied in similar, although smaller, cohorts. I myself have recently contributed to a study in which patient samples from aggressive lymphomas were compared to indolent and tumor-free control samples (12 patients in total) [62]. Azizi *et al.* profiled eight breast cancer patients with scRNAseq [63]. For half of them, tumor-free breast tissue was included, which can be used for paired comparisons of cells from healthy and malignant tissues. Another small cohort used scRNAseq on five lung tumor patients, four of them with healthy control tissue [64].

These cohort studies pursued the same general computational strategy: finding cell types by transcriptomic similarities, comparing the same cell type across two conditions and reporting the inference results (most prominently, lists of differentially expressed genes). Thus, as Crowell *et al.* put it in their 'muscat' manuscript [65], cell type classification methods in this context are "*in silico* sorting approaches for multi-subpopulation multi-sample multi-condition scRNAseq datasets" [65]. In the past years, the experimental units were not samples (patients, mice or other batch units), but instead individual cells [65]. This 'pseudoreplication' leads to misleadingly low p-values (for example with hundreds of genes with $p \ll 0.05$), because the biological question was asked imprecisely. Specifically, inference aims at generalizing from a few observations (few patients) to the wider population (all patients). The intended and biologically meaningful population in the scRNAseq cohort setting would be samples (patients, mice, etc.), not cells, so samples have to be the experimental unit in statistical testing. To discriminate differential testing on cells from that on samples, Crowell *et al.* [65] refer to the latter as 'differential state' analysis.

I now clarify the terminology used in this thesis. In contrast to Crowell *et al.* [65], I refer to multi-condition multi-sample comparisons as differential expression testing, not differential state. I will use it repeatedly as primary example for inference questions addressed with scRNAseq cohorts. In this thesis, such studies will often be referred to as inference-oriented studies, because their goal is more the comparison between treatment groups than the cell type discovery through exploration of large atlas data sets.

In contrast to differential gene expression between conditions, and of little significance to my work, 'marker gene identification' is the process of comparing clusters within samples using cells as experimental unit. The p-values are, as in pseudoreplication above, often misleadingly low. The problem here is not 'pseudoreplication', but instead the circularity of the test: Differences in gene expression are first used to define clusters, but then tested for the null hypothesis that they do not exist. A procedure to avoid this circularity has been proposed [66], but I note again that differential expression testing, not marker gene identification, is the question addressed in this thesis.

In summary, inference with scRNAseq cohorts are an unsolved problem that requires novel statistical frameworks [65]. Inference (such as differential gene expression) is orthogonal to assigning cell type labels [65], otherwise p-values are unreliable due to the circularity [65, 66]. In this work, I argue that not only the statistical tests for inference, but also the cell type classification method itself requires innovation. I propose a novel method for inference-oriented cell type classification, which I motivate in more detail in section 3.1. The method I propose is conceptually related to gene smoothing, and aims at assigning cell type labels, so I introduce these two fields in the next two sections.

1.6 Gene smoothing and imputation

In this section, I introduce existing smoothing and imputation methods for scRNAseq expression values. I note that in my work, the smoothed expression is not a goal on its own, but instead is used for the binary decision of whether a cell is positive for a gene or not. This is in contrast to the existing methods, which often seek to improve data quality in general. The intend of this multi-purpose smoothing is to have a positive impact on downstream analysis such as UMAP, clustering, gene-gene correlations, and others. The circularity of improving a data set with the very same data set has been pointed out by Andrews *et al.* [67], who also showed for the majority of these methods to introduce false signals, such as spurious correlations between genes [67]. Therefore, results obtained with the denoised counts produced by the methods I now introduce require extra caution to avoid over-interpretation.

To avoid confusion, I find it necessary to define the terms smoothing, denoising and imputation, because they are used interchangeably by some authors in the context of scRNAseq gene expression [67–69]. Throughout this thesis, I refer to imputation as the process of replacing zeros by more likely values. Specifically, imputation does not alter non-zero expression values. This is in contrast to smoothing or denoising, which I use interchangeably here. In smoothing, all values are replaced by more likely expression values, irrespective of whether they were zero or non-zero values.

Imputation The existence of drop-out events [26] sparked the development of imputation methods, such as CIDR [70], DrImpute [31] and scImpute [71], and has most recently even been addressed with autoencoders [72]. Of note, these methods assume that non-zero (detected) values do not require any correction. In reality, however, there is no fundamental difference between zero and non-zero counts: they both arise from the same noisy measurement process. A more recent practice is therefore to smooth all values, rather than imputing only the ‘missing’ values.

MAGIC One of the first smoothing methods was MAGIC [68], which smooths gene expression across nearest neighbors. The data sparsity makes Euclidean distance noisy, so in order to find a cell’s k nearest neighbors (kNN) more precisely, MAGIC assumes a diffusion process. Conceptually, random walks are simulated from a given cell to all other cells. This random walk can make the nearest neighbor assignments more precise: Technical noise produces random neighbor edges (c.f. 1.4) between two cells, while biological signal gives rise to systematically connected neighbors. Diffusion processes in scRNAseq data emphasize these systematic connections, because true neighbors are also connected to the neighbors of neighbors [73]. The diffusion principle had previously been adapted for scRNAseq to compute better low-dimensional embeddings (see the destiny package [74]), and has more recently been applied for label transfer between scRNAseq samples with Conos [73] (introduced further in section 1.7.3). While conceptually related, diffusion algorithms such as MAGIC do not directly simulate individual random walks in practice. Instead, they adapt the simpler mathematical solution described in [74]. Specifically, the transition probability from one cell to its neighbors is computed as an exponential function of the Euclidean distance to these neighbors. By computing the n$^{\text{th}}$ power of this cell transition matrix ('Markov transition matrix' in [68]), the probability to 'walk' from one cell to all others with n steps is computed. Gene expression is then smoothed by multiplying the normalized gene expression matrix with this cell-cell transition matrix, which contains the transition probabilities from all cells to all other cells after a random walk with n steps. In other words, a weighted average for gene expression is computed across the kNN as defined by the diffusion process. I note that the authors of MAGIC call this imputation [68], but it clearly is smoothing by the terms used in this thesis, because non-zero values are also altered. The goal of smoothing with MAGIC is to improve downstream analysis such as UMAP and clustering [68]. Furthermore, the authors propose to investigate the expression and correlation of lowly expressed genes, such as transcription factors [68, Fig. 5 therein]. This is not without the risk of overinterpretation: a few or even single expressing cells might be enough to suggest strong correlation after smoothing. Such spurious correlations are artifacts introduced by all existing smoothing methods [67], and MAGIC in particular [35].

kNN-smoothing 1 and 2 kNN-smoothing and kNN-smoothing 2 are more recent methods [33, preprint] that also average the expression profiles of nearest neighbors, with a special emphasis on how these neighbors are found. While MAGIC uses a diffusion process, the kNN-smoothing algorithms take an iterative approach. Each cell is first averaged with its closest neighbor, and these minimally smoothed transcriptomes are used to again find kNN for each cell. Then, the first three kNN are averaged for each cell, and again the kNN information is refined. In each step, more neighbors are used until a preset value is reached. The authors claim that alternating between kNN search and kNN averaging makes the smoothing more precise. To compute Euclidean cell-cell distances for the kNN search in each step, the first kNN-smoothing algorithm uses normalized gene expression, while kNN-smoothing 2 uses the PCA embedding. This is the only difference between version 1 and 2 [33, page 10, preprint], and helps to resolve highly related cell types that differ in only a few genes. The authors find that hierarchical clustering, which performs poorly on scRNAseq data, produces sound results after smoothing the entire transcriptome (more precisely: 1000 highly variable genes). Furthermore, manual thresholding of smoothed marker gene expression can be used to identify major blood lineages (T cells, B cells, dendritic cells and monocytes). Finally, the authors observe that the smoothed transcriptomes are more compact in PCA space than the original ones, and note that stronger dependencies between cells are introduced with increasing smoothing bandwidth (number of kNN). In line with this, Andrews *et al.* [67] report smoothing artifacts introduced by kNN-smoothing. As for MAGIC, they note that kNN-smoothing is circular because it aims at improving a given data set with the very same data set. In this process, Andrews *et al.* remark that "no new information is gained, making it analogous to simply lowering the significance threshold of any statistical test applied to the data" [67]. In other words, kNN-smoothing introduces many false-positive findings in downstream inference tests (gene-gene correlations, cell-type markers and differentially expressed genes) [67]. I note that the kNN-smoothing approach has not passed peer review since the preprint was published two years ago.

Autoencoders: DCA, scVI and SAVER-X One last group of smoothing methods employ neural networks in the form of autoencoders. DCA [36] is a classical autoencoder, i.e. a neural network that tries to reproduce observed data as accurately as possible with the constraint of representing all cells with only a small number of latent features. Architecturally, an autoencoder has a middle layer with only a few artificial neurons, where the exact number is a hyperparameter to be optimized and depends on the data set complexity. In one example, the authors chose DCA's 'bottleneck layer' to contain 32 neurons for the data set under study [36]. Thus, the network is trained to reconstruct the observed expression of thousands of input genes as accurately as possible in its output layer, while being forced to represent the cells as 32-dimensional latent vector in its middle layer. These 32 artificial features ('latent variables') are thus forced to capture different aspects of variation in the data, and neural networks are able to do so in a non-linear way [6]. The output layer represents smoothed gene expression values, because the autoencoder's low-dimensional bottleneck forces it to focus only on the strongest signals, because the abundant but uncorrelated noise can not be accommodated in the limited information encoded in a few (dozen) bottleneck neurons. Since biological variability is typically stronger and additionally correlated, in contrast to random technical noise, the autoencoder is expected to denoise by encoding mostly these relevant cell-cell variations while discarding unwanted variation. Andrews *et al.* [67] also report strong artifacts in gene expression values smoothed with DCA, for the same circularity reasons as for MAGIC and kNN-smoothing.

scVI [69] also uses neural networks to represent cells in low-dimensional latent space. In contrast to DCA, gene expression smoothing is only one of several tasks it is designed for, next to clustering, differential expression testing

[6] Non-linear means in this context that a change in gene expression is not necessarily proportional to a change in the latent variable's value. In contrast to linear methods such as PCA, non-linear operations can not be represented by matrix notation.

and data set integration (batch effect removal). I note that as in the case of MAGIC, the authors of scVI use the term ‘imputation’ contrary to the definition of this thesis, according to which it should be called ‘smoothing’ because non-zero values are also changed. I note that scVI was not tested for false signals by Andrews *et al.* [67] because it had not been published yet. Still, it provides transcriptome-wide gene smoothing without introducing any new information, which makes it likely to also introduce artifacts as the other methods.

SAVER-X is designed to smooth gene expression data more robustly by using information from related reference data sets [30]. The algorithm is the successor of SAVER and is also smooths gene expression by predicting it from correlated genes. These gene-oriented approach is a key difference to kNN smoothing methods such as MAGIC and kNN-smoothing 2. Specifically, SAVER-X [30] trains an autoencoder on external data (the ‘X’ is for external), to learn weights for the gene-gene correlations. This is in contrast to its predecessor SAVER, which uses penalized regression (the LASSO) to regress gene UMI counts on other genes as predictors. The autoencoder of SAVER-X is implemented using the DCA library, and perhaps for this reason also uses 32 neurons in the bottleneck layer. The reference data sets for training are ideally related to the ‘query’ data set that is to be denoised. Conveniently, multiple data sets can be used from human, mouse or both (relying on 15,494 homologous genes). While only UMI data sets can be denoised, SAVER-X can train the autoencoder on non-UMI data. In this case, they use transcripts per million (TPM) and use the ZINB likelihood, even though TPM are continuous rather than discrete counts. In the benchmarking paper by Andrews *et al.* [67], SAVER-X has not been included yet. Its predecessor SAVER introduced less spurious correlations than other methods [67], and the authors hope that SAVER-X breaks the circularity of transcriptome smoothing by adding external information in the form of reference data sets [30].

Other smoothing approaches I again point out SAVER [35], which I briefly introduced together with SAVER-X in the previous paragraph. By regressing the gene of interest with other genes as predictors, SAVER does not rely on kNN information at all. Perhaps for this reason, Andrews *et al.* [67] find that SAVER introduces fewer artifacts than other transcriptome-wide smoothing approaches [67]. Of note, SAVER not only returns the smoothed expression values, but also estimates of their uncertainty [35].

Finally, there are more computational methods that are conceptually related to smoothing of scRNAseq data. For example, MetaCell [75] aims at increasing the signal-to-noise ratio by creating ‘mini-bulks’. For this, it finds highly related cells and pools their raw UMI counts, in order to obtain MetaCells - fewer than the original cells, but with larger library sizes and thus less technical noise. Other approaches attempt to iteratively smooth and cluster the data [76–78].

Summary and reference to my work What unites all smoothing approaches described above is their ambition to provide continuous expression values for the entire transcriptome. The goals of smoothing are often nonspecific and thus rather extensive: Improving UMAP, clustering and finding correlations between weakly expressed genes are frequently named as motivation. As pointed out by Andrews *et al.* [67], smoothing introduces artifacts into the data, so the denoised expression values should be used with caution. Here, we use gene expression smoothing with a very specific goal in mind: For a few selected marker genes, we want to determine which cells are positive or negative for these markers. This binary decision does not make any guarantees for the derived continuous expression values, but instead uses it as a proxy to quantify uncertainty when assigning cell type labels. This is in contrast to the existing approaches, which are multi-purpose, whole transcriptome smoothing methods.

1.7 Assigning cell type labels in scRNAseq data

Here, I introduce cell type classification methods from the single-cell omics field. To organize the ever-increasing number of algorithms, I group classification methods by their core principles as follows. The largest body of tools assumes that clusters are cell types, which would mean the classification task reduces to finding optimal groupings of cells with an unsupervised algorithm. The second group extends this idea by transferring cell type labels from a reference data set, typically annotated using clustering, to a new query data set. Methods from the last group are independent from clusters: They either use gene expression signatures (dozens of genes or more), or they elect to work with a minimal set of marker genes. This last category of methods is conceptually most related to my own work and its small size already shows the need for novel ideas. I start this chapter, however, with a short review of the cell type concept itself.

1.7.1 The cell type concept

The cell type concept is not clearly defined and often debated [79, 80]; for a collection of different opinions see for example [81, comment]. It has been proposed to assign cell identity based on phenotype, lineage and state [82]. Another opinion is to introduce a kind of 'periodic table' for cell types. As periods, groups and isotopes are used in the periodic table of the chemical elements, it is proposed to use lineages, differentiation stages and cell states analogously [83]. Others propose to label cells according to (major) cell type, developmental state, activation status, function and signaling pattern [84]. In practice, the criteria to define cell types evidently depend on the methods used. For flow cytometry ("FACS"), a few protein surface markers are hand-selected, while researchers that employ sequencing technologies tend to use genome-wide cellular states (scRNAseq, ATACseq, etc.). Those with a preference for imaging and lineage tracing assays have argued that these molecular phenotypes are secondary to ontology [85], i.e. that a cell's ancestry and tissue of origin are the decisive criteria for assigning cell types. For the field of evolutionary biology, the homology between species has been proposed as central criterion to define cell types [86]. Taken together, this global perspective on cell biology suggests a rather subjective nature of cell type definitions. A more goal-oriented approach is to use the more general term 'subpopulation', defined as any group of cells for which it is interesting to ask inference questions [87]. Here, a typical inference question is for example to find differential gene expression between conditions. In my work, I adhere to this goal-oriented scheme of cell type definitions.

1.7.2 Labels from clusters

The idea that clusters are cell types and vice versa is so prevalent in the scRNAseq literature that both terms are often used interchangeably. Indeed, several methods have been developed specifically to take clusters and find corresponding cell type labels [88], naturally assuming a direct mapping between them. Also, in a recent comparison, cell type classification methods are evaluated based on how well they recapitulate clustering results reported in the original papers [89]. I note that this concept of assuming clusters equal cell types has limitations, which I point out at the end of this section.

Clustering algorithms for scRNAseq data can be divided into four groups, which I will introduce in turn by discussing representative methods for each in the following. Briefly, the four groups formulate the clustering objective differently. k-means and its derivatives consider the density in feature space. Hierarchical clustering considers only the cell-cell distances. Graph-based clustering also uses cell-cell distances to build a nearest neighbor graph, on which the communities (clusters) are then detected. For modularity-based algorithms, this graph-based community detection works by considering the density of graph edges. The forth group is a heterogeneous collection of methods, some employing deep learning, others smoothing and yet others gene-regulatory networks for clustering. For the distance-based algorithms (hierarchical and graph-based), I note that a popular metric is the Euclidean distance computed on the PCA embeddings. Other metrics in use for scRNAseq data are the cosine similarity and Pearson distance (1 minus Pearson correlation), which are formally not distance functions because they do not fulfil the triangle inequality.

k-means clustering Methods relying on k-means clustering include SC3 [90], RaceID [91] and SIMLR [92]. A recent review [46] mentions k-means as the most popular clustering technique, but I disagree due to the unbroken popularity of Seurat [37] and Scanpy [38]. k-means clustering computes a user-defined number of k centroids in feature space (typically gene expression or PCA embeddings). Cells are assigned to the closes centroid, centroids are recomputed and this process is iterated until convergence. The result often depends on the random initialization of centroids, and so k-means algorithms have to be re-run multiple times to avoid local minima. Classical k-means tends to find equally-sized populations [46], and several adaptions were taylored for scRNAseq data. For example, RaceID includes outlier detection, so that rare cell types are also identified [91]. SIMLR also uses k-means [46] and has proposed an improved distance measure adapted for scRNAseq data [92]. SC3 uses consensus clustering [90] across different pre-processing steps. Specifically, different distance metrics are used, and cells are grouped together if they were nearest neighbors in many different settings.

Hierarchical clustering Hierarchical clustering is now rarely used for scRNAseq data. SINCERA is an early pipeline designed for non-UMI data [93], which uses hierarchicl clustering with Pearson correlation. As Kiselev *et al.* [46] point out in their review, however, hierarchical clustering is prohibitively slow on large data sets, because computation time increases quadratically with the number of cells [46]. Also, it does not perform well on the noisy distances derived from scRNAseq data, which is why several improvements have been proposed. CIDR imputes zero expression before hierarchical clustering, i.e. replaces zeros with continuous values [70]. pcaReduce [94] recomputes PCA after each merging of groups. It thus uses multiple iterations to ensure smaller populations can be detected in spite of technical noise [46]. Furthermore, hierarchical clustering can be used on smoothed expression data [33, preprint].

Graph-based clustering Many popular methods for scRNAseq data analysis [18, 37, 38, 40] use graph-based clustering. Here, the nearest neighbors of each cell are identified and represented in graph notation, i.e. a list of edges that connect nodes (cells). The simplest kNN graphs are unweighted, i.e. with binary edges (1: neighbors, 0: not neighbors), and undirected (if cell A is a kNN of cell B, the inverse is assumed to also be true). Clusters are then detected using community detection algorithms on the kNN graph. Popular algorithms for community detection on neighbor graphs are the Louvain [95] and Leiden [96] algorithms. For a good description of these algorithms, see also [97, predecessor of Leiden]. Both Louvain and Leiden maximize the 'modularity', which compares the observed with the expected density of edges within communities [95, 96]. The modularity can be defined with [96, Leiden] or without [95, original Louvain] a resolution parameter. In particular, the original Louvain algorithm [95] had a resolution limit, which resulted sometimes in over-sized and/or poorly connected communities [96]. This was improved with the introduction of the resolution parameter [97, 98] & [99, preprint], for which the user can select smaller values to obtain more communities and *vice versa*. In modern implementations (Seurat [37], Scanpy [38]), the Louvain algorithm with this resolution parameter is used, while the classical version without it can still be found in tools using the igraph R package, such as Garnett [40]. Another improvement for graph-based clustering that is used by Seurat [37], Scanpy [38] and also Garnett [40] is the concept of shared nearest neighbors (SNN). As noted by Seurat's authors [42, vignette], SNN were introduced to scRNAeq data by PhenoGraph [100] and SNN-cliq [101]. For this, conventional kNN are first found, typically with Euclidean distance in PCA embeddings. SNN are then computed as the overlap of these. For example, Seurat and Garnett both use the Jaccard index and 20 kNN as default ([42, vignette] and [40]), which means that for each pair of cells the number of shared kNN is divided by 20 and used as edge weight in the SNN graph. The rationale of SNN is that they are more robust in high dimensions than conventional kNN [101]. Other methods relying on graph-based clustering are for example Scanpy [38], a python analysis workflow, CellRanger [18], 10x alignment and pre-processing pipeline, Garnett [40], introduced further in section 1.7.4, monocle [39], an analysis workflow with a focus on pseudotime computations, and SCCAF [102], which proposes self-projection clustering to find ideal clusters. In general, graph-based clustering has the property that the number of clusters does not have to be specified by the user, in contrast to k-means. Still, the popular Leiden and modern Louvain algorithms have the resolution parameter, to determine cluster size indirectly. I note that when applied to scRNAseq data, undirected graphs are commonly used but not necessarily appropriate. For example, outlier cells would not be among the first few neighbors of their neighbors, which is ignored by undirected edges.

Other clustering approaches Next to k-means, hierarchical and graph-based clustering, more unsupervised algorithms exist that do not fit into any of the above categories.

For example, SCENIC [103] derives gene-regulatory networks for clustering. Specifically, SCENIC computes for each cell the activity of so called regulons, and then clusters on regulon activity instead of gene expression data. Regulons are found by correlating the expression of known transcription factors with various genes. Significant correlations are filtered by requiring the transcription factor binding motif to be enriched in proximity to the gene body. The output of SCENIC is a binary matrix with regulon activity (active or not) for each cell, which can be used as input for clustering. The authors note that linking many genes by the transcription factors that regulate them, SCENIC can overcome the sparsity of the data [103]. Several tools that use gene regulatory networks for scRNAseq data exist, see [104] for a review, and two recent benchmarking papers [105, 106].

Moana [78, preprint] is conceptually similar to clustering since it is purely data-driven. Moana uses kNN-smoothing 2 [33, preprint] to first aggressively smooth transcriptomes. Here, aggressive means using a large bandwidth, i.e. Moana pools UMI counts from 128 kNN during this step [78, supplementary methods, preprint]. This is followed by manual inspection of the first two principal components (PCs) computed on the smooth transcriptomes. If at least two distinct populations are evident, density-based clustering (using DBSCAN) is used for an initial grouping of cells on the first two PCs. The initial cell labels obtained this way are used as training labels for a support-vector machine (SVM) with linear kernel. In real data sets, these initial cell labels typically correspond to major cell lineages (e.g. T, B and myeloid cells) [78]. Further subtypes (T cell subsets, etc.) can then be separated by repeating these three steps (smoothing, PCA, SVM) for each subpopulation separately. The authors claim this works well even for integrating data sets, due to the excessive smoothing. In summary, Moana thus automates labeling training cells for an SVM classifier applied to PCA embeddings. It can be said that this original approach uses supervised learning (SVM) in a completely unsupervised setting (training cells come from DBSCAN after smoothing and PCA). I note Moana has not passed peer review since the preprint was published three years ago.

It has also been proposed to identify clusters and cell types using neural networks [107]. Interestingly, a recent paper implemented different architectures and reports that deep learning does not outperform classical machine learning [108] on the cell type classification task. The authors note that cell types are rather simple, rule-based conventions and thus the complexity of non-linear neural networks might not be required to solve it.

Multi-omics protocols and clustering For explorative single-cell studies, another group of algorithms is of particular interest. Seurat [109], MOFA+ [110] and a few other approaches [111, 112, MATCHER, LIGER] are able to integrate different modalities measured with multi-omic protocols. For example, the same protocols measures transcriptome and surface proteome for the same cells [113–115, CITEseq, REAPseq, BioLegend totalSeq], others measure transcriptome and chromatin accessibility [116–118, sci-CAR, SNARE-seq, 10x Multiome], or even the two with DNA methylation on top as third molecular layer [119, scNMTseq]. In order to separate this type of multi-omics data from having multiple, independent data sets using different omics protocols, the Satija lab coined the term ‘modalities’ to denote transcriptome, proteome, chromatin accessibility, etc. from the same individual cells. For an extensive overview of published protocols, see [7]. As a representative example, I now discuss the weighted-nearest neighbor approach (WNN) published together with Seurat v4 [120]. Specifically, nearest neighbors are first identified separately on the available modalities (transcriptome, surface proteome, chromatin accessibility, etc.). The WNN method then weights the two modalities by estimating for each cell which neighbor graph has exclusive information. This is possible by the following observation described by the authors [109, preprint]: if a cell's kNN computed on the transcriptome are remote neighbors according to the proteome, but not *vice versa*, this means the proteome has exclusive information for this particular cell that the transcriptome is lacking. For this cell, the proteome receives a larger weight than the transcriptome. The resulting WNN graph thus automatically focuses on the modality with the better signal-to-noise ratio.

Clustering limitations The core idea that clusters equal cell types has obvious limitations. For instance, a recent blog post points out that not all clusters are 'real' [121], i.e. can arise due to artifacts from modularity optimization instead of true biological cell types. Perhaps most importantly, the cell type concept itself is not clearly defined, and as introduced above, other disciplines use criteria other than the molecular states. Grabski *et al.* [2, preprint] show that removing or adding cells at random to a data set results in less or more clusters, which makes a direct correspondence paradoxical since the number of cell types is of course unchanged. Furthermore, Seurat v4 paper shows that a single modality is not enough to separate some highly related cell types: Multi-omic protocols that combine scRNAseq with surface proteome (e.g. CITEseq / TotalSeq protocol) or ATACseq (10x Multiome protocol) were necessary to resolve finer cell types with highly correlated transcriptomes (T cell subsets, dendric cells and other myeloid cells) [120]. For single-omic studies (using conventional scRNAseq protocols), this means that the given data does not necessarily offer fundamental cell groupings that clustering can identify. Instead, cell types are simplifications researchers use to understand complex biological systems, and as such they are discussed rather than discovered. This last thought is a key contribution of my work and fundamental to the method I propose here, as will become apparent in section 3. I will introduce below that this topic is heavily debated. For now, I refer the reader to a collection of opinions from established reasearchers of different fields on the topic of how to define cell types [81, comment] and quote the Satija lab: "Deep biological understanding requires more than a taxonomic listing of clusters" [122].

1.7.3 Label transfer from reference data sets

A popular avenue for cell type classification is to use reference data sets to annotate a new 'query' data set. I start by describing methods harnessing references of transcriptomic profiling in bulk. This is followed by various methods using scRNAseq references, which includes a group of methods that use kNN graphs. Other methods apply neural networks for label transfer, and are even able to transfer labels from other sequencing technologies such as scATACseq.

Transfer from bulk reference SingleR [123] and scMatch [124] use bulk RNAseq or microarray data from sorted populations to annotate cell types in query scRNAseq data sets. scMatch finds for each cell the bulk sample with the highest correlation of gene expression or ontology terms, and assigns the corresponding label. SingleR has refined this strategy. Using the ImmGen data base for mouse and Blueprint Epigenomics as well as Encode for human, SingleR is able to access hundreds of bulk samples for dozens of immune cell types. For each individual cell, Spearman correlations are computed with these pure cell type transcriptomes. Immune cells have rich data bases, so SingleR is able to use multiple bulk samples per cell type, making this first step with Spearman coefficients more robust. For each cell, the top ranking reference cell types are used in a second round, again computing Spearman coefficients and this time using only the genes whose expression differs between the high-ranking reference cell types. I note that this approach relies on the existence of rich transcriptome data bases, so does not necessarily apply to cell types other than immune cells. I note that the Azimuth website by the Satija lab is conceptually similar [125]. Azimuth uses Seurat's integration methods (see below) to annotate novel data sets with known atlases. At time of writing this thesis, the supported tissue types were human PBMCs and pancreas, as well as motor cortex from human and mouse.

Transfer from scRNAseq references Similar reference-based approaches using scRNAseq also exist, for example CHETAH [126]. In a first step, CHETAH computes average transcriptomic profiles for each cell type in the reference data. Using hierarchical clustering with Spearman correlation, these averaged reference transcriptomes are arranged into a classification tree. In the second step, the hierarchical classification tree is traversed for each cell, always following the branch with higher Spearman correlation. At a given branching point, the top 200 genes discriminating the two possible paths are used, for an enhanced signal-to-noise ratio. Of note, when the correlations are low for both branches, CHETAH stops at an intermediate node. Thus, CHETAH can label these cells with intermediate labels and for some applications the user can choose to leave them unassigned. This handling of an ‘outgroup’ is in contrast to clustering algorithms, which assign all cells to one group or another.

scPred [127] computes principal component analysis (PCA) on the labeled scRNAseq reference data sets. It selects PCs that contribute significantly to separating the different cell types, and trains a support vector machine

(SVM) with radial kernel on these discriminatory PCs. To annotate a given query data set, scPred applies the same normalization, centering, scaling and PCA transformation as for the training data. It then predicts cell types using the trained SVM. As CHETAH, scPred is able to assign cells as unassigned when the SVM is unable to confidently assign them to any cell type.

Grabski *et al.* [2, preprint] propose a Naive Bayes-like classifier, pre-trained on 216 cell types from PanglaoDB. I note that version 1 of their preprint still mentioned the Naive Bayes algorithm, while the current version does not although it is still inspired by the concept. The core question of their approach is: what is the probability that a cell is of class k given its expression profile and the expression profiles of known cell types? Grabski *et al.* [2] model gene expression as Poisson random variables where the Poisson rates are either drawn from a log-normal distribution with larger mean (on) or one with lower means (off-high from a log-normal, off-low from an exponential distribution. This hierarchy of log-normal / exponential and Poisson is highly familiar to using negative binomial random variables, i.e. a hierarchy of Gamma and Poisson distributions. The goal of this model is to find a 'barcode' of latent variables that for each cell and each gene can take the values on, off-high, off-low. The authors argue that this latent representation of a cell is robust to batch effects and noise. Importantly, Grabski *et al.* [2] then use 3,389,679 cells from PanglaoDB to derive distribution parameters and the latent variables for 218 cell-types. This yielded 6,996 genes that are useful to discriminate cell types. A query data set can then be classified using Bayes rule, linking the class probabilities to the Poisson likelihoods. I note that this approach has not passed peer review at time of writing, meaning that it is still in the state of a preprint.

Graph-based label transfer Graph-based methods are a common approach for label transfer. The goal is to find nearest neighbors across data sets, so that cell type labels can be transferred from reference to query data. To this end, unwanted variation has to be removed while preserving the variation of interest (cell type differences). Examples for unwanted variation are technical batch effects, even for data stemming from different laboratories and sequencing protocols, and biological differences due to age, organ, disease, species or natural heterogeneity between individuals. Below I introduce selected methods but note that many more exist [73, 122, 128–134, scVI, Scanorama, BBKNN, Conos, Seurat, Harmony, scMerge and LIGER, MNNcorrect]. Of note, there is no objective way to discriminate wanted from unwanted variation. Indeed, a recent paper comparing integration methods points out that a trade-off exists between preserving biological heterogeneity and removing batch effects [135]. Different tools differ in how this choice is realized. Once the different samples have been integrated and are mixed well within clusters, cell type labels can be transferred from an annotated reference to query samples.

One example for this graph-based approach is Conos. It is based on the observation that mutual nearest neighbors between two samples are more reliable than conventional nearest neighbors [134]. However, spurious neighbor edges can still arise without a true correspondence [73]. Thus, inter-sample neighbor edges that mark true biological relations between data sets are mixed with spurious edges that represent noise. Conos relies on the assumption that valid inter-sample edges connects communities that are in turn densely connected by many within-sample edges. By mixing inter- and within-sample edges, Conos builds a joint graph. For this, it takes the 'alignment strength' parameter, where the user can force stronger or weaker alignment by changing the number of mutual nearest neighbors that should be used between data. I note that this parameter represents the trade-off mentioned above, namely between removing noise and retaining biological variation. Once the joint graph has been found, this shared representation can be used for clustering with Louvain, Leiden or the hierarchical walktrap algorithm [73]. If one of the data set has cell type labels, Conos propagates them with a diffusion process along the joint graph. I note that Conos not only propagates discrete labels such as cell type annotations, but can also propagate continuous values and thus is able to smooth gene expression. Of note, more samples make Conos more sensitive, meaning that increasingly rare cell types can be detected as more samples are profiled with scRNAseq [73]. In other words, data integration with methods such as Conos are particularly well-suited for exploratory questions, such as the identification of rare cell types.

As for clustering of a single sample (see above), Seurat is a popular tool for label transfer as well. An initial contribution from the Satija lab was to use canonical correlation analysis (CCA) and dynamic time warping to align two samples [37]. Like PCA, CCA is a feature coordinate transformation based on singular value decomposition. It finds components that maximize the correlations between both samples, effectively focusing on the shared biological signals in the data. Once CCA has found a shared embedding, the samples are aligned along this embedding using dynamic time warping. Clusters can then be determined on this batch-corrected feature space. Using this strategy, samples from different sequencing protocols, treatment groups or individuals can be integrated [37]. To refine this strategy further, Seurat version 3 included the mutual nearest neighbor concepts by Haghverdi *et al.* [134]. Specifically, Seurat finds 'cell anchors' between data sets using the MNN concept, and uses these to compute a correction matrix, as proposed by [134]. I note that the earlier CCA approach had the problem that integration results depended on which sample was chosen as starting point for integration - the reference sample. In Seurat version 3, this problem is replaced by computing CCA embeddings between all pairs of samples, and then iteratively merge the most similar data sets (sharing most anchor cells). I note that this elaborate, time-consuming approach has alternatives. For example, Harmony [131] is wrapped by Seurat, meaning it can be used directly in the Seurat workflow, replacing the elaborate CCA approach.

Transfer with deep learning scANVI [136] is the successor of scVI [69]; and VI stands for variational inference, a technique to solve intractable Bayesian posterior computations [136]). Both scVI and scANVI use neural networks to find a low-dimensional embedding of cells, the so called latent variables. Specifically, the model used by both tools considers gene expression as ZINB random variables with covariates for the sample (patient, mouse or any batch unit), the sequencing depth and biological variation. The latter is captured as a low-dimensional vector (10 dimensions in the scVI paper [69]) of Gaussians, representing the latent variables used for cell type identification. During training, the neural networks used by scVI and scANVI find a non-linear mapping between the latent variables and the ZINB parameters (mean, dispersion and drop-out probability). This is achieved with the autoencoder principle, i.e. the network is challenged with the task of restoring the original data while being forced to reducing it to ten dimensions in its bottleneck layer (see also DCA introduced in section 1.6). I note that the scVI paper never mentions the term 'autoencoder' but still employs this principle, as also noted by a recent scVI adaption by Svensson *et al.* [137]. scVI and scANVI are able to integrate multiple data sets at once. If one of them is annotated with cell type labels, these can be transferred with either methods. scVI takes a simple approach, where the majority vote amongst the labeled kNN decides the cell type label of unannotated cells. scANVI takes a more rigorous approach, using a Bayesian semi-supervised approach, c.f. [136, also Appendix Note D therein].

Another deep learning method to label transfer is scGCN [84]. Like Seurat and Conos, scGCN starts by finding mutual nearest neighbors on CCA components to build a neighborhood graph with intra- and inter-sample edges. A graph convolutional network (GCN) then projects both data sets into the same latent space, using three convolutional layers. Next to this hybrid graph embedding, scGCN can integrate individuals, data sets from different single-cell omics technologies and species.

scArches aims at making label transfer with deep learning portable [138]. Specifically, the trained neural network is applied to novel data sets directly, so raw data do not have to be shared. The users can update the reference network again by providing their own data, and share the thus improved weights further, while keeping their unpublished data sets private.

Reference to my work I consider label transfer from annotated reference data sets a powerful approach to exploration. Indeed, it has been noted that as more scRNAseq data sets become available, label transfer methods will make exploration more and more powerful [84]. In my work, I am interested in inference, for which cell type annotation has different requirements, as I point out in the course of this work. Briefly, integrating data sets comes with the trade-off between removing batch effects more aggressively and keeping more nuanced biological signals. Also, the core concept still is to use clustering results as cell type labels, which has the same limitations pointed out in section 1.7.2. I will thoroughly discuss this in context with my own work in chapters 3 and 4.

1.7.4 Labels from known marker genes

Cell-ID, CellAssign, SCINA, Garnett and the Naive Bayes-like approach by Grabksi *et al.* focus on the expression of cell type-specific genes [7]. In this work, I refer to cell-type specific genes as marker genes or cell type markers. These genes are usually overexpressed by the corresponding cells (positive markers), but sometimes cell types are specifically marked by the absence of expression (negative markers). I also use the term gene signature here to denote a list of marker genes, as done for example by Cell-ID [140, preprint]. Marker genes or signatures can be obtained from data bases such as MSigDB, KEGG, the Gene Ontology (GO), Reactome, PanglaoDB or CellMarkers. Alternatively, gene lists derived from microarrays or bulk RNAseq of sorted cell populations can be used, or the genes are manually curated by the investigator [141].

CellAssign CellAssign [141] works directly on raw UMI counts. The user provides a binary matrix, where 1s and 0s encode which genes (rows) mark which cell types (columns). It then fits negative binomial distributions to the raw UMI counts of single cells, one for each cell type and marker gene. Elegantly, it only assumes that a cell type overexpresses its marker genes compared to other cell types, but infers the exact amount from the data. It can also accomodate covariates, such as the sample of origin (e.g. patients). CellAssign employs Expectation-Maximization (EM) to find the maximum likelihood solution for cell type assignments and parameters of the negative binomial distributions. Briefly, the M-step computes the distribution parameters (mean and dispersion for each gene) from all cells currently assigned to a cell type. This is followed by the E-step, where the cell type assignments are computed from the fitted negative binomial distributions (i.e. the probabilities for a cell to belong to each class). After randomly initializing cell class assignments, the EM algorithm alternates E-step and M-step until convergence. Due to this random initialization, CellAssign has to be re-run several times after which the solution with the highest marginal likelihood is chosen. I note that using the raw UMI counts requires either highly expressed genes (high detection rate in the cell type it marks) or a long list of genes. Otherwise, most cells will have zero values in all markers as is typical for scRNAseq data, preventing cell type assignment altogether. Consequently,

[7] A few other tools can take marker genes for cell type annotation, such as scID [139]. I do not introduce them here in more detail because they have a different core philosophy. scID, for example, can use a reference scRNAseq data set for label transfer and is introduced in the previous section for this reason.

almost all examples the authors provide in the CellAssign paper use dozens of genes [141, Table S2 therein].

SCINA SCINA [142] shares many concepts with CellAssign. For example, it also fits mixture models to gene expression using the EM algorithm. Furthermore, it only uses the provided marker genes for cell type assignments, so like CellAssign ignores much of the information available in scRNAseq data. In contrast to CellAssign, SCINA supports negative markers by simply inverting their expression values (multiplication with -1). Still, CellAssign is more rigorous in taking the structure of scRNAseq data into account. Firstly, SCINA fits Gaussian mixture models to log-normalized expression. This is in contrast to the existing noise models I introduced in section 1.3, which would use negative binomial models for UMI data and zero-inflated models for non-UMI data. And secondly, SCINA assumes perfect bimodality, i.e. all negative (positive) cell types are assumed to express at the same low (high) level. CellAssign, in contrast, estimates one gene mean for each cell type. I furthermore note that the authors of SCINA call it a semi-supervised approach because because the marker genes represent prior knowledge, which is in contrast to the usual definition of combining some labeled data with much unlabeled data.

scSorter scSorter takes two aspects into account when assigning cells to classes. First, it gives higher class probabilities to cells with higher expression of the respective marker genes. And second, it computes the classes' current average expression profiles and gives higher probability to cells closer to respective centroid. Thus, scSorter adds k-means clustering to the marker-based strategy used by CellAssign and SCINA. Instead of using mixture models like these two methods, scSorter solves an optimization problem which computes for each marker gene two expression means: the background' (μ in [143]), and the expression of the marked cell type ($\mu + \delta$ in [143]). Like SCINA, scSorter assumes a single mean for all cell types that are not marked by the gene of interest ('background' expression). In order to assign cell type classes, scSorter's optimization procedure then also takes each cell's distance to the different class centroids into account. These centroids are computed from all highly-variable genes that are not user-supplied marker genes. Thus, in contrast to CellAssign and SCINA, scSorter makes use of all the information available in scRNAseq data sets. The user can fine-tune the optimization with a single parameter that weights how much both aspects, marker genes and whole transcriptome, contribute during the optimization. Of note, scSorter had to solve the 'outgroup' problem: what happens to cells that do not match any cell type definitions supplied by the user? After the optimization converged, scSorter excludes cells which have a low proportion of the respective marker genes expressed at high levels. Furthermore, it uses the centroid expression profiles to exclude outgroup cells. For this, it computes the z-score for all variable non-marker genes, squares these values and uses the χ^2 distribution to exclude cells that are further away from this centroid than expected by chance. With these two steps, optimization and outgroup detection, scSorter was able to tell apart major cell types in diverse tissues such as brain, lung, pancreas and blood. For this, the authors used between 3 and 12 markers per cell type, with a few exceptions [143, supplementary material]. I note the scSorter paper is somewhat in conflict with existing knowledge on the structure of scRNAseq data. First, the authors attribute the expression noise mostly to biological heterogeneity, downplaying the widely accepted impact of technical noise [143, Fig. 1 therein]. And second, they do not explicitly model any distribution, in contrast to CellAssign (which uses negative binomial on raw UMIs).

Garnett Garnett is designed for the "rapid annotation of cell atlases" [40]. It uses elastic net regression, which is a family of penalized regression methods that also includes ridge and Lasso regression. In contrast to CellAssign, SCINA and scSorter, this is a classical supervised method, requiring training data. Since labeled data are tedious to obtain for scRNAseq data, Garnett uses a heuristic to find representative cells for training. Specifically, Garnett first uses 'term frequency-inverse document frequency' (TF-ID) to transform the expression values of supplied marker. Using TF-ID is to avoid dominance of highly-expressed markers over more specific but lower marker genes. As heuristic defense against non-specific nuisance expression, values below a certain threshold are next set to zero [8]. Then, Garnett sums the transformed values for all markers of a certain class, and cells in or above the 75th percentile are chosen as training cells. Using these training cells, Garnett trains a logistic regression classifier with elastic net. Specifically, regularized regression on genes is used to predict the classes of all labeled cells. The regularization term avoids over-fitting by shrinking the gene coefficients, focusing on predictive genes. Garnett uses the glmnet R package internally and follows a hierarchical approach: first, major cell types are separated (e.g. T, B and myeloid cells), followed by classifying subtypes (e.g. helper and cytotoxic T cells) separately for each of the major classes. For this, the user writes and maintains a marker file with definitions that include class names, marker genes and parent class. This is in contrast to CellAssign, SCINA and scSorter, where the user input can be represented by a simple binary matrix with genes in rows and cell types in columns, instead of the tree-like structure used by Garnett. The hierarchical approach has the advantage that Garnett can fall back to high-level cell type labels (e.g. T cell) when a cell can not be assigned to any daughter class (e.g. helper or cytotoxic T cell). Once trained, a classifier (more specifically: the gene weights for a linear predictor of discrete cell type labels) can be transferred to

[8] The threshold is 25% of the 95th percentile of TF-ID value. The authors explain that highly expressed markers can leak into the expression of other cells during the sequencing protocol, but this strategy of course also corrects true biological 'leaky' expression.

new data sets. With this approach, Garnett performed well on human lung cells after being trained on murine data. Also, Garnett can be applied to ‘gene activity scores’ derived from scATACseq data, but then assigned only about half of the cells correctly. Like scSorter, Garnett has proposed a way to solve the outgroup problem, i.e. how to identify cells that do not fit to any of the supplied cell type definitions. For this, Garnett clusters a random selection of cells with the Louvain algorithm on a SNN graph, and selects the same number of cells from each cluster to represent the outgroup. This is to ensure that the outgroup is not dominated by the most abundant cell type. With their approach, the authors of Garnett report moderate labeling success (mostly between 50 and 80%, depending on the data set), and aim their method at the ‘rapid annotation of cell atlases’ [40, paper title]. The authors observe that Garnett struggles on complex data sets [40, Fig. S9 therein], i.e. whenever cell types are highly related. After my own experience with Garnett, I note that another limitation lies within the heuristic that identifies representative cells for training. Specifically, a marker gene can only be used by one class without creating conflict. To make a concrete example, the gene PDCD1 (aka PD1) can only be used to identify PD1$^+$ cells in a single cell type, but can mark subsets of both cytotoxic and helper T cells [according to my own observations]. Lastly, I point out the difference in philosophy to other tools. The authors of Garnett propose to curate a data base of pre-trained classifiers, while other methods (CellAssign, SCINA, scSorter) implicitly suggest to curate lists of marker genes themselves.

Cell-ID Cell-ID [140, preprint] is strikingly different from other methods in this section. It can operate completely unsupervised, and also do reference-to-query label transfer. Still, it is included here instead of the above sections because it also has a key focus on finding a few marker genes to assign cell type labels. Cell-ID first ranks genes by how strongly they are associated with each cell in the query data set, and tests for each cell the gene enrichment in known gene signatures (e.g. from data bases or reference data sets). If no signatures are supplied, it allows the user to explore the *de novo* gene signatures. Technically speaking, genes are ranked by their association with cells based on a low-dimensional embedding that is shared between cells and genes. This shared embedding is computed by multiple correspondence analysis (MCA). With MCA, genes specific for a cell are embedded into close proximity to that cell. Of note, if a marker gene is not detected in a cell at all, it can still be highly ranked for that cell due to it’s expression in neighboring cells. The extracted Cell-ID signatures are transferable between batches and have value in their own right: they offer excellent cell type definitions. When the user supplies known gene expression signatures, Cell-ID can also assign cell type labels directly. Due to the transferable nature of gene signatures, Cell-ID can also do label transfer from an annotated reference data set. For this, it matches cells from the query data set with the Cell-ID gene signatures in the reference data set (either individual cells, or groups of cells). I note this method has not passed peer review yet.

Summary and limitations of marker-based approaches To sum up this group of methods, marker-based approaches are promising avenues to cell type classification. One advantage of marker-based approaches is that, unlike clustering, they do not assign all cells but instead estimate the uncertainty in class assignments. Also in contrast to clustering and label transfer, it has been noted [144] that marker-based assignments make use of established knowledge from the literature, i.e. cell type markers. This is important to connect novel findings from scRNAseq studies to 'legacy knowledge' [80], i.e. results reported in the literature that were based on flow cytometry, imaging or other marker-based approaches. Perhaps most importantly, cell type definitions are more useful if a small number of markers is used to define cell types, as noted in recent blog posts [121, 145, blog posts]. These so called 'actionable cell types' [145, blog post] are simple and transferable, and therefore enable downstream experiments for validation and deeper biological understanding. While marker-based tools require the researcher to provide lists of marker genes, these can often be obtained readily from the literature or data bases such as PanglaoDB, CellMarkers or pathway data bases (e.g. MSigDB, GO, Reactome, KEGG, Wikipathways) [140, preprint].

Before marker-based methods can be routinely applied in scRNAseq cohort studies, further improvements are necessary. For example, they all use a rather large number of marker genes. CellAssign takes dozens of genes [141, Table S2 therein], and even the authors of scSorter used between 3 and 14 markers per cell type, with a few exceptions [143, supplementary methods]. These are large numbers when considering the technologies commonly used for validation experiments: Flow cytometry and imaging usually are easiest with a handful of markers, and more than a dozen is hardly feasible for many laboratories. For SCINA and scSorter, I furthermore note that they currently have no way of including different patients into their models. scRNAseq cohort studies profile dozens of patients already today (e.g. [60, 61]), so it is crucial for marker-based methods to work across many different individuals. In this context, I furthermore note that it is unclear whether the cell type definitions used by current marker-based methods (marker files or binary gene-class matrices) are flexible enough to accomodate the heterogeneity of diseased individuals. For example, cancer cells are known to down-regulate prominent markers, and these can be different between patients or even between clones from the same patient (for example antibody light chains in [62]). Finally, marker-based methods have so far only been shown to work on discrete, well-separated cell types. For example, they are often tested by classifying the major cell types in lung, pancreas and blood [40, 143], such as T, B, myeloid, endothelial and glia cells and neurons. For all marker-based methods tested, a recent report found they all struggle on complex tissues, i.e. when cell types have correlated transcriptomes [40, 89, shown for SCINA, Garnett and DigitalCellsorter]. I note neither the original method publications, nor this last benchmarking paper ventured into diseases such as cancer, where highly related T cell subsets are present - thus, it is likely their performance on these would be rather poor. Taken together, scRNAseq studies that have inference, not exploration, as their main goal – such as cohorts comparing two groups – call for novel marker-based methods that overcome above limitations.

1.8 Aims and relevance of this thesis

Single-cell RNA sequencing (scRNAseq) provides transcriptomic information for thousands of individual cells from a given sample. It has been applied to explore gene expression variability [146] and differentiation processes [39], to discover new cell types [57–59], and to study heterogeneous subpopulations in large atlas projects [8, 52–56]. A more recent development are cohort studies [60–64, 147], in which scRNAseq data from multiple patients are compared between different disease conditions (healthy controls, different treatments, indolent or aggressive tumor subtypes, etc.). In contrast to purely exploratory scRNAseq studies, these multi-sample multi-group multi-subpopulation comparisons aim at inference, i.e. statistical hypothesis tests. For example, differential gene expression testing between treatment groups is a typical inference goal, and lists of differentially expressed gene a typical inference result [65]. To allow biological interpretation and contextualization of such results, the cell type classification used for inference has special requirements: First, it should connect well to legacy knowledge generated from marker-based approaches such as imaging and cell sorting [80]. Second, it should be amenable to validation experiments [145, blog post]. And third, the classification method should be able to separate fine cell type subsets in spite of their highly correlated transcriptomes. No existing method currently meets all three of these prerequisites. Briefly, classification methods either require large numbers of marker genes (CellAssign [141], SCINA [142]) or are unable by design to incorporate prior knowledge (graph-based clustering with Seurat [37] or Scanpy [38]).

In this thesis, I set out to address these issues. I develop a novel marker-based classification method for scRNAseq data, called Pooled Count Poisson Classification (PCPC). To apply it, a researcher starts with marker-based cell type definitions, such as CD3E$^+$CD4$^+$FOXP3$^+$ for regulatory T cells, and picks expression thresholds to extract cells of the corresponding cell type. By sharing information across related cells, Pooled Count Poisson Classification (PCPC) can use even weakly expressed marker genes to identify cells of the corresponding cell type. With a single marker gene per cell type, PCPC is able to separate major lineages of blood cells (such as T and B lymphocytes). With four markers or less per cell type, PCPC also resolves highly complex tissues such as a tumor microenvironment, including for example fine subsets of regulatory T cells. Thus, PCPC provides simple, clear and falsifiable cell type definitions that are ideal to obtain and communicate inference results from multi-sample multi-condition multi-subpopulations studies. To illustrate this use case, I apply PCPC to a recently published cohort of lymphoma patients profiled with scRNAseq. With PCPC, I classify cytotoxic T cells as CD3E$^+$CD8B$^+$ cells and find genes for which the expression depends on the aggressiveness of the harboring tumor entity. The goal of PCPC is that results from such an analysis can be summarized in concise, testable statements, in this case: CD3E$^+$CD8B$^+$ cells from aggressive tumors have higher expression of the gene LGALS1 than those from indolent tumors. For these reasons, PCPC has the potential to prove useful in communicating inference results from scRNAseq studies, complementing exploratory approaches (e.g. clustering and data integration). At the same time, PCPC provides a language to discuss and debate cell types themselves in health and disease.

Chapter 2

Methods and Data sets

All analysis in this thesis was done in R version 3.6.3 or 4.0.0. To ensure reproducibility upon re-executing R scripts, I used base R's set.seed function directly before all non-deterministic computations, in particular: Truncated principal component analysis with irlba, neighbor search with RcppAnnoy (version 0.0.18) and UMAP with uwot (version 0.1.9). The latter requires also to set the number of computer processors n_sgd_threads to 1 in order to ensure reproducible UMAP embeddings. I describe all methods thoroughly here, but for convenience the R scripts processing MALT, CBMC and lymphoma data sets are available to the Anders group in my dataMisc R package (data miscellaneous) on the university's scientific data storage (SDS), and are available upon request to everyone else (felixfrauhammer@gmail.com). I note in particular the following scripts of the dataMisc package: malt_preprocess.R, malt_preprocess_singlets.R, cite_preprocess.R and lymphoma_preprocess.R, currently stored in the private repository `https://github.com/FelixTheStudent/dataMisc` and on the SDS. I have also written the scUtils package that is available on CRAN `https://cran.r-project.org/package=scUtils`, which can be used to generate feature plots and compute col/rowsums of sparse matrices. I note that the SDS drive also has data packages for the multiple sclerosis data in [60] (dataMS) and the autism spectrum disorder cohort [61] (dataASD). As the dataMisc package, these two contain the raw UMI counts, the nearest neighbors, UMAP embeddings and clusterings for the data sets, amongst other things.

2.1 scRNAseq analysis workflow

The following workflow was applied to CBMC and MALT data (see next sections for data set details).

1. **Filtering genes.** Genes were used for downstream analysis (namely

PCA) if they were expressed in more than 10 cells (or 0.1% of all cells, whichever number is larger) and if their variance-mean ratio (VMR) was more than 1.5-fold higher than the Poisson expectation $\Xi = \frac{1}{N}\sum_j \frac{1}{s_j}$, where s_j is a cells total UMI count. Deriving Ξ is detailed elsewhere.

2. **Normalization and transformation.** Raw UMI counts were divided by the cell library size s_j (total UMI), followed by square root transformation for variance stabilization. Feature plots (i.e. UMAP embeddings colored by gene expression) also show the normalized and transformed values.

3. **Principal component analysis (PCA).** For PCA, normalized and transformed genes were scaled and the first 60 principal components computed with the prcomp_irlba function from the irlba package (version 2.3.3). Briefly, the rationale behind retaining only the first few components is that this denoises the data [33, preprint] & [46]. Biologically relevant variation is expected to be correlated and therefore to be captured by lower components, while noise is random and expected to be represented by higher components. Also, truncated PCA computes faster (see irlba package documentation). Thus, I chose 60 components [1] in the MALT, CBMC and lymphoma data sets.

4. **k-nearest neighbors (kNN).** 50 kNN were approximated with RcppAnnoy (version 0.0.18). Euclidean distances computed on the first 60 components were used.

5. **UMAP embeddings** were computed from the 50 kNN with the uwot package (version 0.1.9), by setting parameter X to NULL and providing the kNN IDs and distances as list to the nn_method parameter. Furthermore, I set spread=20 and min_dist=1 as this results in less dense UMAP embeddings, which improves feature plots and other visualizations [2]. The same values of spread and min_dist were used for CBMC, MALT and lymphoma data sets.

6. **Louvain clusters** were computed with igraph (version 1.2.6). For this, an undirected graph was constructed manually in R as symmetric binary matrix (known as adjacency matrix) and passed to the gr-

[1] I note that inspecting a scree plot (plotting component variance over component rank) is widely used instead, which suggested fewer PCs (e.g. 12 for CBMC data). However, I have anecdotal evidence that choosing too few components performs poorly (CBMC T cell subsets only separate with 18 or more, not with 12 components), while including more than necessary is not harmful (UMAP embeddings with 30 and with 60 PCs look virtually identical as judged by eye). My rule of thumb is thus to exclude hundreds or thousands of the high-noise components, while the exact number appears to have little impact for performance. I speculate the reason is that lower components have higher variance and thus contribute more to Euclidean distances used for the k nearest neighbors (kNN) search.

[2] Informally, I note this creates embeddings that resemble those computed with tSNE when run with default parameters.

aph_from_adjacency_matrix function. I note this implementation of Louvain has no resolution parameter (see section 1.7.2).

2.2 Feature plots

In this thesis, I use the term 'feature plot' for two-dimensional embeddings (from UMAP, tSNE, etc.) that are colored by the expression of one gene per plot. I note this usage of 'feature plot' was coined by Seurat and its developers [42]. Clear feature plots are crucial for Pooled Count Poisson Classification (PCPC), and in the following I describe how ggplot2 (version 3.3.2) was used to create them. Of note, I have implemented them in the scUtils package available through CRAN (version 0.1.0) as well. Feature plots in this thesis use ggplot2's coord_fixed function for a fixed axes aspect ratio. To avoid overplotting, I compute less dense UMAP embeddings (using spread and min_dist as described in section 2.1) and choose a small point size in plots (between 0.3 and 0.5, depending on the data set). The raw UMI counts were divided by the total UMI for each cell (or neighborhood, for smoothed expression) and zeros were replaced by the one tenth of the smallest non-zero value. For coloring, scale_color_viridis from viridis (version 0.5.1) and viridisLite (version 0.3.0) was used with log2 transformation and the closed breaks implemented in scUtils. Briefly, closed breaks guarantee the minimum and maximum values are labeled in the color legend, and that labels are human readable.

For the lymphoma cohort feature plots, the following extra steps were taken: All points were plotted with transparency (alpha=0.1), except for the cytotoxic T cells (CD3E$^+$CD8B$^+$, alpha=1.0). To make overexpression clearer, non-zero expression was plotted on top of zero expression.

2.3 Doublet removal

For the results in section 3.3.3, doublets were removed as described in the following. After doublet removal, the same analysis workflow as above (section 2.1) was applied to the remaining 6629 cells in order to find kNN, Louvain clusters and UMAP embedding. Doublet removal steps: 1200 synthetic *in silico* doublets were simulated, each by pooling the UMI counts of two cells selected at random from the MALT data. Raw UMI counts of doublets and MALT cells were concatenated and 50 kNN were computed with the above workflow (section 2.1). The percentage of doublets amongst the 50 kNN was computed for synthetic doublets and measured cells, and visualized as histogram. Based on this, I excluded 1783 cells with more than 5 synthetic doublets amongst its 50 kNN (>10% of neighborhood, blue in Fig-

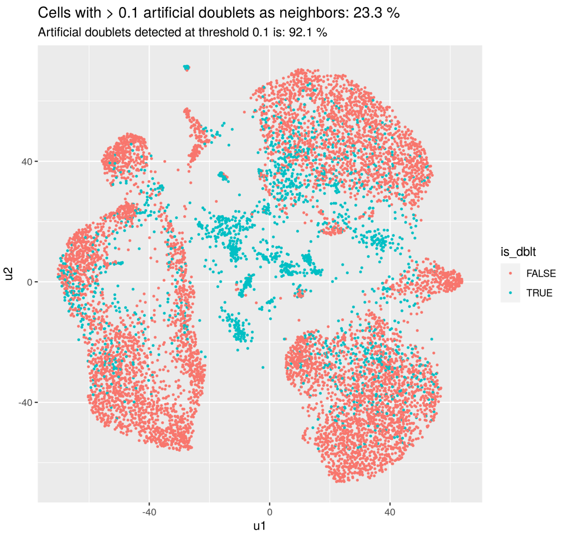


Figure 2.1: Potential doublets removed from MALT data. is_dblt=TRUE ('is doublet') shows cells for which the proportion of simulated doublets amongst 50 kNN is more than 0.1 (10%).

ure 2.1), leaving 6629 cells for further analysis (red cells). To sanity-check this approach, I confirmed that most synthetic doublets would also have been detected with this threshold (1110 out of 1200, or 92.1%), see also Figure 2.1. Furthermore, the excluded cells were mostly located centrally in the UMAP embedding within groups of cells that were double-positive for lineage markers (CD3E, MS4A1 and others, not shown), and otherwise were scattered randomly amongst all cell types.

2.4 CBMC data

The CBMC data generated by Stoeckius *et al.* [113] were downloaded from path/GSE100866_CBMC_8K_13AB_10X-RNA_umi.csv.gz (transcriptome) and path/GSE100866_CBMC_8K_13AB_10X-ADT_umi.csv.gz (proteome), where path is `ftp://ftp.ncbi.nlm.nih.gov/geo/series/GSE100nnn/GSE100866/suppl`. As internal control for the CITEseq protocol, Stoeckius *et al.* included murine cells, and I excluded 612 cells with more than 10% of murine UMI counts from both messenger RNA (mRNA) and protein data.

RNA processing The 50 kNN of each cell, Louvain clusters and a UMAP embedding were computed with the workflow introduced above (section 2.1). I note the CBMC data were generated by somewhat overloading the 10x machine with cells [113], resulting in many doublets and multiplets [3]. The

[3] In this thesis, I define multiplets as droplet barcodes ('cells') that express markers from more than two cell type lineages. I note that capturing many multiplets from soluble cells by random chance is unlikely. Private communications with researchers from the field suggest that they might instead arise due to antibody aggregates 'glueing' multiple cells together, and would thus be specific to the surface proteome protocols (CITEseq, TotalSeq, REAPseq).

way PCPC reacts to the presence of many doublets and multiplets is of interest, which is why I did not attempt to remove them when generating the results in section 3.2.

Protein processing I manually annotated cell types based on the surface proteome and denote the resulting labels as 'protein types' throughout this thesis. The goal is to create an independent 'ground truth', so that methods based on the mRNA only can be tested without circularity. To this end, I aimed at labeling protein types with high confidence (avoiding false-positives at the cost of not labeling all cells). This resulted in 5128 out of 8005 cells with a protein type, leaving only a minority of cells undefined. I observed no overlap between these protein types, as expected from the 'manual gating' strategy I now describe in the following. I normalized surface protein UMI counts with the centered log ratio transformation [4] with the following clr function, adapted from Seurat's source code:

```
clr <- function(x) {log1p( (x) /
(exp(sum(log1p((x)[x > 0]), na.rm = TRUE)/length(x + 1)))
```

Here, x is an R vector with the raw UMIs of a given CITEseq / TotalSeq antibody (each vector element is from one cell). Next, I chose thresholds to find clearly positive (e.g. CD3$^+$) and cleary negative (e.g. CD3$^-$) cells for the following major lineage markers available in the surface proteome (ADT data): CD3 for T, CD19 for B, CD56 for natural killer, CD34 for haematopoietic stem cell, CD14 and CD11c for monocytes [5]. I note the conservative thresholding left some cells in between as undecided, to ensure the purity of protein types. For erythrocytes and platelets / megakaryocytes, there are no good protein markers in the CBMC data. To mark these two cell types and doublets/multiplets containing them, I used HBB, HBG2, HBA1 (erythrocytes) and GP9, PF4 and PPBP (platelets / megakaryocytes) - requiring all three markers for lineage-positive cells, and the absence of all three markers for lineage-negative cells. Having identified positive and negative cells for all lineages, I then selected each cell type as negative for all lineage marker except for their own (two markers for monocytes, three for erythrocytes/platelets, and one marker for all others). In order to discriminate T cells from monocytes [6], both of which can be positive for CD14 and CD11c, I required T cells to be negative for both and on top positive for

[4] The CLR transformation is used for compositional data. Surface proteine UMIs are typically not normalized to total protein UMIs, because the resulting fractions can depend heavily on the cell type. This is because only up to a few dozen features are used, in contrast to the genome-wide scRNAseq data.

[5] I note that CD11c is expressed on other myeloid cells (natural killer, dendritic and macrophage cells) and some T and B cells, next to monocytes. These were excluded, see below.

[6] In flow cytometry, light scatters (forward and side scatter) is used for this, because monocytes have more 'granularity' and scatter more light.

either CD4 or CD8. At the same time, this subdivided T cells into cytotoxic ($CD8^+$) and helper ($CD4^+$) T cells. Of note, I left $CD11c^+$ natural killer cells unassigned in spite of reports of $CD11c^+$ subsets, because they systematically showed larger colSums and were $CD14^+$ (not shown). To mark doublets and multiplets, I required two and more than two lineage markers to be positive, respectively. Of note, I only marked doublets involving T cells ($CD3^+$ plus one other lineage marker, or potential cytotoxic-helper doublets: $CD3^+$ $CD4^+CD8^+$), as these were relevant to the results I discuss in this thesis (see section 3.2).

2.5 MALT data

The filtered count matrix from CellRanger output was downloaded from https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/malt_10k_protein_v3. UMAP embedding, kNN and Louvain clusters were computed as described in section 2.1, doublets were removed as described in section 2.3. The MALT data is generated by 10x Genomics and has the following specifications (selected information taken from [148, website]):

- Cells from a dissociated Extranodal Marginal Zone B-Cell Tumor (MALT: Mucosa-Associated Lymphoid Tissue) stained with TotalSeq-B antibodies.
- Cell Ranger 3.0.0, run with –expect-cells=10000
- Single Cell Gene Expression Dataset using v3 chemistry and using a pre-release set of TotalSeq-B antibodies (https://www.biolegend.com/totalseq).
- 8,412 cells detected
- Sequenced on Illumina NovaSeq with approximately 32,000 reads per cell
- 28bp read1 (16bp Chromium barcode and 12bp UMI), 91bp read2 (transcript), and 8bp I7 sample barcode

I note that the TotalSeq antibodies are a commercialization of the CITEseq protocol by Stoeckius *et al.*.

2.6 Lymphoma cohort

The scRNAseq samples from 12 patients with follicular lymphoma ('lymphoma cohort') was recently published by Roider *et al.*, in a collaboration between the Dietrich lab, me and my colleagues. Specifically, I processed the raw sequencing files with CellRanger as described in [62], helped to explore the data and assisted the leading author with my computational expertise. For this thesis, I use the lymphoma cohort to demonstrate how PCPC can be used in multi-patient multi-condition comparisons. The analysis described in the following is my own and independent from the published manuscript [62].

Sample processing and cell type classification All lymphoma samples (patients) were processed separately with the workflow described in section 2.1 (computing 50 kNN on 60 components, and running UMAP with spread = 20 and min_dist = 1, as for the MALT and CBMC data sets). Cytotoxic T cells were classified with PCPC as CD3E$^+$CD8B$^+$ in all samples separately, except for patient DLBCL1 for which no T cells were detected. Also, in patient FL3, cells were additionally required to be negative for CD4 expression (as indicated in Figure 3.12). Marker genes were selected after thorough exploration. In particular, CD3E was in agreement with CD3D and CD3G and mutually exclusive with MS4A1 and other B cell markers. Similarly, CD8B was co-expressed with CD8A and formed a distinct subpopulation in all samples, with no or hardly any CD4 detected in all samples except patient FL3.

Differential gene expression (aggressive *vs.* indolent) Raw UMI counts of cytotoxic T cells were pooled (summed up) for each patient separately, representing so called 'pseudobulk' samples. DESeq2 (version 1.28.1) with default parameters was used to test these pseudobulks for differential expression between four aggressive (patients DLBCL2/3 and tFL1/2, excluding DLBCL1 due to lacking T cells) and four indolent (patients FL1/2/3/4) tumors. Of note, the 12 samples were aligned with two different versions of CellRanger, and for this reason some genes are only present in half of the cohort samples. For differential expression testing only, these missing genes were excluded. I note, however, that those remaining for differential expression testing contributed at least 91.8% of the totalUMI in all cells, and 96% or higher in the majority of all cells in all samples.

2.7 Cluster markers and ROC curves

For thorough exploration, two methods were used and the results inspected manually for finding cluster markers. First, the functions roc and auc from the pROC R package (version 1.16.2) were used. These were also used to compute receiver operating characteristics (ROC) curves in Figure 3.2. And second, Seurat's FindMarkers function was used with default parameters (Wilcoxon test).

2.8 Tissue complexity

For the Pearson correlation heatmaps illustrating tissue complexities, the normalized expression values (k/s) were averaged across all cells from the same cluster. Averaging was done for all measured genes (33538 for MALT data, 20400 for CBMC data). Pearson correlation was computed on the square roots of the cluster averages, to stabilize the variance across different gene expression strengths. The heatmaps present correlations starting from 0.7 and higher, as the lowest value was 0.72. To summarize tissue complexity into a single number, I took the same approach described by Abdeel *et al.* [89]. Specifically, for each cluster average C_i, the highest correlation to other clusters was selected, and these maximal correlations were averaged:

$$\text{Complexity} = \underset{j}{\text{mean}}(\max_{\forall i, i \neq j} \underset{\forall i,j}{\text{corr}}(C_i, C_j))$$

This formula is adapted from [89].

2.9 Seurat

Seurat version 3.2.2 was used in this thesis. Clusters in Figure 3.8 were computed from raw UMI counts by sequentially applying the following Seurat functions (non-default parameter values are indicated in paranthesis). For scTransform normalization: CreateSeuratObject, SCTransform, RunPCA, FinNeighbors (dims=1:30), FindClusters. The same was applied again after removing potential doublets as detailed in section 2.3. For log-normalization: CreateSeuratObject, NormalizeData, FindVariableFeatures, ScaleData, RunPCA, FindNeighbors (dims=1:30), Find Clusters. Clusters were visualized on the embedding computed with the workflow described in section 2.1, to enable comparison with other figures, and the removed doublets were visualized in grey.

2.10 Garnett

Garnett (version 0.1.17) was run on the MALT data with the marker files supplied in supplement A. For this, a 'CellDataSet' was created with monocle (version 2.16.0) and size factors computed with its estimateSizeFactors function. Marker files were checked with Garnett's check_markers function, revealing considerable marker overlap (not shown). For example, Garnett was concerned about CCR7 being expressed not only by CCR7$^+$ cytotoxic T cells, but also CCR7$^+$ helper T cells and many B cells. It seems that even though Garnett is hierarchical, marker genes are assumed to be not detected at all in other cell types. Then, the train_cell_classifier was applied with num_unknown=50, as recommended for smaller data sets [7] by the tutorial at https://cole-trapnell-lab.github.io/garnett/docs/ [accessed in December 2020 and March 2021]. In order to avoid mapping from symbols to ENSEMBL-IDs and back, I set db="none" and cds_gene_id_type as well as marker_file_gene_id_type to "SYMBOL".

[7] I note that Garnett is designed for large atlas data sets and so the default value for num_unknown is 500, which is why the value 50 as recommended by the Garnett tutorial seems reasonable for the MALT data's 8412 cells. For this reason, I discuss the results from num_unknown=50 in the results (section 3.3.5). To be rigorous, I re-ran Garnett with num_unknown=500 for the fine cell type hierarchy. While this labeled a high percentage of cells (93.8%, not shown), cell type labels were widely incorrect. For example, CD19+ cells were randomly scattered across the UMAP embedding, labeling as many T cells as actual CD19+$^+$ cells.

Chapter 3

Results

3.1 Motivation and structure

The main contribution of this thesis is a novel cell type classification approach, which we refer to as Pooled Count Poisson Classification (PCPC). I start from cell type definitions based on very few marker genes, for which cells are either positive or negative. For example, regulatory T cells ($\mathrm{T_{regs}}$) can be defined as $\mathrm{CD3E^{+}CD4^{+}CD8B^{-}FOXP3^{+}}$. These marker genes are often well established in the literature (as in the $\mathrm{T_{reg}}$ example), are compiled in data bases, or can be found with exploratory data analysis (e.g. graph-based clusters and derived markers, as introduced in section 1.7.2). Given these definitions and a scRNAseq data set, the computational question that is answered here then is how to find positive and negative cells for each marker (e.g. $\mathrm{CD3E^{+}}$ and $\mathrm{CD3E^{-}}$), so that cells can be labeled according to the given cell type definitions.

Inference, exploration and classification I argue that cell type classification should have two equally important goals, for which we need specialized algorithms: Completely data-driven exploration on the one hand, for the unsupervised search of 'fundamental' cell types and novel subpopulations. And inference on the other hand, for multi-sample multi-group comparisons using scRNAseq data (introduced in section 1.5). Marker-based ('inference-oriented') cell types are simplified models that can be discussed and debated in context with the literature, in order to find the most appropriate cell type definitions given the inference question. They make inference results compatible with so called legacy knowledge, i.e. findings from non-sequencing studies using for example sorted cell populations or imaging [1]. Also, they make inference results (e.g. lists of differentially expressed genes) testable with these non-sequencing methods, making the scientific progress

[1] I acknowledge Wagner *et al.* [80] for coining the term 'legacy knowledge'.

more effective by providing falsification opportunities. For these reasons, I refer to definitions based on the expression of a few markers as 'inference-oriented' cell types, and sometimes as 'actionable' cell types as proposed by Valentine Svensson [145, blog post]. I observe that most existing classification algorithms (introduced in section 1.7) are ideal for exploration, with the goal of discovering new cell types or answering similar exploratory questions. In particular, unsupervised tools such as clustering and data integration are not hypothesis-driven and ignore prior knowledge such as marker genes. For inference, by contrast to exploration, classification has very different requirements, which I will define in the next two paragraphs to motivate the design of my own classification method.

Classification under high technical noise To take a data set's signal-to-noise ration into account, I propose that it is useful to assign the label CD3E$^+$ for two types of cells: First, those cells that have a high UMI count for CD3E. And second, all cells with transcriptomes which, in the given data set, are indistinguishable from this first group of highly expressing cells. I reason this way, because the technical noise in scRNAseq data prevents us from accurate statements of true expression rate for individual genes in individual cells. Indeed, methods that attempt such statements introduce false signals, such as spurious gene correlations [67]. Instead, I call a cell CD3E$^+$ if the cell's k nearest neighbors (kNN) showed high CD3E expression, and motivate this by assuming that the transcriptomes from the kNN are in effect indistinguishable from that of the cell of interest in the given data set (c.f. section 4.3 for a discussion of this assumption). Then, while no conceivable algorithm could derive a reliable estimate of the cell's number of messenger RNA (mRNA) molecules at time of lysis, it still is reasonable to say that this cell, judged by all information we have on it, was part of a CD3E-expressing subpopulation. This is an important innovation over the prevalent idea that clusters equal cell types (introduced in section 1.7.2). While valuable for exploratory analysis, the idea that clusters correspond to 'fundamental' cell types (as in data-driven, or unbiased) ignores that the signal-to-noise ratio is limited in any given data set. Instead, the goal of Pooled Count Poisson Classification (PCPC) is to provide researchers with a divide-and-conquer strategy: The first step is to define cell types (e.g. CD3E$^+$CD4$^+$FOXP3$^+$ for T$_{\text{regs}}$) using the literature, exploratory analysis and scientific discussion. The second step is to use PCPC to find cells fitting these definitions in the given data. My goal is to provide a method that also gives feedback during this second step, i.e. the investigator sees whether the data quality is high enough to separate the desired subpopulations, or not. Thus, the question is not whether cell type labels are correct or not, but whether they are 'correct enough' to answer a given scientific question.

Simple, transferable and useful cell type labels There is an increasing number of multi-sample multi-group comparisons with scRNAseq cohorts (c.f. section 1.5). With a focus more on inference than on exploration, these studies would benefit from cell type labels that are easily transferable to validation experiments or the literature, so that the resulting inference insights drive biological understanding foward. To make a specific example, a scRNAseq study might find that $\mathrm{T_{regs}}$ upregulate certain cytokines after stimulation with compound C. Such lists of differentially expressed genes are very typical inference results in transcriptomics, and many technological platforms exist to verify such findings. As a general rule, fewer marker genes provide higher transferability, which is especially true when using *in situ* imaging techniques or fluorescence activated cell sorting (FACS) followed by functional assays. For these downstream experiments, it is crucial that cell type definitions used in the scRNAseq analysis are transferable to these technologies, i.e. that the scRNAseq cell types are actionable [145, blog post]. Put differently, whether the scientific argument built by single-cell studies and their follow-up experiments is convincing or not, I argue, will depend on the characteristics of the cell type definitions themselves. I note that it might be the same research group that tries to verify their finding, or it might be other scientists who would like to integrate findings from the most recent literature into their current research effort. For group comparisons with scRNAseq data, I claim that inference results are most potent if the underlying cell type definitions are simple, transferable and useful:

- **Simple cell type definitions** allow researchers to understand the complexity of biology with a mental representation of reduced complexity. They make scientific discussions more precise and debates clearer. The goal is to derive simple rules for a complex world, or put again differently, to come up with a scientific model that is as simple as possible while explaining as many experimental observations as possible. The exact optimum of this trade-off will be and should be subject to rigorous debate and for this it is crucial that the cell type definition is simple and easily communicated. For example, one publication might claim that $\mathrm{CD3E^{+}CD8^{+}LAG3^{+}}$ cells are cytotoxic T cells with an exhaustion phenotype that are unable to destroy cancer cells. T cell exhaustion is a highly debated field, and a colleague might challenge this definition by observing that LAG3 is also expressed by active T cells, and for exhaustion more markers such as PD1 (HUGO symbol PDCD1) or others might be necessary. It is the simplicity of the marker-based cell type definitions that now allows such a debate to home in on the most useful scientific model to explain cytotoxic T cell behaviour in tumors and tumor models. In contrast, using clusters as cell type definitions lacks this simplicity: In the popular graph-based Louvain algorithm [95], clusters are obtained by optimizing the modularity scores on a neighbor graph defined on thousands of genes. Thus, clusters are not only found by a complex algorithm, but are themselves abstract and complex groupings of cells.

- **Transferable cell types** are available to validation experiments. This makes the inference results, but also the cell type definitions themselves, testable and amenable to falsification. As in the $\mathrm{T_{reg}}$ example above, scRNAseq data often times generate new hypotheses. Since a hypothesis can not be proven with the same data set it was generated with, more experiments are required. To use the wording of Valentine Svensson [145, blog post], cell type definitions should be 'actionable', i.e. have a small number of specific marker genes that can be used in validation experiments using common experimental platforms. To validate the $\mathrm{T_{reg}}$ reaction to compound C (see example above), researchers might in a first step test if their the markers used for their $\mathrm{T_{reg}}$ definition (CD3E, CD4, FOXP3) show good correlation between mRNA and protein expression. If so, they could use flow cytometry to correlate these $\mathrm{T_{reg}}$ markers with cytokine release with and without compound C in a larger cohort of patients. If the correspondence between mRNA and protein is questionable, one possibility is to use *in situ* imaging. To name another option, scRNAseq data from additional patients can be queried with the same inference question: Do $\mathrm{CD3E^{+}CD4^{+}FOXP3^{+}}$ cells to upregulate cytokine expression in response to compound C?

- **Useful cell type definitions** are those that allow researchers to find all cells in question in their data set. In sparse scRNAseq data, the expression magnitude of a marker gene can be as important as its specificity, because the goal is to label not only correctly, but also as many cells in the data as possible. Further points one might consider are if the gene product is a cell surface protein, which allows for cell sorting of living cells and avoids complex fixation protocols, and if it is useful to define cell populations in many settings, such as different organs, disease states or even organisms.

To date, no algorithm exists that fulfills all three criteria, as I will demonstrate and discuss in this thesis (see in particular discussion in section 4.1.3). To fill this gap, I propose Pooled Count Poisson Classification (PCPC), a novel inference-oriented cell type classification method.

Structure of this thesis This chapter presents the results of this thesis and is organized into sections 3.2, 3.3, 3.4 and 3.5. I now briefly describe the content of each section as an overview. Note that each section discusses the relevant literature whenever this helps to contextualize the presented results. A focused discussion of crucial aspects, one by one, can be found in the discussion (4).

The classification method I present in this thesis, PCPC, uses count pooling across neighbors and an expression threshold for classification. Section 3.2

introduces the classification approach in detail. To start with a simple use case, this section uses scRNAseq data from cord blood cells (CBMCs), i.e. clearly separated cell types with highly expressed marker genes. Of note, the section clearly states a generative model for the technical noise in scRNAseq data with UMIs, and motivates PCPC classification approach directly from it. Specifically, the section introduces kNN pooling as principled method to reduce technical noise, and discusses the similarities and differences to gene smoothing tools. Most importantly, section 3.2 shows how pooled counts can be used to make a statistically informed decision on whether a cell is positive for a marker gene, negative or ambiguous. I end section 3.2 with a classification demonstration using the CBMC data set, annotating all major cell type lineages with one marker per major lineage.

In section 3.3, I demonstrate how a more complex scRNAseq data set can be divided into finer and finer structures. The data set used to illustrate this is derived from the microenvironment of a B cell tumor in mucosa-associated lymphoid tissue (MALT). I first show how the MALT data set has highly related T cell subsets and thus is more difficult for classification algorithms than the CBMC data set. Also, I demonstrate that relevant markers such as FOXP3 can be expressed weakly, and that PCPC is able to handle even such sparse signals. This section also discusses the idea that cell type definitions are intelligent choices rather than objective truths, at least with the signal-to-noise ratio available in most existing data sets. The thought that cell types in scRNAseq cohort studies should be discussed, rather than discovered, is central to my work, and section 3.3 prepares the more thorough discussion (c.f. section 4.1). In section 3.3, I also compare PCPC to graph-based clustering and the marker-based tool Garnett.

Section3.4 provides practical considerations for using PCPC, again using the MALT data. For example, the strategy to pick marker genes and their expression thresholds in order to achieve good classification results is layed out. Also, the section explores the bias and variance of smoothed gene expression, in order to demonstrate that pooling nearest neighbors is legitimate for cell type classification, and not introducing major artifacts.

Section 3.5 applies PCPC to a lymphoma cohort recently published by us and others, making clear how I intent my method to be used in case-control studies that use scRNAseq. In particular, I ask how cytotoxic T cells found in aggressive lymphoma subtypes differ from those in indolent ones. I use PCPC to classify cytotoxic T cells as CD3E$^+$CD8B$^+$ cells in each patient sample separately, and compare their gene expression between subtypes. Specifically, differential gene expression testing using four patients per subtype, I find CD3E$^+$CD8B$^+$ cells from aggressive lymphomas over-express LGALS1, a gene implicated in T cell inhibition and apoptosis.

Finally, one important note to understand how PCPC would be used in practice. PCPC requires simple, efficient and interactive R code so that classification in dozens of scRNAseq samples is feasible. In particular, a convenient and flexible way to pick marker genes and their expression thresholds is required. To this end, I started developing the *cellpypes* R package at the final stages of writing this thesis. For this reason, it is placed in the outlook (chapter 5), and I encourage readers interested in this aspect to read section 5.2 ahead of time.

3.2 Classification principle (CBMC data)

In this work I propose to use inference-oriented cell type definition. For example, T_{regs} can be defined as $\text{CD3E}^+\text{CD4}^+\text{FOXP3}^+$, and this definition is simple and transferable. I will illustrate in this section that it is also useful, in the sense that we can extract cells of this type from scRNAseq data. The method I have conceived for this purpose is called PCPC, and this section explains the principle of how it functions in five consecutive subsections. Section 3.2.1 defines gene expression strength to avoid diffuse terminology. Section 3.2.2 introduces the noise model for scRNAseq data used by PCPC, section 3.2.3 proposes kNN pooling for gene smoothing, section 3.2.4 uses this principled smoothing for classification. Lastly, section 3.2.6 provides a technical note on the chosen smoothing method.

3.2.1 Definition of gene expression strength

In this work, I define the gene expression strength π as fraction of total mRNA molecules at time of lysis. I note the exact value of π is unknown in scRNAseq experiments, but can be estimated with some precision from the data. I note that this definition does not normalize for transcript length, because current UMI-generating protocols only count molecule ends. This makes it inappropriate to use the popular transcriptomic units RPKM (short for Reads Per Kilobase Million) and TPM (short for Transcripts per kilobase Million). I also refrain from using the term ‘expression rate’, because ‘rate’ implies events per time. For example, mRNA transcription from the genome has a certain rate, measuring the molecules per hour. In contrast, the expression strength π also includes mRNA degradation, and therefore has no direct relationship with the ‘expression rate’. The only context this work uses ‘rate’ is when speaking about the Poisson distribution’s only parameter, which traditionally is referred to as Poisson rate.

For other technologies, such as quantitative real-time polymerase chain reaction, it is common to normalize to selected house-keeping genes. I choose to normalize π to the entirety of all genes instead, because this is most robust to measurement noise and deprived of selection bias. In summary, a gene’s expression strength π is defined as the proportion of total mRNA molecules that is allotted to that particular gene. In this work, I utilize per mille (‰) for better readability.

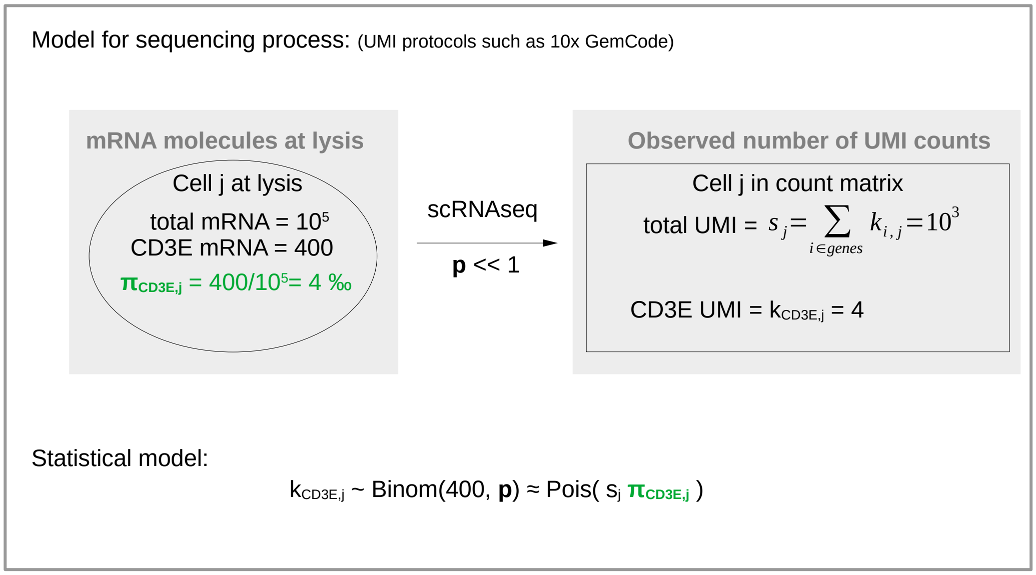

Model for sequencing process: (UMI protocols such as 10x GemCode)
mRNA molecules at lysis
Cell j at lysis
total mRNA = 10^5
CD3E mRNA = 400
$\pi_{CD3E,j}$ = 400/10^5= 4 ‰
scRNAseq
p << 1
Observed number of UMI counts
Cell j in count matrix
total UMI = $s_j = \sum_{i \in genes} k_{i,j} = 10^3$
CD3E UMI = $k_{CD3E,j}$ = 4
Statistical model:
$k_{CD3E,j}$ ~ Binom(400, **p**) ≈ Pois(s_j $\pi_{CD3E,j}$)

Figure 3.1: Statistical model for UMI counts. For a given gene such as CD3E, the scRNAseq assay (including cell capture, library preparation, sequencing and computational preprocessing) converts mRNA molecules to UMI counts with probability p. This conversion is imperfect ($p \ll 1$) and introduces technical noise, which is accounted for by PCPC by modeling the UMI $k_{\text{CD3E},j}$ as a Poisson random variable. The underlying distribution only has a single unknown that has to be inferred, namely the the fraction of mRNA molecules π_{ij}. Symbols: gene i, cell j, UMI counts k_{ij}, total UMI s_j, true unknown expression strength π_{ij}.

3.2.2 Statistical model

There is still some debate on how UMI counts are best modelled, e.g. whether the negative binomial instead of the Poisson distribution should be used or not [1, 32, 44]. Perhaps adding some confusion to the debate, it is often left unclear whether the specified distribution models individual cells, a cell's kNN, clusters or even the entire data set. In this section, I therefore precisely state the generative model for UMI counts as assumed throughout this work.

PCPC assumes that the UMI count of a given gene in an individual cell is a Poisson random variable with a Poisson rate that is unknown. Figure 3.1 is a schematic on the basis of which I now motivate using the Poisson model. At time of lysis, the exact number of mRNA molecules in a given cell j is unknown. To make an example, Figure 3.1 shows a cell with ten thousand total mRNA molecules, 400 of which encode the gene CD3E. The gene expression strength of CD3E is then defined as the fraction of the cell's total mRNA molecules, and in this example is $\pi_{\text{CD3E},j} = \frac{400}{10^5} = 0.004 = 4‰$. For readability, all figures throughout this work provide gene expression strengths π and their estimates as per mille (‰).

During a typical scRNAseq experiments, cell j is lysed and all of its mRNA molecules released into a small reaction volume, e.g. a microwell [8] or a water droplet inside immersion oil [6, 16]. With a certain probability p, mRNA molecules are converted to UMI counts in the final scRNAseq data matrix. As indicated in Figure 3.1, p is between 0 and 1 but typically much smaller than 1 ($p \ll 1$). I note that p summarizes several steps: Before it is recorded as UMI count in a data file, mRNA molecules have to be reverse transcribed, amplified, tagged with sequencing adapters, sequenced with no or very few errors and then be detected and quantified *in silico* within the reference genome by alignment [18, 20] or pseudoalignment [21]. Each of these steps can fail for some of the mRNA molecules, with the result always being that it is not observed as UMI count in the scRNAseq data. Hence, the model summarizes all steps as Bernoulli process with the probability p. Of note, p includes the reverse transcription step, that has been estimated to be as low as 6 - 32%, depending on which version of the 10x genomics products is used [15, 16]. Thus, the rate p is likely dominated by, but not restricted to, limitations in the reverse transcription step of scRNAseq protocols. The right part of Figure 3.1 shows how the resulting UMI counts for cell j might look like, assuming that roughly 1% of mRNA molecules in cell j give rise to observed UMI counts, i.e. $p = 0.01$. Cell j's total UMI counts s_j are two orders of magnitude smaller than the number of mRNA molecules, as expected from a stochastic downsampling with 1% success probability [2]. For CD3E, 4 UMI counts were observed in this example, out of 400 mRNA molecules. Of note, observing 3 or 5 counts instead of 4 would have been almost as likely - the final number is subject to a stochastic process, and the variation between different outcomes of this thought experiment are purely due to technical noise, without any biological component. Strictly speaking, the expected UMI counts for CD3E in cell j, or $k_{\text{CD3E},j}$, follow a binomial distribution with a size of 400 and a probability of $p = 0.01$ (see Figure 3.1). In scRNAseq experiments, both n and p are unknown. Importantly, we can only estimate their product, i.e. n and p are not separately identifiable: Increasing one of them is just as likely as long as the other decreases by the same factor. Therefore, I use the Poisson distribution to model the counts $k_{\text{CD3E},j}$, which is not uncommon in scRNAseq data analysis [1, 32]. The Poisson distribution approximates the binomial distribution for small p and large molecule numbers, and only uses a single parameter called the Poisson rate. Coming back to the example in Figure 3.1, PCPC models UMI counts of gene CD3E in cell j as Poisson random variable with the Poisson

[2] As noted, modern protocols make 10% or even 30% possible (p=0.1 or p=0.3), which would make 10^4 rather than 10^3 total UMIs a more representative example. In practice, total UMIs can easily span this range – in my experience, T cells have thousands of total UMI, while neurons have ten thousands.

rate $s_j \pi_{\text{CD3E},j}$. In summary, using Poisson random variables to model UMI counts arises naturally from above considerations of the sequencing process.

For classification, the crucial question will be to group cells according to their unknown value of π_{ij}. To this end, the next section introduces gene smoothing and how it can mitigate the challenges posed by technical noise. In particular, I chose to smooth with neighbor pooling, and will use the Poisson random variable model to motivate this choice. Later in this section, I will then show how from the same model, a simple decision strategy based on Poisson distribution tails lends itself to our goal of deciding for each cell whether it is positive or not for a certain marker.

3.2.3 kNN pooling smoothly separates cell types

Using the noise model above, I propose kNN pooling in this section, a principled method to smooth gene expression values. I show how kNN pooling separates cell types in a simple example. In the next section, I then show how pooled counts are useful to separate cells using Poisson tails.

Given an inference-oriented cell type definition for regulatory T cells such as:

$$\text{T}_{\text{reg}}\text{: CD3E}^{+}\text{CD4}^{+}\text{FOXP3}^{+}$$

how do we find T_{regs} in a given scRNAseq data set? In flow cytometry, manual gating is prevalently used. This strategy is not directly applicable to scRNAseq data, since mRNA typically is not as abundant as the corresponding protein, which leads to considerable technical noise in scRNAseq data (see previous section). Here, I propose that smoothing gene expression can overcome this obstacle and restore the original signal to the extent that cell types can be discriminated with high accuracy. Algorithms leveraging kNN information dominate the field of scRNAseq data analysis (examples include UMAP [47] and graph-based clustering [95]), and we reasoned that they could also be helpful to solve our classification problem. Here, I propose to use kNN pooling to smooth gene expression and classify cell types in scRNAseq data. Specifically, I start from the noise model introduced by Figure 3.1, and assume that the UMI counts k_{ij} for gene i in cell j are described by a Poisson random variable:

$$\begin{aligned} k_{ij} &\sim \text{Pois}(s_j \pi_{ij}) \\ s_j &= \sum_i k_{ij} \end{aligned} \tag{3.1}$$

Here, s_j denotes the total UMI counts of cell j, and π_{ij} is the unknown expression rate as a fraction of s_j. We denote the set of 50 nearest neighbors as $\mathcal{N}_j$, where 50 is fixed for each cell (discussed in 4.3). We now pool counts and totals across cell j's nearest neighbors $\mathcal{N}_j$:

$$
\begin{aligned}
K_{ij} &= \sum_{j' \in \mathcal{N}_j} k_{ij'} \\
S_j &= \sum_{j' \in \mathcal{N}_j} s_{j'}
\end{aligned}
\tag{3.2}
$$

To make a concrete example with a common T cell marker, $K_{CD3E,j}$ is computed by summing the UMI counts of transcript CD3E across the kNN of cell j (including that cell j itself), where I choose 50 nearest neighbors as 'bandwidth' (more on the bandwidth can be found in section 3.4.3). I next note that division by the overall neighborhood size S_j gives the smoothed expression value $\frac{K_{CD3E,j}}{S_j}$, for simplicity referred to as $\frac{K}{S}$ from here onward (and throughout this thesis, for different genes depending on context). Intuitively, $\frac{K}{S}$ is to the neighborhood $\mathcal{N}_j$ what $\frac{k_j}{s_j}$ (or $\frac{k}{s}$ for simplicity) is to cell j; namely the fraction of total transcripts allotted to the transcript CD3E. Thus, $\frac{K}{S}$ and $\frac{k}{s}$ are both estimators of π_{ij}, but the latter is dominated by technical noise, while $\frac{K}{S}$ uses information from multiple cells. In statistical terms, $\frac{K}{S}$ has less variance, but more bias than $\frac{k}{s}$ (detailed in section 3.4.3).

Figure 3.2 shows the CBMC data set and illustrates that $\frac{K}{S}$ is much better at separating cell types than $\frac{k}{s}$. To use an independent ground truth for the cell types, I use the robust protein signal of this CITEseq data set to define cell types by manual gating (see methods section 2.4). Figure 3.2a shows the resulting cell type labels, and only cell types that are relevant in the following are shown for clarity: Two T lymphocytes subtypes (green and dark green), NK cells (orange) and doublets (red). Figure 3.2b shows the same UMAP embedding, colored by the values of CD3E $\frac{k}{s}$. Evidently, T cells highly express CD3E as expected for a T cell marker, but even in the T cell population (as defined in Figure 3.2a), some cells have had zero counts for CD3E (dark purple points in Figure 3.2b). These zeros are typical for sparse scRNAseq data, and it is not unlikely that virtually all T cells had at least a few CD3E mRNA molecules at time of lysis, but we simply did not observe them as UMI counts. These so called 'drop-outs', as we now know, are to be expected due to Poisson noise [32] (as introduced in section 1.3). Figure 3.2b also shows that CD3E is expressed in doublets and a few NK cells. Doublets are expected to have some CD3E expression in data sets with many T cells, because these doublets are created when a T cell is randomly captured and sequenced together with a second cell of any type. For the NK cells, it is feasible that the observed expression of the T cell marker CD3E is due to biology ('leaky expression') or due to a technical artifact. For example, ambient mRNA can cross-contaminate droplets, and several tools have been proposed to remove such artifacts [149]. As a side-note, the Poisson model described in the previous section is valid in any case, whether CD3E expression in NK cells is of biological or technical nature.

Developing the classification method proposed in this thesis (see next section) was inspired by the observation that smoothed marker gene expression

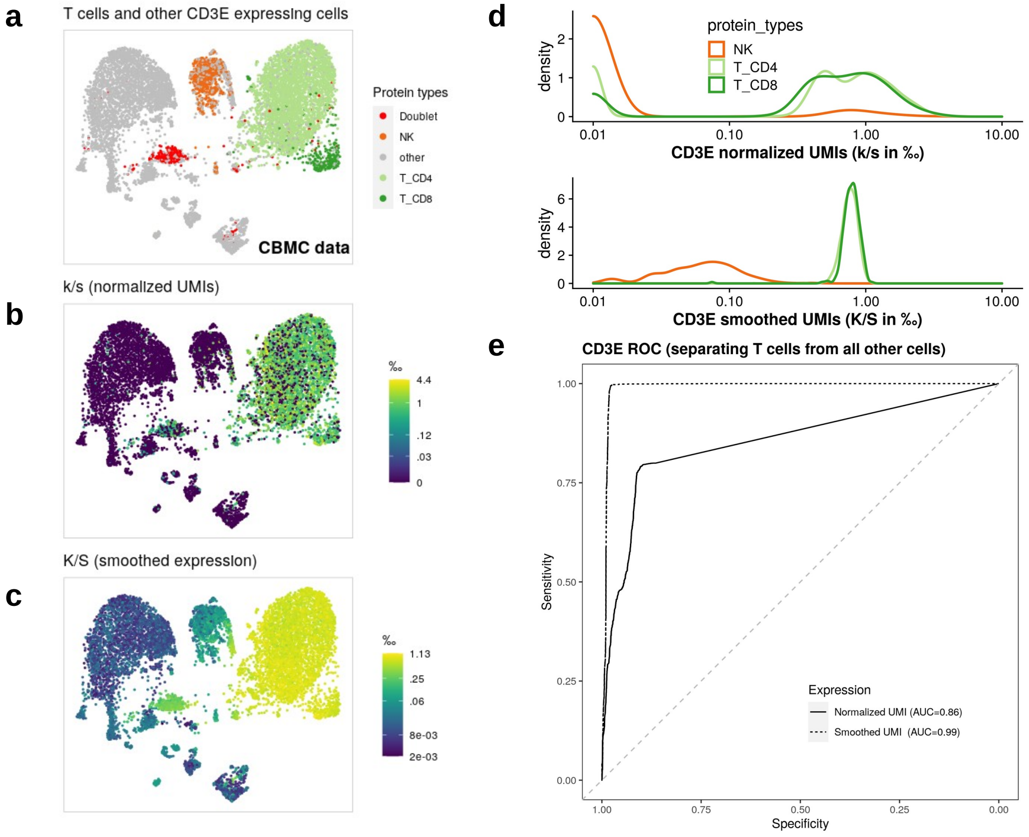


Figure 3.2: Smoothed CD3E expression identifies T cells in the CBMC data. **a, b, c** show UMAP embeddings (c.f. section 2.2). **a** Cell types were manually annotated using surface protein levels (c.f. section 2.4), and of these 'protein types' the following selection is shown: T cells (green), NK cells (orange) and doublets (red). **b, c** CD3E expression before and after smoothing, shown as fraction of the library size in ‰ (per mille). **d** Density plot of CD3E expression before and after smoothing, colored as in a. Most values in the upper panel (k/s) are zero, which appears as continuous (instead of discrete) peak around 0.01 due to the kernel density estimation. **e** Receiver operating characteristic (ROC) curve for classifying T cells with CD3E before and after smoothing. The area under the curve (AUC) is provided in the inset.

is able to separate cell types even in presence of high technical noise. As illustrating example, Figure 3.2c, d and e show how smoothed CD3E expression separates T cells from all other cell types, including NK cells. As evident from Figure 3.2c, the smoothed expression value $\frac{K}{S}$ is higher in T cells than in NK cells and all other cell types, even the doublet cells marked in Figure 3.2a. The separation of T cells from the closely related NK cells is even more obvious from the density plots depicted in Figure 3.2d. Before smoothing (upper panel), the NK cells which express CD3E form a peak that is inseparable from the two T cell subtypes. After smoothing (lower panel), this peak is gone, and all NK cells were estimated to have lower expression than T cells. Likewise, T cells that had zero expression before smoothing formed a peak around zero (upper panel in Figure 3.2d) that is not present after smoothing. Put differently, if we choose a threshold in the density plot Figure 3.2d, a lower threshold will select more T cells as CD3E$^+$, but at the risk of contaminating them with more NK cells. For the unsmoothed value $\frac{k}{s}$, these false-positive cells are more abundant. To quantify this improvement after smoothing, I construct a simple ROC classifier (Receiver operating characteristic) for $\frac{K}{S}$ and $\frac{k}{s}$. Figure 3.2e shows the resulting ROC curves, recording the respective ability to separate T cells from all other cell types (defined by surface protein expression, see methods section 2.4). Moving along the curves, a lower and lower threshold will increase sensitivity, i.e. more T cells are marked as CD3E$^+$ cells. For the unsmoothed value $\frac{k}{s}$, however, this comes at the cost of decreased specificity (note the reversed x-scale of Figure 3.2e), as a lower threshold also incorrectly selects more NK cells. Strikingly, for the smoothed expression $\frac{K}{S}$, this trade-off is much lower: the sensitivity can be increased to 1 without decreasing specificity much. The area under the ROC curve (AUC, see Figure 3.2e) measures the ability to separate cell types with high specificity and sensitivity, and an AUC of 1 indicates perfect binary classification. I observe that $\frac{K}{S}$ achieves an AUC of 0.99, demonstrating the clean separation of T cells from all other cell types. In contrast, the unsmoothed value $\frac{k}{s}$ has an AUC value of 0.86, indicating the incomplete separation of T and NK cells when CD3E is not smoothed.

Thus, kNN pooling of CD3E is perfectly able to identify T cells in the CBMC data, and we will see in later examples that this approach works with weaker markers and less distinct cell types as well (see section 3.3).

Comparing the legends of Figure 3.2s b and c, it is apparent that the variance is reduced after smoothing: Expression values range from 0 to 1.13‰ for $\frac{K}{S}$ (Figure 3.2c), while they range from 0 to 4.4‰ for $\frac{k}{s}$. I interpret this observation, taken together with the results from above, as follows: denoising the expression through kNN pooling allows us to better estimate each cell's actual expression rate $\pi_{CD3E,j}$. For T cells, this expression rate was close to 1‰ (see green peak in density plot Figure 3.2d), while NK cells had values for $\pi_{CD3E,j}$ around 0.1‰. This tenfold difference is what allowed an AUC of 0.99, i.e. virtually perfect binary classification into CD3E$^+$ and CD3E$^-$ cells, once the gene expression was smoothed. I stress that the goal of kNN pooling was exactly this binary classification, i.e. a qualitative decision. Specifically, I do not claim quantitative reliability of the smoothed expression value, but rather propose to use it as a proxy to group cells into CD3E$^+$ and CD3E$^-$ populations. In contrast, there are many imputation methods that have tried to restore a continuous and precise estimate of π_{ij}, for example in order to measure correlations between individual genes more precisely, or to improve clustering results [31, 33, 35, 36, 68, 71, 78]. Andrews *et al.* [67], however, could show that such continuous estimates of π_{ij} are difficult without introducing artifacts, such as false correlations between genes. Thus, it is likely that in scRNAseq data, the transcript success rate p (introduced by the model in Figure 3.1) is not high enough for quantitative statements concerning individual genes. In cases where indeed the precise number, fraction or variability of mRNA molecules from an individual gene is of interest for the scientific question, investigators would ideally use single-molecule imaging techniques rather than scRNAseq, as demonstrated recently [150]. To summarize this thought: kNN pooling can not guarantee that all T cells (or better: CD3E$^+$ cells) actually had CD3E molecules at time of lysis. Instead, with the specific sequencing depth of a given scRNAseq data set, PCPC marks those cells as CD3E$^+$that had a transcriptome associated with CD3E expression. I will revisit these considerations in more detail in the next section, when I introduce the statistical approach to divide cells into positive and negative populations.

3.2.4 Classification with pooled counts

I showed in the above section that modeling UMI counts as Poisson random variables is a principled approach, which is why count pooling can overcome technical noise. To find cells that are positive for a marker gene, I next propose to simply compare their pooled UMI counts to a user-defined threshold. This section introduces my approach, involving the tails of a Poisson distribution around the given threshold. The next section demonstrates how the CBMC data set can be fully annotated with cell type labels this way. Application to more complex tissues is demonstrated in section 3.3, and to a cancer cohort in section 3.5. While automating PCPC is challenging (see sections 3.3.5 and 4.2), interactive code makes it swift and simple (section 5.2).

In this section, I suppress the index i because we consider always the same gene. To find CD3E$^+$ cells, PCPC pools counts across neighbors and compares them to a given threshold value. I choose deliberately to work with pooled counts K_j and the smoothed value $\frac{K_j}{S_j}$, rather than for example averaging $\frac{k_j}{s_j}$ across kNN, because of the useful properties that K_j has. Specifically, as sum of Poisson random variables, it itself is a Poisson random variable. Let K_j denote the UMI counts of a given marker gene, pooled across cell j and its nearest neighbors, which together I denote $\mathcal{N}_j$. Then, I model the pooled counts as a Poisson random variable:

$$K_j \sim \text{Pois}(\sum_{j' \in \mathcal{N}_j} s_{j'} \pi_{j'})$$

where $s_{j'}$ are the kNNs' known totals, and $\pi_{j'}$ are their unknown expression rates. I assume that kNN have virtually indistinguishable transcriptomes (discussed in section 4.3), and thus approximate their expression rates $\pi_{j'}$ as identical. I write this common expression rate as $\pi_{\mathcal{N}_j}$, and will discuss this assumption of equal rates in section 4.4. Then, the pooled counts for cell j can be modeled as:

$$\begin{aligned} K_j &= \text{Pois}(\pi_{\mathcal{N}_j} \sum_{j' \in \mathcal{N}_j} s_{j'}) \\ &= \text{Pois}(\pi_{\mathcal{N}_j} S_j) \end{aligned} \tag{3.3}$$

For a given threshold value τ, PCPC defines positive cells as those where K lies clearly above values we would expect from a Poisson distribution where $\pi_{\mathcal{N}} = \tau$:

$$\mathcal{F}_{\text{Pois}}(K_j, \tau S_j) > 0.99 \tag{3.4}$$

where $\mathcal{F}_{\text{Pois}}$ is the Poisson cumulative distribution function (CDF). Likewise, negative cells are those where the CDF is below 0.01, while cells outside of the two distribution tails remain unassigned. Thus, PCPC constructs a Poisson distribution around τ in order to quantify the uncertainty which UMI counts inherit from the stochastic sequencing process. This uncertainty is modeled as Poisson noise, as introduced in section 3.2.2.

The strategy to find positive cells is illustrated in Figure 3.3. The threshold $\tau = 0.45‰$ is shown as dashed line in Figure 3.3a. The grey points around it represent cells whose smoothed values $\frac{K_{ij}}{S_j}$ were too close to τ, and can be 'drawn' with high probability from a Poisson distribution with Poisson rate τS_j. Clearly above (below) this area of uncertainty lie the positive (negative) cells marked with blue (red) points. Figure 3.3b shows the same cells in the UMAP embedding. Evidently, most T cells in the right third of the embedding[3] were selected as positive (blue dots), and all other cell types as negative. The percentage of unlabelled T cells is very small, demonstrating that CD3E$^+$ is a useful definition to extract T cells from this data set. Interestingly, most unassigned cells (grey dots) are located in areas of 'leaky' CD3E expression (NK cells and doublets, explained more in Figure 3.2). In these grey cells, the smoothed value $\frac{K_{ij}}{S_j}$ was higher than in negative cells (red dots), but not high enough to be counted as positive. The fact that these unassigned cells were NK cells and doublets highlights that it can be useful to leave some cells unassigned.

Finding suitable values for the threshold τ is crucial for this approach. In this example, I found $\tau = 0.45$ by inspecting the resulting CD3E$^+$ cells in the UMAP embedding (blue cells in right panel of Figure 3.3). Specifically, I compared the blue area to the area of CD3E expression (marked by $\frac{k}{s}$, see Figure 3.2b), and chose the threshold that maximizes agreement between the two. I will show a clear example for this thresholding strategy with the MALT data set in Figure 3.6 (section 3.3). In later sections, I discuss difficulties to automate picking thresholds manually (sections 3.3.5 and 4.2), but give an outlook on interactive code to make this task simple and fast (section 5.2). For now, it is enough to say that a manual threshold, once the PCPC user has found it, can split cells into positive and negative populations, with

[3] See also Figure 3.2a for a transcriptome-independent definition of T cells.

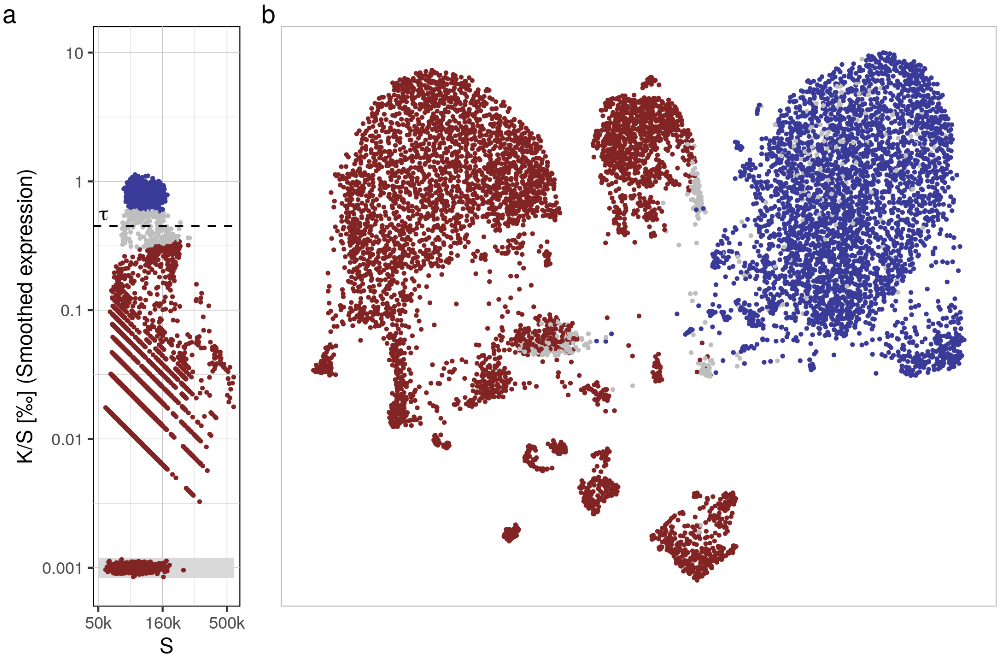


Figure 3.3: Poisson tails identify CD3E$^+$ cells. **a**: Smoothed expression $\frac{K}{S}$ (per mille, ‰) over neighborhood size S (total UMI, 50k=50,000), computed across 50 kNN for each cell. Both axes are logarithmic. For better visualization, the minimal value on the y-axis is 0.001‰, cells with lower expression are visualized as jittered points inside the grey box. The dashed line indicates user-defined a threshold τ. The Poisson distribution function of K given τ and S divides cells into positive (blue, CDF > 0.99), negative (red, CDF< 0.01) and unassigned (grey, all others). **b**: UMAP embedding, cells are colored as in left panel.

unassigned cells in between. This approach is reminiscent of flow cytometry, where it is common practice to set gating thresholds manually, and leave cells in between unassigned. The difference to flow cytometry is that scRNAseq data is not biased for a few genes, but the user can select marker from the entire transcriptome. This makes marker-based cell type definitions more valuable, because if the utility of a given marker is in doubt, any researcher may download the respective data set and use PCPC to propose and test other markers.

To focus on a technical aspect, the width of the grey area in Figure 3.3 decreases when the neighborhood size S_j is larger. Put differently, even though τ is always the same, each cell has a different cut-off value for the smoothed value $\frac{K_{ij}}{S_j}$. Cells with sparser kNN have smaller neighborhood sizes S_j, and my interpretation is that for these cells, less information is available to judge whether the neighborhood expression rate $\pi_{\mathcal{N}}$ was more extreme than the threshold τ or not. PCPC accounts for this, by increasing the area of uncertainty around τ for smaller neighborhood sizes. For this reason, we choose to show the smoothed value $\frac{K_{ij}}{S_j}$ plotted over S_j in Figure 3.3, instead of a histogram. The histogram of the smoothed expression would perhaps be more intuitive to some. It does, however, not display the information contained in neighborhood size S_j and therefore might fail to show separation of positive and negative cells as clearly.

A further technical note is that K_{ij} and $\frac{K_{ij}}{S_j}$ show a different relationship between S_j and the associated uncertainty. As for all Poisson random variables, the variance of K_{ij} increases linearly with the Poisson rate τS_j. Thus, the pooled counts K_{ij} themselves are less certain when more 'information' is added. In contrast, the smoothed value $\frac{K_j}{S_j}$ has less uncertainty for larger S_j and so π_{ij} can be compared with τ with increasing confidence. This increase *versus* decrease with larger S_j becomes obvious when considering the variances of K_{ij} and $\frac{K_{ij}}{S_j}$:

$$\begin{aligned} \mathrm{var}(K_{ij}) &= \tau S_j \\ \mathrm{var}(\frac{K_{ij}}{S_j}) &= \frac{1}{S_j^2}\,\mathrm{var}(K_{ij}) = \frac{\tau}{S_j} \end{aligned} \tag{3.5}$$

Here, I use the variance property that when a random variable K_{ij} is scaled by a factor $(\frac{1}{S_J})$, the variance decreases by that factor squared. As 3.5 shows, the uncertainty for the pooled count increases with larger S_j, while it decreases for the smoothed value $\frac{K_{ij}}{S_j}$, and this explains why the grey area of uncertainty in Figure 3.3a is broader for small S_j.

I end this section with noting some parallels between statistical testing and above strategy. Equation 3.4 and its rationale are reminiscent of hypothesis testing, where the null hypothesis would be that the Poisson rate $\pi_{\mathcal{N}}$ was precisely τ. In contrast to actual hypothesis testing, however, this null hypothesis would be completely unreasonable: the user specifies the threshold τ precisely such that cells are divided into two groups by it, and is not interested in the actual confidence and p-value with which the null hypothesis can be rejected. Instead, PCPC uses the Poisson model to quantifying the uncertainty in UMI measurements, and aims at selecting cells whose transcriptomes are similar to those of highly expressing cells. For this reason, we do not apply multiple testing correction: testing multiple cells is not directly linked to generating more false positives that would need adjustment. Instead, the majority of cells are expected to lie clearly above or below the threshold τ, because by definition the user chooses the threshold so that it separates positive from negative cells as clearly as possible.

3.2.5 Annotating cell types in the CBMC data with PCPC

The above sections focused on separating T cells from other cell types, i.e. only two classes. Before I move on to classification in more complex tissues (section 3.3), this section demonstrates how PCPC can annotate all cell types in the CBMC data - which contain mononuclear blood cells from a healthy donor, i.e. transcriptionally very distinct cell types.

As mention above, I have manually annotated the CBMC data with ‘protein types’, i.e. cell type definitions based on the CITEseq protocol’s surface proteome (see methods in section 2.4). Figure 3.4 shows these protein types, next to the cell types I assigned with PCPC using well established blood cell marker genes. I will elaborate more on how to find suitable thresholds for PCPC below, in sections 3.3.2 and 5.2.

Overall, there is very good correspondence between the plots, demonstrating that our approach can perform well with as few as a single marker gene per cell types (except T cell subsets, which I defined using two marker genes). I observe the largest discrepancy between both annotation methods for classical monocytes (Mono_c in a, $CD14^+$ cells in b). Here, I was very conservative with the protein type annotation, stringently labeling only a few cells. While the general agreement still is good, I note that PCPC’s $CD14^+$ cells include a few multiplets. This is a good opportunity to discuss the selection of marker genes: I chose $CD14^+$ as definition for classical monocytes, but now find through exploration that this includes some unwanted cells, in this case multiplets[4]. Hence, I may find it advisable to instead define classical monocytes as $CD14^+CD3E^-$, for example, potentially getting rid of doublets. An alternative would be of course to first exclude doublets, either with dedicated tools [151–155] or simulations (see method section 2.3).

Another point of interest are the two T cell subtypes (T_CD4 and T_CD8 in Figure 3.4a). Evidently, PCPC does not separate these two cells without errors, although only a few. The authors of the Seurat package note for the same data set that indeed separating the two T cell subsets is also not achieved by Seurat [120]. The reason is that cord blood contains mainly naive cells, which have their differences on other modalities than the transcriptome. In line with this, Seurat version 4 was able to separate them only with multimodal analysis, i.e. when the transcriptome was analyzed together with the the surface proteome [120].

In summary, PCPC annotates cells in the CBMC data set in good overall agreement with the cell types defined on rich surface proteome data, while

[4] Multiplets are defined here as ‘cells’ that express markers from more than two cell type lineages. The CBMC data was generated by somewhat overloading the 10x machine with cells[113], resulting in many doublets and multiplets

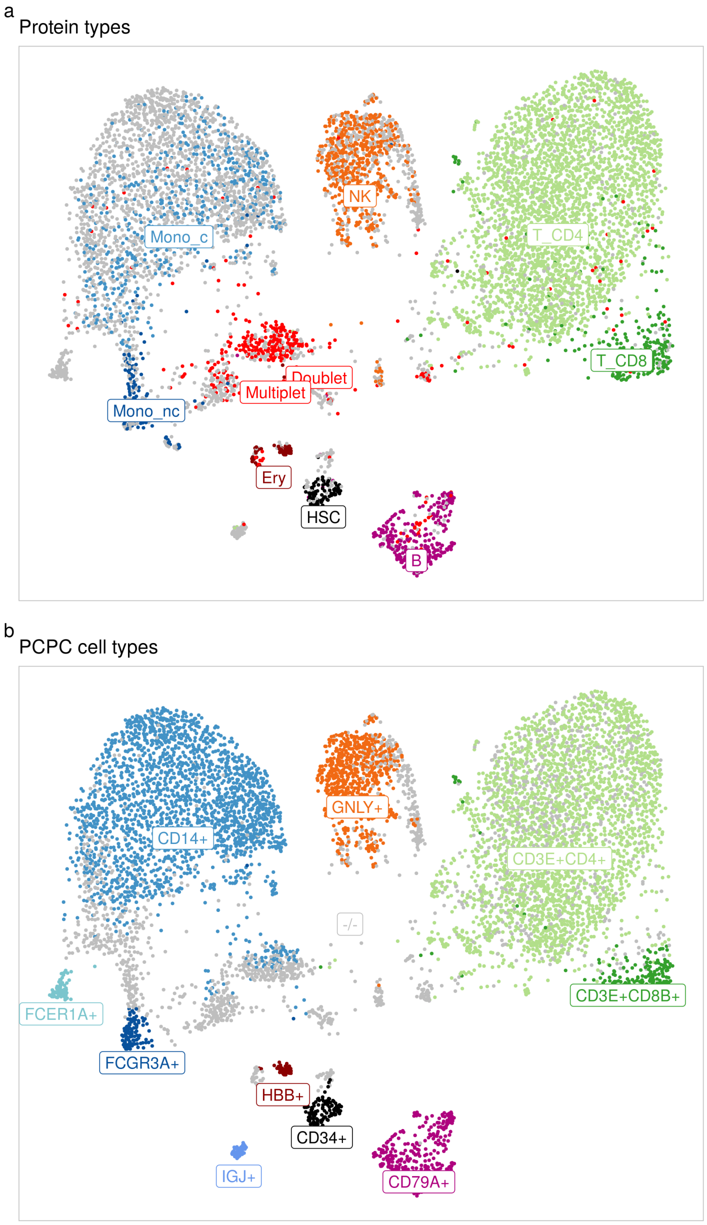


Figure 3.4: PCPC applied to CBMC data, visualized in UMAP embedding. **a** Colors indicate 'protein types', i.e. cell types defined with manual gating on the CITEseq surface proteome data (see methods). Grey cells were left without assignment after stringent gating. **b** Colors indicate cells that PCPC with manual thresholding found positive for the indicated markers. Colors match those in a, and grey cells (labeled with -/-) were negative for all markers. **Symbols:** Mono_c: classical monocytes. Mono_nc: non-classical monocytes. Ery: erythrocytes. HSC: haematopoietic stem cells. NK: NK cells. T_CD4: T helper cells. T_CD8: cytotoxic T cells.

only using sparse scRNAseq data.

3.2.6 Pooled counts result in inverse-variance weighting

I end this section with one last technical considerations on the proposed smoothing method, and why I call kNN pooling a principled approach. We found that the smoothed expression value $\frac{K_{ij}}{S_j}$ has the interesting property that it is the inverse-variance weighted estimator of π_{ij}. Specifically, we found that $\frac{K_j}{S_j}$ is equivalent to averaging the weighted, normalized UMI counts $w_j' \frac{k_j'}{s_j'}$, where j' is amongst the cell j's kNN ($j' \in \mathcal{N}_j$), and the weights are the inverse of the expected variance:

$$w_j = \frac{1}{\mathrm{var}(\frac{k_j}{s_j})} \tag{3.6}$$

Estimators using inverse-variance weighting achieve particularly low variances [156], which we reason will allow to estimate π_{ij} more precisely and potentially lead to better classification results. We now prove that computing $\frac{K_j}{S_j}$ is equivalent to inverse-variance weighting under the assumption that all neighbors $\mathcal{N}_j$ had the same expression rate as cell j. As above, we write this rate, which is common to all neighbors $\mathcal{N}_j$, as $\pi_{\mathcal{N}_j}$. Furthermore, with reference to equation 3.3, we note that a cell j' from this neighborhood has UMI counts distributed according to ():

$$k_{j'} \sim \mathrm{Pois}(s_{j'} \pi_{\mathcal{N}_j}) \tag{3.7}$$

We now find the weights $w_{j'}$ defined in equation 3.6:

$$\begin{aligned} \mathrm{var}(k_{j'}) &= s_{j'} \pi_{\mathcal{N}_j} \\ \mathrm{var}(\frac{k_{j'}}{s_{j'}}) &= \frac{1}{s_{j'}^2} \mathrm{var}(k_{j'}) = \frac{\pi_{\mathcal{N}_j}}{s_{j'}} \\ w_j &= \mathrm{var}(\frac{k_j}{s_j})^{-1} = \frac{s_{j'}}{\pi_{\mathcal{N}_j}} \end{aligned} \tag{3.8}$$

This first line follows from the Poisson distribution itself, whose variance is equal to the Poisson rate. And the second statement follows from the following property of the variance itself: $var(cX) = c^2 var(X)$: the variance of values scaled by a constant is scaled by the square of that constant (see for example [157]). With the weights $w_{j'}$ from equation 3.8, we can now proof that $\frac{K_j}{S_j}$ is the inverse-variance weighted average of $\frac{k_j}{s_j}$ across the neighbors $\mathcal{N}_j$ (for simplicity, I write $\mathcal{N}_j$ as $\mathcal{N}$ in the following):

$$\begin{aligned} \left(\sum_{j' \in \mathcal{N}} w_{j'} \right)^{-1} \sum_{j' \in \mathcal{N}} w_{j'} \frac{k_{j'}}{s_{j'}} &= \pi_{\mathcal{N}} \left(\sum_{j' \in \mathcal{N}} s_{j'} \right)^{-1} \sum_{j' \in \mathcal{N}} \frac{s_{j'}}{\pi_{\mathcal{N}}} \frac{k_{j'}}{s_{j'}} \\ &= \left(\sum_{j' \in \mathcal{N}} s_{j'} \right)^{-1} \sum_{j' \in \mathcal{N}} k_{j'} \\ &= \frac{K_j}{S_j} \end{aligned} \tag{3.9}$$

Note that the first line is the formulated claim, i.e. that $\frac{K_j}{S_j}$ is the weighted average of $\frac{k'_j}{s'_j}$. Again, this is valid only under the assumption that all cells $j' \in \mathcal{N}_j$ have the same expression rate $\pi_{\mathcal{N}_j}$ ($\pi_{\mathcal{N}}$ in above equation for simplicity), and the validity of this assumption is discussed further in section 4.3.

3.3 Resolving complex tissues (MALT data)

Mucosa-associated lymphoid tissue (MALT) is populated with lymphocytes (such as T and B cells) and found next to epithelium such as the skin, eyes or lungs. PCPC performed well on CBMC data, separating T cells from other major blood lineages with high sensitivity and accuracy (see above sections). As naive immune cells, the CBMC data set contains clearly separated cell types, i.e. very distinct populations in transcriptional space (with the exception of two T cell subsets, see above). In this section, I now ask if PCPC is able to separate highly related subpopulations as well. Specifically, I show classification for the MALT data set, which is a complex tumor microenvironment from MALT tissue, with transcriptionally similar T cell subsets. I start this section by demonstrating the MALT data is of higher complexity than the CITEseq data. I show how thresholds are best selected and that this works for low expression strengths as well. I then show cell type annotations from PCPC, and for context also show and discuss differences to annotations from Seurat and Garnett. Thus, this section demonstrates PCPC's ability to resolve complex tissues such as the MALT data set. Practical considerations, such as how to pick marker genes and the number of kNN for pooling, are presented in section 3.4. Finally, I will apply PCPC to a cancer cohort with multiple patients, as a proof of concept, in section 3.5.

3.3.1 Tissue complexity

Abdelaal *et al.* recently benchmarked different classification algorithms [89]. They found that all of the studied algorithms struggle with complex data sets, i.e. whenever the pairwise correlation between cell populations is high [89]. It is thus a crucial question if PCPC is able to discriminate between related subpopulations. To distinguish complex and simple data sets, Abdelaal *et al.* proposed the following measure of tissue complexity: they average the normalized transcriptomes across each cluster, and compute the Pearson correlation between these clusters. More complex tissues will have higher correlations between the contained cell types, captured by high correlations between clusters. To derive a single number, Abdelaal *et al.* furthermore propose take each cluster's largest correlation value and to average these maximal Pearson correlations. The resulting value is by definition below one, and closer to one for more complex tissues. The authors note that the summarized measure also depends on the number of cell types in the data set, not only their similarity.

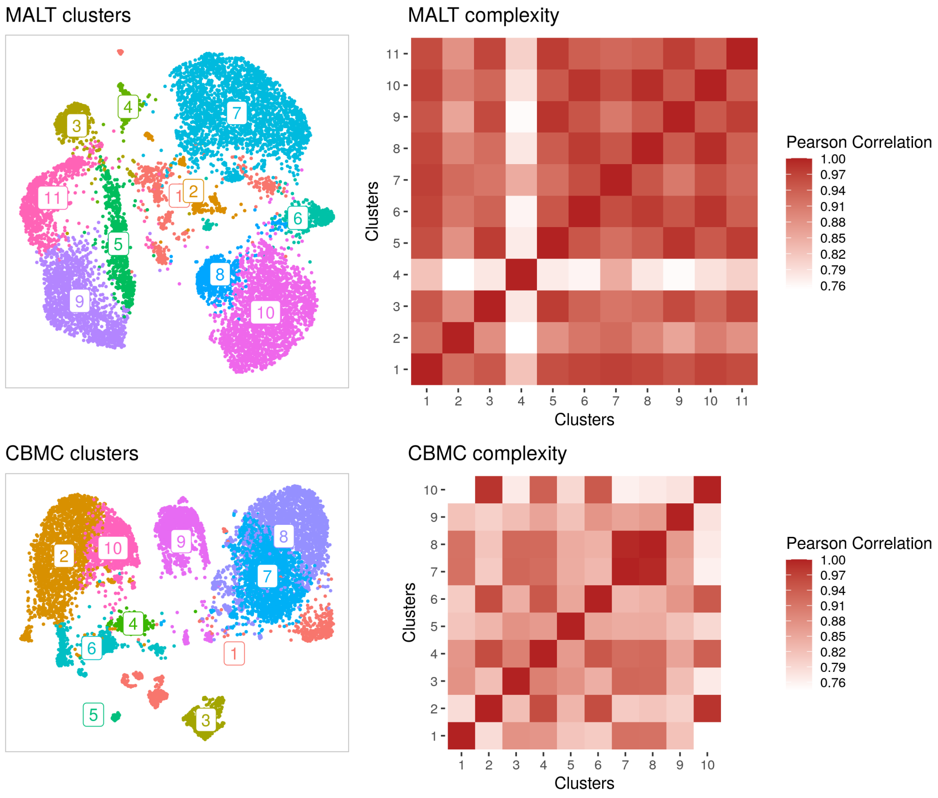


Figure 3.5: Tissue complexity for MALT and CBMC data. The left panels show Louvain clusters computed on 50 nearest neighbors. In the right panels, Pearson correlation between cluster averages are shown, computed on square-root normalized UMI counts and using all genes without filtering. This strategy to measure tissue complexity is adopted from [89].

Figure 3.5 demonstrates that the MALT data set is of much higher complexity than the CBMC data. For the indicated clusters (left panel), the MALT data overall has higher correlations to other clusters (indicated by intense red colors in right panel). For the next section, the T cell subpopulations are of particular interest. They are represented by clusters 3, 5, 9 and 11. As apparent from the top-right panel in Figure 3.5, these clusters also show particularly high Pearson correlations to each other. I conclude that the tissue complexity in the MALT data is higher than in the CBMC data, especially since this fits the biological expectation: the MALT tumor microenvironment can be assumed to have more related cell types from the same major lineage than cord blood cells. Thus, the ability to separate these related T cell subsets is a trait of a good classification algorithm, and PCPC is applied to these T cell subpopulations in the next section.

I end this section by briefly discussing Pearson correlation as measure for tissue complexity. In particular, Abdelaal *et al.* not only used manual inspection of heatmaps, but also summarized such heatmaps into a single value. Using the same approach, the two heatmaps in Figure 3.5 can be summarized into complexity measure values of 0.97 for MALT and 0.95 for the CBMC data (see Methods and [89]). These values are surprisingly similar to each other, given that the heatmaps (right panel in Figure 3.5) show such clear differences. One reason for this might lie in the clustering of the CBMC data. It is debatable if all of the clusters shown in Figure 3.5 (lower left panel) correspond to unique cell types. In particular, clusters 2 and 10 are very similar monocyte clusters, and clusters 7 and 8 are both CD4$^+$ T cells. In both instances, the modularity score optimization by the Louvain algorithm [95] resulted in the observed clusters, and a different clustering algorithm might have decided differently. This dependency on the given clustering output makes the complexity measure proposed by Abdelaal *et al.* lose some objectivity. Instead, a robust complexity measure could measure the maximal correlation between individual cells. Or, if Pearson correlations between individual cell transcriptomes proof very noisy, the averages between nearest neighbors could be used instead of clusters.

A second limitation might be that the proposed score is computed on all genes. It is worth noting that the complexity score might change considerably when genes are filtered first, excluding genes where no biological signal can be detected above technical noise.

Finally, I have observed that the Pearson correlation between clusters depends on each cluster's average library size. For example, clusters 2 and 4 in the MALT data show systematically lower Pearson correlation to all other clusters than the rest. Interestingly, clusters 2 and 4 have the highest and lowest average library sizes: less than 1000 UMI counts on average in cluster 2, more than 14000 UMIs in cluster 4, and roughly around 3000 UMI counts in all other clusters (data not shown). This apparent connection of outlier library sizes and low Pearson correlation could on the one hand be explained by biology: perhaps clusters 2 and 4 simply are more different from the rest than all other clusters. On the other hand, library size is a known confounder in scRNAseq data, and it would make sense that the Pearson correlation depends on it. Although it is beyond the scope of this work, this dependency could be investigated using preprocessing strategies that aim at removing effects from library sizes (such as scTransform [41]).

Overall, the above findings indicate short-comings of the summarized complexity measure proposed by Abdelaal and colleagues. Apparently, their proposal to summarize tissue complexity' into a single number struggles to faithfully capture tissue complexity. Abdelaal *et al.* also inspected individual Pearson correlations in heatmaps [89], and my findings suggest this approach is more apt at discriminating complex cell type mixtures from simple tissues.

3.3.2 Selecting thresholds and expression strength

The above section demonstrated that the MALT data set is more complex than the CBMC data, and resolving complex tissues is a difficult classification task. In this section, I showcase some of PCPC's properties using three selected genes, and derive strategies for selecting marker gene thresholds. This is in preparation of the full cell type annotation of the MALT data set (see Figure 3.7), where I also describe a strategy to select the appropriate marker genes.

Figure 3.6 shows how PCPC detects JUN$^+$, CD3E$^+$ and FOXP3$^+$ cells, which represent relevant cell (sub)types in the MALT data set. Briefly, JUN marks a B cell subset, CD3E marks all T cells and FOXP3 marks regulatory T cells (the final cell type annotations are discussed below, see Figure 3.7). The left

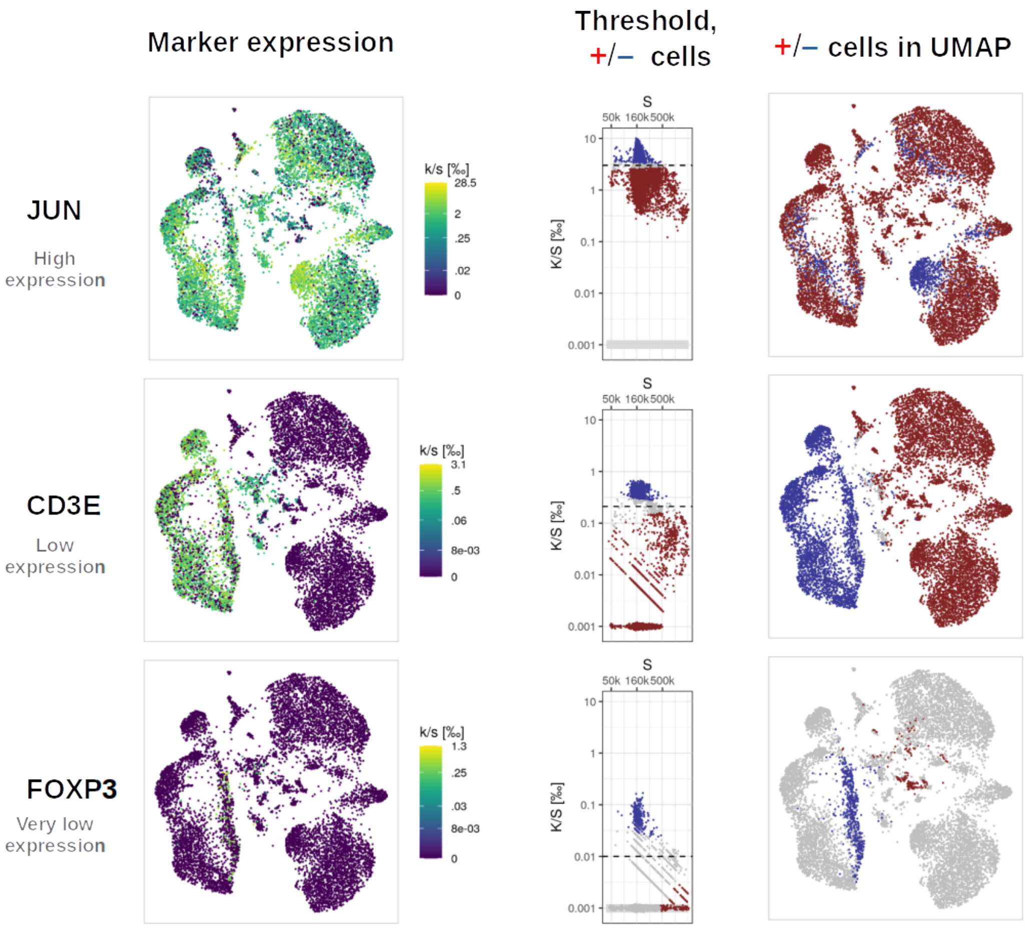


Figure 3.6: Weak and strong marker genes in MALT data. The threshold divides cells into positive (+, blue) and negative (-, red) k: gene UMIs, s: total UMIs, ‰: per mille, K: pooled gene UMIs. S: pooled total UMIs, 50/100/500k: 50/100/500 thousand.

column in Figure 3.6 shows normalized UMI counts, the thresholding process is illustrated in the center, and the right column shows the resulting positive and negative cells. The strategy I propose for finding a suitable threshold is to make the left and right columns in Figure 3.6 congruent. That is, the blue UMAP area in the right column should correspond to the UMAP area with high expression in many cells (left column). By trial and error [5], the most appropriate threshold can then be found, assisted by the smoothed expression plot (central column, introduced in Figure 3.3). Importantly, the positive and negative cells which PCPC finds (blue and red dots in Figure 3.6) are always defined relative to a threshold, not an absolute expression value. JUN, CD3E and FOXP3 in Figure 3.6 serve well to illustrate this distinction between relatively and absolutely negative expression: As evident from the third row in Figure 3.6, most cells are absolutely negative for FOXP3 in the sense that no UMI counts were detected in these cells or in any of their nearest neighbors (apparent as dark purple areas of cells in left column, and cells with 0.001‰ or less in the middle column). Knowing about the sparsity of scRNAseq data, however, this absolute definition of 'negative' is not very useful for classification. That is because zero UMI counts across all neighbors could either mean the mRNA molecules were really not there, or it could mean they were not observed because the marker is lowly expressed (detailed in section 3.2.2). Conversely, JUN expression is much higher and UMI counts were detected in most cells. For classifying JUN$^+$ cells, it is therefore not relevant to find cells that were positive in absolute terms, but rather with relation to the user-defined threshold (dashed line in Figure 3.6, first row, second column). Put differently, absolute UMI values are not helpful in dividing cells according to marker gene expression, it is the fold change to other cell types that is decisive.

Of note, the three genes in Figure 3.6 are expressed at different orders of magnitude, but still PCPC manages to find many positive cells for all markers. This demonstrates that classification can work well for a reasonable range in expression strength. Put into numbers, JUN shows very high expression with up to 10‰ after smoothing (up to 28.5‰ before smoothing), while FOXP3+ is expressed at less than 0.1‰ (1.3‰) in almost all cells. Thus, expression strength of relevant cell type markers spans at least three orders of magnitude in the MALT data, and PCPC is able to identify positive cells for this entire range. This is important, because lowly expressed transcription factors such as FOXP3 can thus be used as markers with PCPC.

One property of PCPC is that the number of unlabeled cells (grey dots in Figure 3.6) depends on the gene's expression strength. JUN is highly expressed and has hardly any grey cells, i.e. almost all cells can be labeled as JUN$^+$ or JUN$^-$. In contrast, FOXP3 has low expression, leaving the majority of cells unassigned (grey in Figure 3.6, bottom right). In other words, we can either identify FOXP3$^+$ or FOXP3$^-$ cells, but not both at the same time. So if we

[5] Whether manual thresholding can be (semi-)automated is explored in sections 3.3.5, 4.2 and 5.2

define T_{regs} as FOXP3^{+}, we need positive markers for the remaining T cell subsets [6]. I note that this is a limitation of scRNAseq data in general: The sparsity makes it difficult to rigorously prove the absence of expression for lowly expressed markers. Classification with PCPC thus requires a certain level of marker expression, and I argue this is even desirable if the goal is to generalize inference results. Using the term coined by Valentinse Svensson, actionable cell types are those that can be identified with a few good markers in many different experimental settings [145, blog post]. The lower the expression of a marker gene, the more likely it becomes that this gene can not be used in other experimental settings due to the respective platform's measurement noise. While it is in principle possible to count unassigned, grey cells as negative, and thus divide T cells into FOXP3^{+} and FOXP3^{-}, it is more advisable to curate lists of positive markers wherever possible instead. I envision that PCPC encourages such positive cell type definitions, gently steering researchers towards cell type definitions that generalize well to many experimental settings. Lastly, I note that it is not unreasonable to expect the expression of many established cell type markers to be sufficiently expressed. After all, they are typically selected for their strong, unambiguous expression in a cell type.

Researchers will often encounter the situation that multiple marker genes can be chosen from. For example distinguishing T_{regs} from other T cells is often done with IL2RA (often called CD25) instead of FOXP3, because IL2RA is a surface protein and thus accessible to cell sorting experiments using flow cytometry. For the transfer of inference results, the exact marker choice is not crucial: Insights into FOXP3^{+} cells from the MALT data can and should be compared with literature findings on T_{reg} cells defined with IL2RA expression. In experiments other than scRNAseq, cell types are often defined using protein technologies (e.g. flow cytometry and immunofluorescence). Depending on the correlation of protein and mRNA levels, not all protein markers can readily be transferred and their mRNA be used by PCPC. To stay with the T_{reg} example, a researcher would then build his scientific case by providing evidence, experimental or from the literature, that FOXP3 mRNA and IL2RA protein by large select the same cells. Alternatively, she or he could repeat the analysis with IL2RA^{+} instead of FOXP3^{+} cells and show that the claimed scientific findings remain unchanged. In other cases, good mRNA marker genes first have to be identified, for example using exploratory analysis of the scRNAseq data set. Specifically, cluster markers found in the scRNAseq data set can be used, and ideally are validated experimentally or showing their ubiquitous utility in other scRNAseq data sets. I will elaborate more on the strategy of how to select markers in section 3.4.1.

In conclusion, the cell type definitions I propose, together with PCPC, allow for clear communication of biological discoveries. Presented this way, a

[6] Typically, I expect positive markers to exist in most settings where PCPC is applicable. In this case, possible subsets of CD3E^{+} T cell are (c.f. Figure 3.7): CD4^{+}ANXA1^{+}, CD4^{+} CXCL13^{+}, CD8B^{+}CCR7^{+} and CD8B^{+}ICOS, next to CD4^{+}FOXP3^{+} T_{regs}.

research finding can be interpreted and commented by colleagues and competitors in light of the cell type definitions used. I envision that this scientific debate enabled by PCPC can help to guide deeper biological understanding in many fields.

3.3.3 PCPC on the MALT data

Above, I showed that the MALT data is a complex classification task, recommended strategies for threshold selection and demonstrated how our method is able to use even lowly expressed marker genes for classification. I now turn to annotating the entire MALT data set with cell type labels. In later sections, I will further investigate the performance of Garnett on this data set, and how clustering with Seurat behaves in the presence of correlated cell types.

The MALT data was generated from a single B cell lymphoma sample and has its name from the tissue of origin: Mucosa-associated lymphoid tissue (MALT). This lymphoma microenvironment is a complex mixture that contains mostly B cells, B cell-derived cancer cells and the T cells that have infiltrated the tumor. With 3000 T cells, roughly a third of the MALT data forms a resource to explore and study T cell subsets in cancer. The process of annotating the MALT data with PCPC yields simple cell type definitions that can be tested, verified and refined by researchers working on related data sets: scRNAseq data sets with similar tumor-infiltrating T cells exist already today for many tumor types, including breast cancer [63], lung tumors [64], pancreatic adenocarcinoma [158] and nodal B-cell lymphoma [62], amongst others. Furthermore, promising single-cell data are also generated with mass cytometry [159, 160]. A common language such as the marker-based naming scheme used by PCPC could therefore help researchers to form hypothesis that can be tested across platforms and cancer entities.

Figure 3.7 shows cell type classification with PCPC for the MALT data. For better visualization, two different color schemes are provided: Figure 3.7a colors by major cell type and Figure 3.7b shows colors with high contrast that are also suitable for most forms of color blindness. The major cell types, as colored in Figure 3.7a, are: CD19$^+$ (B cells and B-cell derived tumor cells), TOP2A$^+$ (yellow, cycling cells), LYZ$^+$ (black, myeloid cells), TTN$^+$ (purple, probably muscle cells), CD3E$^+$CD8B$^+$ (green, cytotoxic T cells) and CD3E$^+$CD4$^+$ (red, helper T cells). As noted above, the tumor-invasive T cell subsets are of particular interest but are difficult to classify due to highly correlated transcriptomes (c.f. tissue complexity in Figure 3.5). Figure 3.7 demonstrates that PCPC resolves them with great detail. For example, T$_{\text{reg}}$ cells (CD3E$^+$CD4$^+$FOXP3$^+$) can be further subdivided into CCR7$^+$ and ICOS$^+$ subpopulations, representing naive-like and effector-like T$_{\text{regs}}$ (see next paragraph). The ICOS$^+$ T$_{\text{reg}}$ subpopulation consists of only

124 cells (1.5% of all cells), indicating that PCPC also works for cells with low abundance. The annotation of the T cell subsets demonstrates that PCPC can resolve fine hierarchies in complex tissues, which is a difficult task for all classification algorithms [89, in particular Fig. 8].

As evident from Figure 3.7, manual annotation with PCPC assigned a cell type label for the majority of cells, leaving only a small proportion without cell type annotation (grey cells in Figure 3.7). Out of the entire data set, the fraction of labeled cells was 82%, which further increased to 87% after excluding potential doublets (not shown, see methods section 2.3). Out of these, only 2.3% had more than one cell type label, showing the chosen markers define mutually exclusive cell types. For comparison, Garnett labels less than 50% of cells using the same markers on the same data (see section 3.3.5). Interestingly, PCPC seems to provide a trade-off between the number of labeled cells and how detailed the cell type definitions are. Specifically, when T_{regs} and cytotoxic T cells ($CD3E^{+}CD8B^{+}$) are classified without subdividing them according to ICOS and CCR7 (not shown), the percentage of labeled cells in the MALT data increases by 3% to 85% (90% after doublet removal). In other words, whenever subpopulations have highly related transcriptomes, such as $ICOS^{+}$ and $CCR7^{+}$ subsets of T_{regs} and cytotoxic T cells, separating these subpopulations comes at the cost of leaving a few cells ‘in between’ unannotated. Thus, PCPC adapts to the signal-to-noise ratio, i.e. if the technical noise is too high to label difficult cells, it does not assign these edge cases. This is in contrast, for example, to clustering cells, where every cells has to be assigned to one of the clusters regardless of its biology. I note that when subtypes can not be assigned by PCPC (e.g. $ICOS^{+}$ and $CCR7^{+}$ T_{reg} cells), then the user can still choose to label them with a broader cell type (T_{reg}, for example). In later sections, I will discuss Garnett’s annotations and Seurat’s clustering of the MALT data in more depth. In conclusion, PCPC is able to thoroughly classify cells from complex tumor microenvironments.

3.3.4 Clustering on MALT data

In the previous paragraph, I demonstrated how PCPC can be used to annotate cell types in the MALT data set. In order to compare these results to the current standard in the field, this section investigates how Seurat’s clustering behaves on the same data. Outside of the scRNAseq field, clustering is an exploratory algorithm, i.e. not suitable for inference, and my observations in this section should be seen as a reminder of this simple fact. I conclude here that clustering and PCPC are complementary, not competing. Garnett [40] offers a more direct comparison because it follows the same marker-based philosophy as PCPC, and I refer the reader to the next section for its cell type annotations of the MALT data.

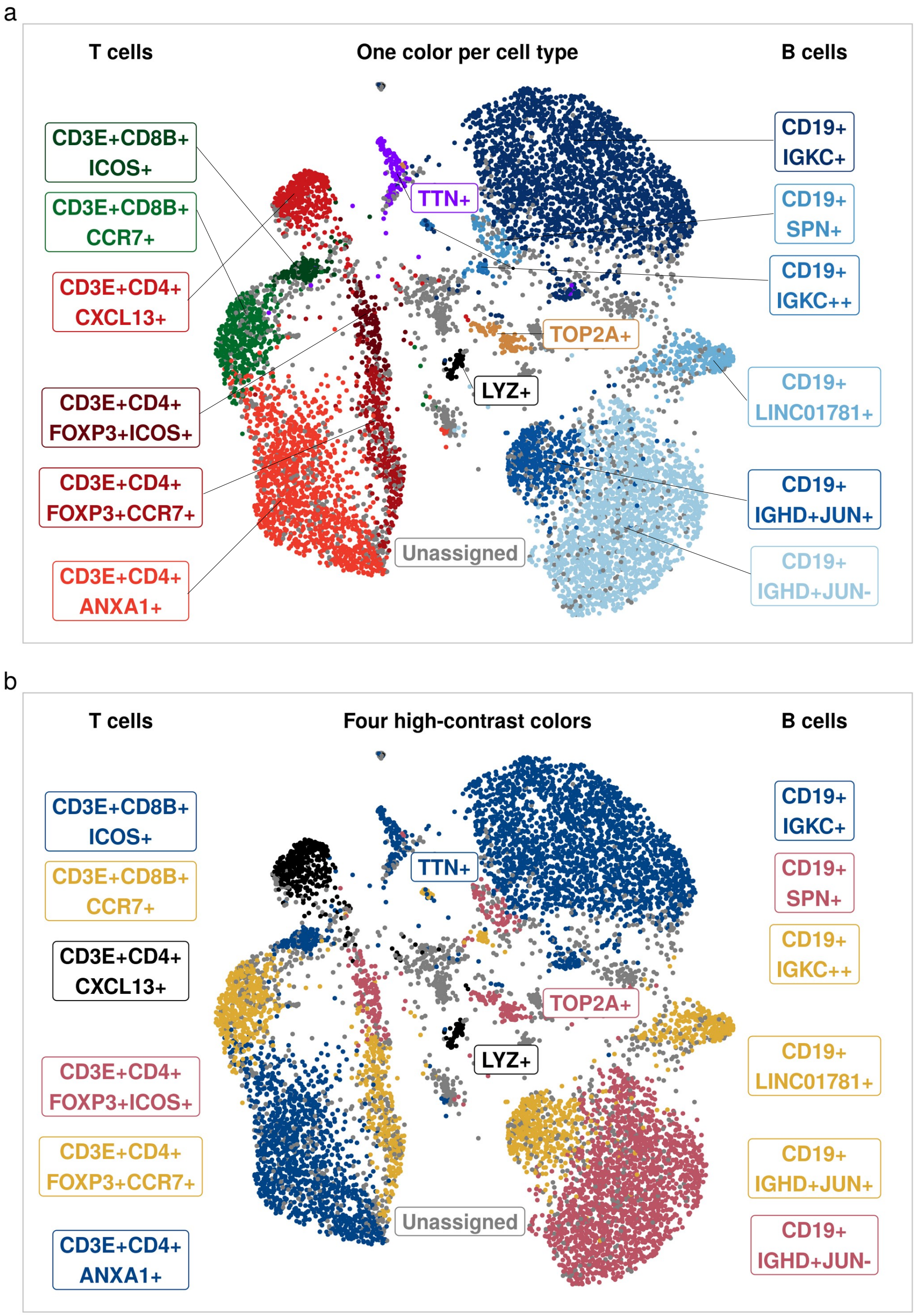


Figure 3.7: PCPC applied to the MALT data set. Cells are colored by the indicated cell types (**a**) or four high contrast colors (**b**, colorblind-friendly). Unassigned cells are shown in grey. Black lines connect cell type definitions (boxes) to the corresponding cells and are omitted in (b) for clarity.

Clustering algorithms are unsupervised, and it is a common notion that this makes them unbiased, in the sense that it is exclusively biological signal, and not prior knowledge, driving the cell type annotations. At the same time, the term ‘unbiased’ is meant to denote objectivity, in the sense that clustering eliminates subjective choices by the researcher. In order to discuss these assumptions of objectivity and lack of subjective bias, I applied Seurat with three different preprocessing strategies that a researcher might reasonably adopt. Specifically, I compare two different normalization strategies that the authors of Seurat suggest as alternative options [161, tutorial]: scTransform (more recent) and log-normalization (more established). For scTransform, I run the workflow with and without removal of potential cell doublets (see methods section 2.3), representing a second subjective researcher decision. If clustering were truly unbiased (independent of subjective choices), these subjective choices should not change the resulting cell labels much. Figure 3.8 shows the Seurat clusters for all three preprocessing strategies. The colors were matched with those in Figure 3.7, and for clarity I will refer to subpopulations with the names used in Figure 3.7. As evident from Figure 3.8, clustering is not unbiased but is instead influenced by various subjective decisions. For example, cluster 0 in Figure 3.8a (CD19+IGKC+ cells) is split into two clusters when removing doublets (Figure 3.8b). In other words, for cluster 0 in the MALT data, the notion that clusters are cell types creates the paradox that removing cells increases the number of cell types present.

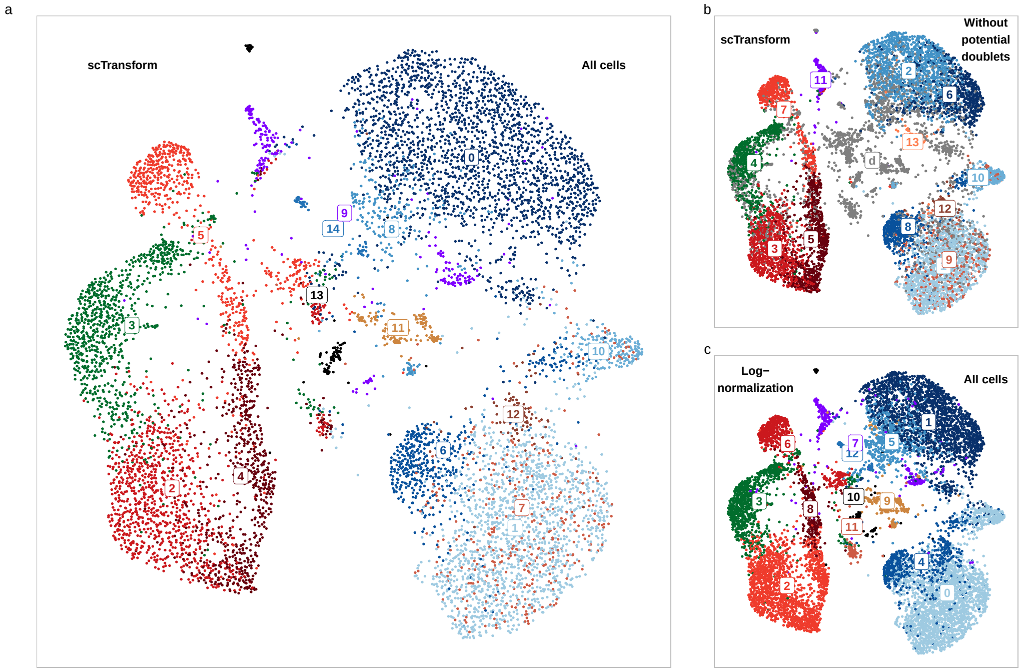


Figure 3.8: Clusters on MALT data depend on preprocessing. Seurat clusters were computed after normalization with scTransform, either using all cells (**a**) or after removing potential doublets (**b**, see methods section 2.3), and with log-normalization and all cells (**c**). Default parameters were used (e.g. resolution=0.8). Clusters are visualized on the same UMAP embedding used above (computed as described in methods section 2.1). Although not used for clustering, potential doublets are still shown in **b** as grey cells, for clarity.

As noted elsewhere, the same paradox occurs when cells are removed at random [2, preprint]. Another example for bias in clustering is provided by the $\mathrm{T_{reg}}$ cells. While PCPC extracted a distinct group of $\mathrm{CD3E^+CD4^+FOXP3^+}$ cells (vertical, longish in the UMAP embedding, c.f. Figure 3.7), Seurat's algorithm distributes them to two separate clusters. Both of them, namely clusters 4 and 5 (Figure 3.8a), contain not only $\mathrm{T_{regs}}$ but also cells without any evidence of FOXP3 expression (c.f. Figure 3.6), suggesting that clustering is at odds with the well established $\mathrm{T_{reg}}$ cell type definition. Graph-based clustering is the state-of-the-art for cell type classification, and was unable to clearly identify $\mathrm{T_{reg}}$ cells on the MALT data. PCPC thus removes obstacles in studying $\mathrm{T_{regs}}$ in human tumors. For example, it was able to resolve a $\mathrm{ICOS^+}$ subset (see section 3.3.3) that resembles the $\mathrm{ICOS^+}$ $\mathrm{T_{reg}}$ cells recently reported in a cohort of lymphoma patients [62, Fig. 2 therein]. As in the MALT tumor, these were B cell-derived malignancies which can be studied in finer detail using PCPC.

Another conclusion from Figure 3.8 is that clusters do not always capture biologically relevant cell types. For example, cluster 5 in Figure 3.8a contains $\mathrm{CD3E^+CD4^+CXCL13^+}$ cells (T follicular helper cells [7]), and a group of potential doublets, next to $\mathrm{FOXP3^+}$ cells. This clustering result, peculiar as it is, is not robust to pre-processing: Using log-normalization (Figure 3.8c) yields different cluster labels for these T cell groups. These strong differences between preprocessing strategies show that unsupervised clustering is not unbiased, but instead depends strongly on subjective preprocessing decisions.

Interestingly, modern clustering algorithms offer a solution for including subjective researcher decisions. Namely, Seurat has the parameter 'resolution', which can be used to produce more and smaller clusters. So in principle, this resolution parameter might be used to resolve the $\mathrm{T_{reg}}$ subpopulations after all. For example, Seurat might well be able to identify subsets of cytotoxic T cells (green in Figure 3.8, c.f. Figure 3.7) when a larger value for the resolution parameter is used. For exploration, this approach is most valuable: Finer and finer clusters can be tested for whether they are 'real' [121, blog post] and which marker genes they might have. For inference, I argue this approach is undirected and not goal-oriented. For many scientific questions, the literature dictates marker genes that have to be used in order to draw convincing conclusions regarding a certain cell type. While this is the declared goal of PCPC, clustering was built for exploration and using its resolution parameter does not guarantee the desired result is obtained. For example, we saw above in Figure 3.6 (top row) that JUN expression subdivides $\mathrm{CD19^+}$ $\mathrm{IGHD^+}$ cells into subpopulations (clusters 6 and 1 in Figure 3.8a), but also marks tiny subsets of helper T cells (cluster 2), cytotoxic T cells (cluster 3), $\mathrm{CD19^+IGKC^+}$ cells (cluster 0) and $\mathrm{CD19^+LINC01781^+}$ cells (cluster 10). A researcher interested in the $\mathrm{JUN^+}$ subset of any of these subpopulations

[7] I refer to $\mathrm{CD3E^+CD4^+CXCL13^+}$ cells from the MALT data as T follicular helper cells here, because they highly express many of the specific markers defined by Roider *et al.* [62, Fig. 2 therein] (not shown)

would have to choose smaller and smaller values for the resolution parameter, hoping Seurat at one point would produce a cluster that exactly overlaps with these cells. As defined by Crowell *et al.*, a subpopulation can be any group of cells for which it is interesting to ask inference questions [65]. The JUN$^+$ T cells are just one hypothetical example illustrating the flexibility required from inference-oriented classification methods, highlighting the need for tools such as PCPC. To bring it to the point, the ignorance to existing prior knowledge is a strength in exploration, but a nuisance in inference. Thus, clustering is a powerful tool for exploring which cell types are present in a data set, but is not flexible enough to accommodate the diversity of scientific inference questions.

Admittedly, PCPC requires substantial manual work before yielding cell type annotations. Still, I argue this time is a good investment and point out the hidden labor that clustering has: If clusters were to be used for inference, a rigorous researcher would re-cluster his data again and again with different choices of analysis strategies, including normalization, clustering parameters (e.g. Seurat's resolution) or even the clustering algorithm itself. Out of these results, he would then use his experience and prior knowledge to select the most appropriate clustering result. If multiple data sets are analyzed together, one would on top iterate through different parameters and/or methods for batch correction. The term 'unbiased' is therefore an unfortunate choice, as it glosses over this manual human labor that is always required by rigorous science, even when using unsupervised methods. With PCPC, I argue the invested time is well directed towards the scientific problem that the researcher tries to answer. Importatnly, the personal bias he introduces is not hidden in abstract preprocessing choices, but visible as the selection of marker genes. Thus, I argue it is always the researcher who decides the cell type annotations, even when using unsupervised algorithms such as clustering.

From the observations in this section, I draw the following conclusions. Clusters do not directly correspond to cell types in the MALT data. Instead, clustering is a powerful tool for exploring the cell types present. PCPC complements this exploratory approach with a flexible classification method that adapts to various inferential research questions. Thus, instead of having a single algorithm for all applications, I argue that exploration and inference have different requirements for cell type labels, calling for separate, specialized algorithms.

3.3.5 Garnett on MALT data

In the sections above, I showed that PCPC is able to resolve fine cell type hierarchies in the MALT data. In particular, not only higher cell type levels such as CD3E$^+$CD4$^+$ can be annotated, but also lower levels of fine-grained definitions, such as CD3E$^+$CD4$^+$FOXP3$^+$ICOS$^+$. This is encouraging, because such "deep levels of annotation" [89] are difficult for all classification methods [89]. To put the results into context, I next ask how Garnett [40] performs on the MALT data. While many computational tools exist for cell type classification, comparing Garnett and PCPC is particularly instructive because both use marker-based cell type definitions. These are simple and transferable, so Garnett already fulfills two out of three criteria introduced in section 3.1. The question in this section is consequently to address the third trait of good cell type definitions: Are Garnett cell type annotations useful, in the sense that many cells from a given scRNAseq data set can be labeled?

Garnett's annotations of the MALT data To allow a direct comparison to PCPC, I tested Garnett on the MALT data, and using different levels of cell type hierarchies. For the finest cell type definitions, I used the same marker genes as with PCPC in Figure 3.7 above. For cell type hierarchies of coarse and intermediate level of detail, I left out genes marking finer subtypes (e.g. ICOS, CCR7, FOXP3, IGKC, etc.) as stated in the methods and in the caption of Figure 3.9. The cell type labels for these three hierarchy levels are shown in Figure 3.9. In all three cases, Garnett failed to label a large proportion of the cells. From coarsest to finest hierarchy, the percentage of labeled cells were 45, 39 and 23%. Garnett has the option to fall back to Louvain clustering so that more cells can be labeled. When using

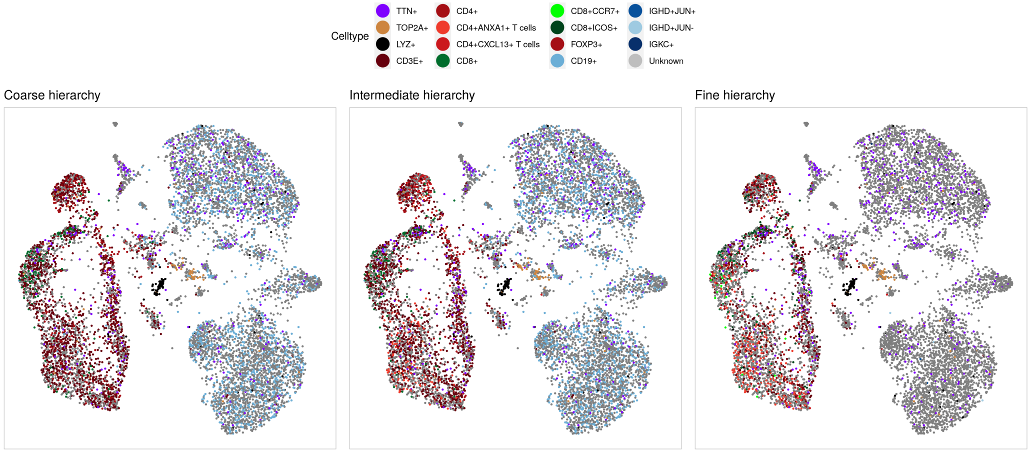


Figure 3.9: Garnett [40] applied to MALT data with different levels of hierarchy detail. The colors were matched fo Figure 3.7 where possible. The coarse hierarchy uses only major lineage markers (TTN, LYZ, TOP2A, CD19, CD3E) and two for defining helper and cytotoxic T cells (CD4$^+$ and CD8B$^+$). The intermediate hierarchy further subdivides helper T cells into ANXA1$^+$, CXCL13$^+$ and FOXP3$^+$ (T$_{\text{regs}}$). The fine hierarchy subdivides T$_{\text{regs}}$ and cytotoxic T cells with ICOS and CCR7, and also separates CD19$^+$ cells as in Figure 3.7 using IGKC, IGHD, SPN, LINC01781 and JUN.

these ‘cluster-extended types’, 52, 53 and 58% of cells were assigned a cell type, indicating that Garnett often times was unable to match clusters to cell types. Thus, Garnett annotates less than half of the cells in the MALT data, and this does not improve much when using cluster-extended types. Garnett’s low labeling success is not entirely surprising, given what the authors themselves report [40]. Even on blood cells from a healthy donor, i.e. a rather simple cell mixture, Garnett labeled only 74% of cells [40, 71% correct plus 3% uncorrect labels]. To test Garnett on more complex tissues, the authors used lung tissue from the mouse cell atlas [8] and Tabula muris [52]. When labeling only the major cell types without subsets (B cells, T cells, endothelial cells, etc.) in these two lung data sets, Garnett achieved 58% and 71% correctly annotated cells, respectively, leaving a third and a quarter unassigned [40, c.f. Fig. S4 therein]. When the authors applied Garnett to a finer cell type hierarchy in the mouse brain, few cells were assigned a cell type at all (only 30% correctly assigned, see [40, Fig. S9 therein]).

Garnett labels fewer cells than PCPC Having observed Garnett’s performance now gives some context to evaluate PCPC. In section 3.3.3, I showed that PCPC has labeled more than 80% of cells in the MALT data, with highly detailed cell type definitions (e.g. CD3E$^+$CD4$^+$FOXP3$^+$ICOS$^+$). In comparison, Garnett labeled less than 50% of cells, making PCPC’s performance a most encouraging result.

Garnett misclassifies more cells than PCPC Next to labeling few cells overall, Garnett also struggled to assign some of the cell types as intended. Most prominently, Garnett hardly found any CD19$^+$ cells (blue in Figure 3.9), labeled many B and T cells as TTN$^+$ cells (purple in Figure 3.9) and was unable to tell CD4$^+$ and CD8B$^+$ T cells apart. Also, many CD4$^+$ T cells remained unassigned (center and right panel in Figure 3.9) or were labeled with the more general label CD3E$^+$, again indicating that Garnett struggled to distinguish them from CD8B$^+$ T cells (left panel).

I next asked how Garnett’s ‘cluster-extended’ mode performs on the MALT data. Comparing Figure 3.10a and b, I observe it did not drastically improve classification: Garnett still was unable to identify TTN$^+$ cells (purple) and to separate CD4$^+$ and CD8B$^+$ cells (green and bright red). I now consider TTN$^+$ cells (purple in Figure 3.10a) in more detail to discuss why Garnett fails in labeling cells. This is in preparation for discussing the challenges in automating cell type classification in general (see following section and also section 4.2). As evident from Figure 3.10c, TTN is detected in many cells all across the UMAP embedding, and this causes problems when Garnett internally chooses which cells to use when training the logistic regression classifier. Indeed, during training, Garnett used 345 cells to represent the class of TTN$^+$ cells (not shown), whereas only 116 TTN$^+$ cells exist in the data (as judged by PCPC, c.f. Figure 3.7). When picking training sets, Garnett

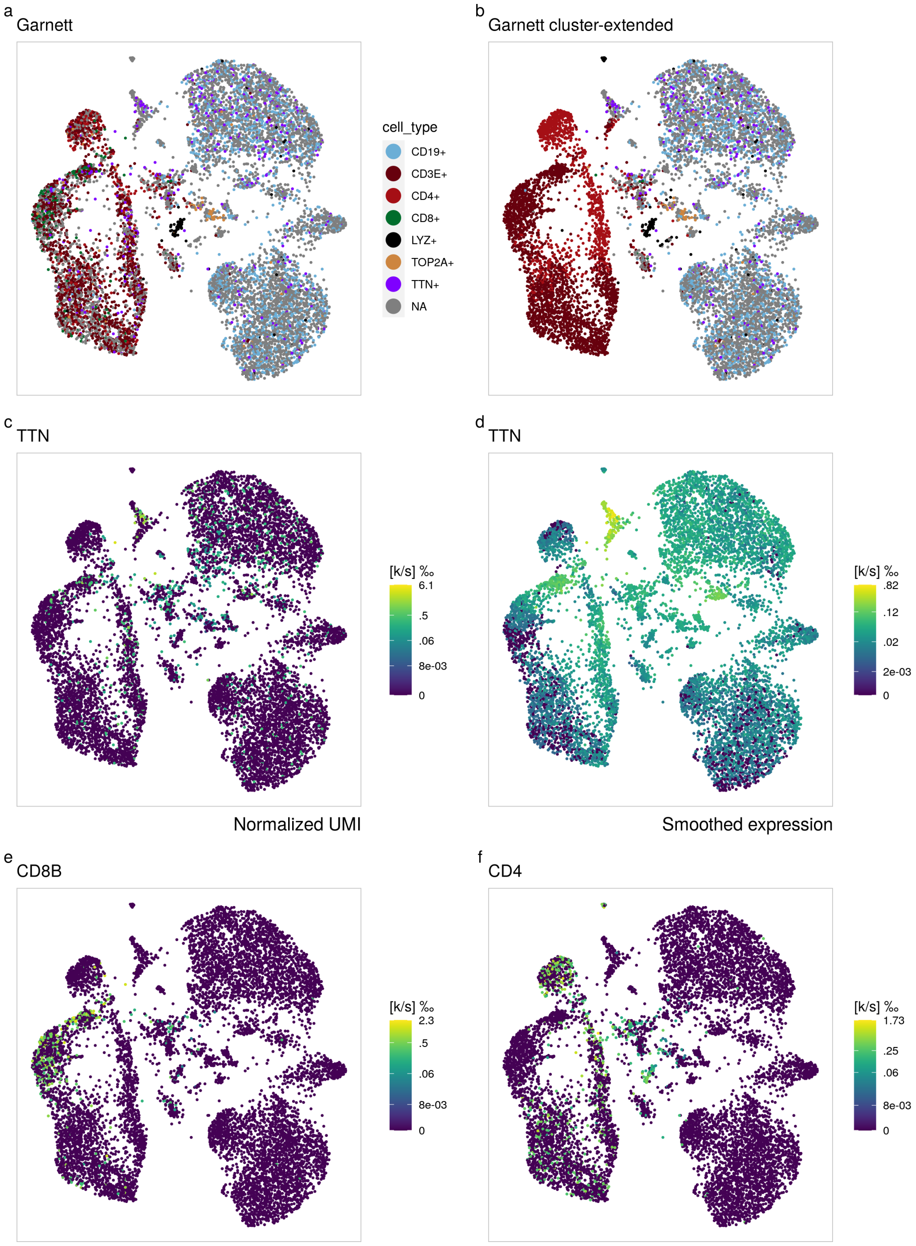


Figure 3.10: Garnett [40] applied to MALT data.

uses the reasonable assumption that the highest expressing cells represent the marked cell type. In practice, however, it is not straight forward to determine these highly-expressing cells. Specifically, Garnett uses a heuristics to decide which cells to include: It considers all cells above a threshold, and this threshold is automatically computed as 25% of the expression's 95th precentile. In the case of TTN, it appears that this rule-of-thumb selected too many cells (specifically: 345 cells) for training, including many CD19$^+$ and CD3E$^+$ cells, resulting in final annotations that frequently mislabel cells as TTN$^+$ (namely: purple dots in Figure 3.10a,b). In other words, the correct cut-off to label TTN$^+$ training cells would have been at higher expression, but the hard-coded heuristics Garnett uses failed to recognize this.

Another instance to discuss challenges in automated cell type assignments are the CD4$^+$ and CD8B$^+$ cells. Garnett was unable to tell these two T cell subsets apart, as evident from comparing Figure 3.10a,b with the marker expression in Figure 3.10e,f. Strikingly, the cluster-extended mode of Garnett was apparently misled by the irrelevant bimodality of CD4. Specifically, it failed to label the lower half of the CD4$^+$ T cells (compare Figure 3.10b and f) and resorted to labeling them as CD3E$^+$ T cells instead. Also, the cluster-extended type was unable to identify CD8B$^+$ cells at all (absence of green dots in Figure 3.10b). This might be because Louvain clustering tends to return rather large clusters in scRNAseq data in my experience (see also Figure 3.5 for Louvain clusters of the MALT data), and perhaps this is at fault here: CD8B$^+$ cells might simply have ended up in same cluster as (some) CD4$^+$ cells, so that Garnett was not confident in deciding for the one or the other. This prevented Garnett obviously from resolving finer cell type hierarchies as well: FOXP3$^+$ are not detected at all (Figure 3.10), let alone subdivided into CCR7$^+$ ICOS$^+$ subsets.

Finally, I speculate on why CD19$^+$ cells were not annotated well by Garnett. Garnett uses unsmoothed, normalized counts, and it is possible that simply not enough genes are correlated with CD19. MS4A1 is a highly expressed marker in these cells, but it is also expressed by TTN$^+$ and TOP2A$^+$ cells in the MALT data (not shown). Obviously, this confused Garnett in the first logistic regression iteration where it tried to separate the major lineages CD19$^+$, CD3E$^+$, LYZ$^+$, TTN$^+$ and TOP2A$^+$ (see marker files in supplement A). I note that it is a short-coming of my analysis here that I have not defined TTN$^+$ and TOP2A$^+$ cells as subset of CD19$^+$ cells. It is possible to improve the marker file and thus perhaps CD19$^+$ cell classification. Compared to PCPC, I note that this type of error-correction is tedious since Garnett has long computation times and it is intransparent which cells were selected for training. Even if CD19$^+$ cells were not classified due to an inadequate marker file, the T cell subsets could also not be resolved to Garnett, leaving my conclusions from this section valid.

Automating marker-based classification is challenging In this section, I use some of the above examples to illustrate why automating cell type classification based on few markers per cell type is challenging in general. I discuss this more concisely in section 4.2.

As a first example, I continue discussing TTN$^+$ cells from the previous section. Figure 3.10c shows that while cells with non-zero TTN expression include many B and T cells, the expression there is below 0.5‰ (greenish), while the actual TTN$^+$ cells show larger expression (yellowish). This difference becomes even more evident when the smoothed values are displayed (Figure 3.10d). This clear expression difference allowed me to find TTN$^+$

and TTN$^-$ cells with PCPC (see section 3.3.3). In contrast, the heuristics implemented by Garnett failed doing so (see previoius section). This could be a general problem for automation: For a human, spotting the correct threshold interactively is trivial even without Figure 3.10d as guidance, while implementing this decision with algorithmic heuristics is challenging.

TTN and its ambiguous threshold is not a rare example. In fact, CD4 in the MALT shows at least three expression regimes (Figure 3.10f): Very high, moderate and undetected. In particular, the T cells (in the left half of the embedding, c.f. Figure 3.7) show very high CD4 expression in the upper UMAP area, low expression in the lower, and no expression at all on the left. I note these three regimes are more evident from smoothed expression not shown here, but the unsmoothed expression in Figure 3.10f gives an idea. An algorithm now has the difficulty to decide the following question: given the user-provided knowledge that there are CD8B$^+$ and CD4$^+$ cells, how should the T cells be divided? Should only the very high CD4 expression count, while the moderately expressing cells should remain unassigned? Or are both to be assigned as CD4$^+$ cells? While difficult to capture with algorithmic rules, to a human observer, the answers to these questions are more obvious. Using the prior knowledge that CD8B and CD4 are mutually exclusive in T cells (except in very rare exceptions such as progenitors in the thymus), the human would compare CD4 and CD8B expression (Figure 3.10e and f) and reason as follows: While CD4 expression is surprisingly weak in the lower UMAP half, the otherwise highly expressed CD8B is hardly detected at all, so it is reasonable to label these cells as CD4$^+$. This rational can be further consolidated by an interactive user through goal-oriented exploration. For example, a trained biologist would also consult CD8A and CD40LG to see that they are correlated with CD8B and CD4, respectively (not shown). Thus, he would with ease know that the cells with moderate CD4 expression are still to be labeled as CD4$^+$, and would pick a threshold with PCPC accordingly. And there is another strength of manual thresholding. In cases more debatable than this simple T cell subsetting, scientific peers and reviewers might challenge the classification, and the above arguments can then be brought forward in a scientific debate. By contrast, for a marker-based algorithm provided with only CD4 and CD8B, this reasoning is impossible and can result in annotations that fail to capture the intended cell types.

Garnett conclusions Garnett can go from marker-based cell type definitions to cell type labels in a fully automated manner, without requiring as much manual labor as PCPC. Also, classifiers can be transferred to new data sets. Still, I conclude that Garnett is not a direct competitor for PCPC. As the authors note, it is instead for “rapidly annotating atlas” [40] data sets. I see this application has value in the exploratory setting of large studies. With more and more available data sets, Garnett’s trained classifiers will become better and better, helping scientists to explore new data sets faster and with more confidence. By contrast, PCPC addresses the need for a marker-based method intended for inference. Here, it is crucial that the majority of cells is labeled, so that hypothesis testing such as differential gene expression can harness as much information as available.

As noted above, PCPC labels less cells when finer cell type definitions are used. Interestingly, Garnett shows the same trend, i.e. the coarsest hierarchy allowed for labeling the largest proportion of cells (45%, compared to 23% with the finest hierarchy). This suggests that a trade-off exists between cell type resolution and confidence: Telling highly similar subpopulations apart comes at the price of leaving many cells 'in between' unassigned. It is a good sign for both methods that classification with PCPC and Garnett reflects this trade-off in the number of labeled cells, as this indicates conservative results whenever the technical noise is higher than the biological signal.

Lastly, I note Garnett might perform better when more marker genes are used, or more work is invested when designing the marker files (marker files are in supplement A). For example, Garnett allows manual thresholds. I note that rerunning Garnett with different threshold values becomes unfeasible with higher numbers of genes / cell types, and is probably outside of its intended use case. Here, the short computation times and interactive usage of PCPC are more appropriate.

3.4 Practical notes for using PCPC (MALT data)

This section illustrates how I envision PCPC to be used by (computational) biologists. Using the MALT data as example, I first describe how marker genes can be selected. I next show how PCPC encourages researchers to clearly define their scientific question.

3.4.1 Marker selection strategy

I now describe how I selected the marker genes to annotate the MALT data, i.e. the genes used in Figure 3.7. This illustrates how I envision PCPC will be used by (computational) biologists. For any potential marker gene, it starts by visualizing its expression in the UMAP embedding (a so called feature plot, see for example Figure 3.6, left column). UMAP places related cells next to each other, and a suitable marker stands out by high expression in a specific, local area of the embedding. Choosing between different candidate markers is then a process of comparing their feature plots and the resulting positive cells (c.f. Figure 3.6, left and right columns), and consulting the literature. This strategy of selecting the right marker from a few potential candidates can be assisted by the interactive code presented in the outlook (section 5.2). I now give concrete examples for markers used in Figure 3.7. I obtained CD3E, FOXP3 and CD19 from the literature, as they represent common markers for T cells, T_{regs} and B cells, respectively. All three markers in principle have several alternatives. Next to CD3E, for example, other components of the T cell receptors would be good candidates, such as CD3G, TRA or CD247 (also known as zeta chain). I chose CD3E because it is expressed higher than the other components and the corresponding protein complex (CD3) is widely used as pan-T cell marker in flow cytometry. This last point is important, because inference results obtained for $CD3E^+$ cells found by PCPC can be expected to have a direct correspondence to legacy knowledge in the literature, i.e. results obtained from cells sorted with anti-CD3 antibody - assuming sufficient correlation of mRNA and protein levels. In the case of JUN (also: c-JUN, AP-1), a targeted literature search confirmed its role in B cell lymphoma [162, 163], after JUN had come up as cluster marker. Specifically, I first clustered the MALT data with the Louvain algorithm (for clusters see Figure 3.5). Then, as part of exploratory analysis, I ranked genes by their fold change between a given cluster of interest and all other cells (see methods section 2.7). JUN was amongst the top markers (marking cluster 8 in Figure 3.5), i.e. genes with the highest expression difference to all other cells. With the same strategy, I found CCR7 and ICOS, which divide the T_{regs} into two subsets [8] (see Figure 3.7). CCR7 marks naive-like T_{regs} [164], and I note that an alternative choice in the MALT data would be SELL (CD62L). ICOS marks cells that are also $ITGAE^+$ (known as CD103, not shown), which marks effector-like T_{reg} cells [164]. These two T_{reg} subsets thus illustrate that often, a researcher may choose from a few marker genes that would give highly similar classification outcomes. Further below, I explain how I chose ICOS over ITGAE and CCR7 over SELL, and how this flexibility of PCPC is helpful in adapting to the exact scientific question.

Instead of actively choosing, one could in principle use multiple markers in parallel. For example, T_{regs} could be discriminated from other $CD3E^+CD4^+$ cells using FOXP3 and IL2RA together. With PCPC, this is achieved with simple logical operations: A T_{reg} cell would then be required to be both $FOXP3^+$ and $IL2RA^+$. As a consequence, the same or fewer cells would be labeled using both markers compared to using only one of them, but never more than that. This strategy is advisable if purity is more important than yield, i.e. if mislabeled cells should be avoided at the cost of labeling fewer cells. I note, however, that the goal of PCPC is to use as few genes as possible for classification to keep the cell type definitions simple. So instead of requiring T_{regs} to be $FOXP3^+IL2RA^+$, a researcher could rather demonstrate that his main findings are the same, irrespective of whether T_{regs} are defined as

[8] Conveniently, CCR7 and ICOS also distinguish two subsets of cytotoxic T cells in the MALT data.

$FOXP3^+$ or as $IL2RA^+$. Thus, a researcher can convince himself his findings are robust, and help his colleagues to interpret his results in context of the literature that might sometimes use FOXP3 and other times use IL2RA.

In summary, choosing marker genes is a process that involves exploratory analysis and consulting prior knowledge from the literature. I envision this strategy to be effective in most biological settings. It is particularly apt to classify cell types in well researched cell mixtures, such as immune cells from mouse or human. This scenario is encountered in many studies with an emphasis on inference, such as case-control cohorts of patients profiled with scRNAseq (see for example [60–64]). Here, the goal is to gain disease-specific insights into well-known cell types such as T cell subsets. I argue that the inference results from such a cohort study are more tangible with the unambiguous cell type definitions provided by PCPC. Still, even if no prior knowledge on cell types and their markers were available at all, PCPC could in principle work exclusively by using genes found as cluster markers during exploration. An example for this could be a scRNAseq study that builds a gene expression atlas of a rarely studied organ, species or disease. In contrast to the inference-oriented cohort study above, this is a purely exploratory setting, in which PCPC would aim at proposing novel cell type definitions that can be tested by others. More clearly, exploratory tools such as clustering [37, 95, 101] and label transfer from annotated reference data sets [73, 84, 122, 138, 165] are central in such exploratory studies, while PCPC could be used at the end to communicate the novel cell types in a simple, transferable and useful manner [9].

3.4.2 PCPC adapts to the scientific question

With PCPC, it is trivial to adapt cell type definitions to the scientific need, so that the same data can be used to answer various research questions. Using the T_{reg} cells from the MALT data, I give two concrete examples.

The first example is how to choose between ICOS and ITGAE, both of which seem to mark a similar group of cells representing effector-like T_{regs} (not shown, c.f. section 3.4.1). In the literature, ITGAE is established as a marker for this population [164], making it a good choice to connect to 'legacy knowledge'. In contrast, we recently observed $ICOS^+$ T_{regs} in a related cancer entity [62, Fig. 2 therein], and could now ask if this subpopulation also exists in the MALT data set. Thus, the marker choice depends also on the scientific question: Do we generate new insights into previously described cell types, or are we re-visiting the MALT data to perhaps generalize the results obtained from a new study [62]. Another consideration is the experimental follow-up of any findings the scRNAseq data might bring. In contrast to ITGAE, ICOS

[9] Simple, transferable and useful are used here in the sense introduced at the beginning of this section.

can also mark a subset of cytotoxic T cells (see Figure 3.7), which is useful when resolving cell types with as few markers as possible is desirable. This is the case for example in flow cytometry, where more than a handful of markers are challenging on most instruments. When sorting all four populations (two T_{reg} and two cytotoxic), the number of required marker is reduced by two when choosing ICOS and CCR7 over of ITGAE and SELL. This ability to switch between marker genes makes PCPC a flexible tool to harmonize the scRNAseq data analysis with the literature and follow-up experiments.

The second example for goal-oriented classification considers cell type definitions at different levels of detail. A researcher might first be interested in the abundance of a very specific T_{reg} subpopulation, such as CD3E$^+$CD4$^+$ FOXP3$^+$ICOS$^+$ cells (dark red in Figure 3.7). As noted above, an investigator may select them with PCPC in his/her own data and also in related tumors, such as B cell lymphoma [62, see Fig. 2c therein], and compare pathway activities or cell abundance, for example. Next to such questions concerning detailed subsets, an independent endeavor could be to investigate all CD3E$^+$CD4$^+$FOXP3$^+$ T_{regs} together. Indeed, I observe that UMAP has arranged the MALT T_{regs} vertically in the embedding, so that they form a distinct elongated shape. Explorations of the MALT data with feature plots (not shown) suggest this longish appearance represents a gradient of T_{reg} phenotypes, going from a naive-like CCR7$^+$ to an effector-like ITGAE$^+$ phenotype [164]. Specifically, going from lower to higher locations in the UMAP embedding, T_{regs} highly express SELL and CCR7, then CCR6, then IL32, then CTLA4, then ICOS, then TNFRSF18 and finally ITGAE/CD103 (not shown). Thus, a researcher could use PCPC to extract all CD3E$^+$ CD4$^+$FOXP3$^+$ and subject them to analysis with pseudotime tools [39, for example] in order to describe and characterize the spectrum of T_{reg} transcriptomes. Equally, any higher level in this cell type hierarchy could be chosen: all helper cells with CD3E$^+$CD4$^+$, and all T cells with CD3E$^+$.

In conclusion, I propose that answering different questions with the same data set requires flexible classification as implemented in PCPC. I argue that ‘the’ ideal cell type annotation does not exist, but instead has to adapt to the exact research question. This way, a scientific argument can be built more clearly and persuasively.

3.4.3 Pooling bandwidth: Bias and variance

PCPC uses kNN pooling, raising the following questions. How valid is kNN pooling? Is this likely to introduce artifacts, and what would be the consequences? How to choose the ‘pooling bandwidth’, i.e. the number of kNN to use? This section discusses these points, and I also refer the reader to section 4.3, which provides a focused discussion on how to find appropriate kNN.

A useful statistical concept for considering the bandwidth is the bias-variance trade-off. Briefly, pooling counts across more kNN will make the smoothed value ($\frac{K}{S}$, see 3.2) less noisy, decreasing variance. At the same time, however, increasing the pooling bandwidth will also include cells with different cellular states, biasing the individual estimate towards a false estimate (namely, the global average of the entire data set). In order to understand this trade-off, I smoothed FOXP3 expression across increasing numbers of kNN for two related subpopulations. To avoid circularity, I use the Louvain clusters from Figure 3.5 to define subpopulations in this analysis, rather than cell type labels found using PCPC [10].

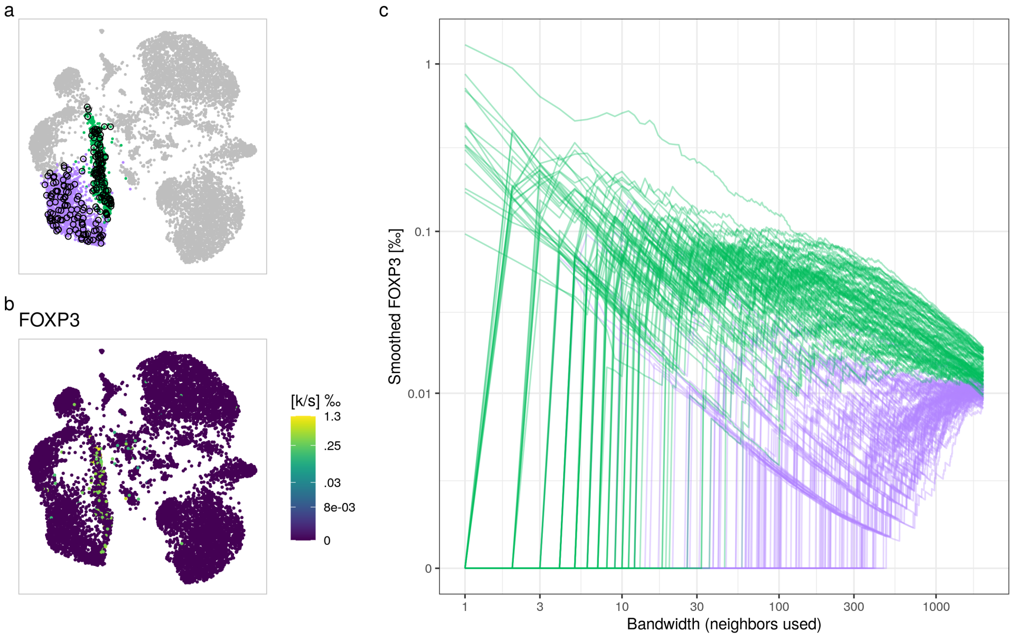


Figure 3.11: Smoothing bias and variance for FOXP3 expression. **a** 100 cells each were randomly selected (black circles) from Louvain clusters 5 (green, T_{reg} cells) and 9 (purple, naive helper cells). **b** Feature plot of FOXP3, marking cluster 5. Expression is shown as UMI counts k divided by total UMI s and multiplied with 1000 to show per mille (‰) for human readability. **c** The smoothed FOXP3 expression for the 200 cells selected in (a) is shown for increasing numbers of kNN. Each line represents one cell from (a), and the smoothed value was computed as pooled FOXP3 UMIs K divided by pooled total UMIs S (K and S as in section 3.2). Both axis are log10-transformed, and a pseudocount of 0.001 was added to the per mille (‰) values on the y-axis in order to avoid log10(0).

[10] I note that the TotalSeq antibodies in the MALT data included CD25 (IL2RA) marking T_{regs} (cluster 5), but unfortunately its signal was too weak (few UMI counts compared to many other antibodies) to define T_{reg} 'protein types', which would have been an even better ground truth than Louvain clusters.

Bias and variance Figure 3.11 illustrates the bias-variance concept for two subpopulations and a single marker gene. To make the following considerations rigorous, I picked the following example: Clusters 5 and 9 (blue and purple in Figure 3.11a and c) have highly correlated transcriptomes (c.f. heatmap in Figure 3.5, section 3.3.1). At the same time, cluster 5 is marked by a weakly expressed gene, the transcription factor FOXP3 (Figure 3.11b). For 100 random cells from clusters 5 and 9 (Figure 3.11a), the smoothed FOXP3 expression was computed with increasing bandwidth (Figure 3.11c). The variance in estimating FOXP3 expression becomes obvious from the width of the green 'ribbon', i.e. how widely the green lines are spread out vertically (Figure 3.11c). For few kNN (e.g. 10), the green ribbon is broad, while for 1000 kNN it is very narrow. Thus, increasing the bandwidth reduces variance in PCPC's FOXP3 expression estimates. At the same time, the bias increases. This is evident from the green and the purple ribbons, converging towards the same value when thousands of kNN are used. Naturally, the convergence value is the data set-wide average, i.e. when pooling counts across all 8412 cells in the MALT data. So on the one hand, it is important to pick the bandwidth large enough to obtain a non-zero expression estimate. For example, smoothed FOXP3 expression is 0 for many green cells in Figure 3.11c when using only 10 kNN, simply because FOXP3 was not detected in any of the 10 neighbors [11]. On the other hand, the bandwidth should not be too large, as otherwise the expression estimates become systematically wrong (biased). While bias and variance in smoothing seems important, the crucial question for PCPC is how the bandwidth impacts the classification outcome (FOXP3$^+$ cells).

Bandwidth and classification outcome To test the influence of the bandwidth parameter, I classified FOXP3$^+$ cells from the MALT data using different numbers of kNN with the same threshold $\tau = 0.01‰$. I find the overlap is in general high, and more neighbors find more positive cells (not shown in the figures). For example, when picking a small number such as 20 kNN, PCPC identifies 384 cells. Reassuringly, the majority of these (362, 94%) are also detected with 100 kNN, while at this high number of kNN 173 cells on top get labeled as FOXP3$^+$. Testing kNN values between 20 and 100 consistently gave large overlaps and always more positive cells with more neighbors. This is not unexpected: More neighbors decrease variance in Figure 3.11, and the larger neighborhood size decreases the area of uncertainty around the user-defined expression threshold (c.f. middle column in Figure 3.6). Still, 100 kNN would be too high for most data sets because larger bandwidths, while labeling more cells, are not *per se* better: It is a fundamental assumption of PCPC that a cell's kNN are, with the

[11] Averaged across all FOXP3$^+$ cells (cluster 5), FOXP3 expression is as low as 0.075‰, meaning that for a cell with 3000 total UMIs we expect a FOXP3 UMI count with only a 22.5% chance. Classification still works well because cluster 9 only has 0.003‰, i.e. 25-fold lower FOXP3 expression than cluster 5.

given data, indistinguishable from this cell. If it can not be told apart from highly expressing cells, we can call a FOXP3$^+$ cell beyond reasonable doubts, with the given data. I further discuss this noise-robust definition of positive cells in section 4.3. Thus, like with smoothing (c.f. section 1.6), choosing the bandwidth for classification also has a trade-off. More kNN will leave fewer cells undecided, but at the same time might more severely violate the assumption that positive cells have transcriptomes that are associated with FOXP3 expression. As a final note on the bandwidth, let us consider the task of separating green from purple cells in Figure 3.11c by eye. Evidently, this can be achieved with a wide range of bandwidth values, and 50 kNN (used throughout this thesis) may be a reasonable default value. Still, more principled approaches exist to find appropriate neighbors for our purposes, which I will further discuss in section 4.3.

3.5 Multi-sample multi-group comparison (lymphoma data)

PCPC is a flexible tool to annotate cell types with inference as goal. The ideal use case for which I have designed PCPC is differential testing in multi-sample multi-group multi-subpopulation studies (see section 1.5). As an example, I choose the lymphoma cohort recently published by Roider *et al.* [62]. In this section, I will use PCPC to classify a single cell type and perform differential gene expression testing. This approach can be scaled to annotate all subpopulations present in the data, as demonstrated for the MALT data (c.f. section 3.3.3).

Analysis goal As introduced in methods section 2.6, the 'lymphoma cohort' represents scRNAseq data sets prepared from the lymph nodes of 12 patients, 9 of which had B cell lymphomas. Some B-cell lymphomas in the lymph node are aggressive, while others are indolent and associated with a good prognosis. Specifically, follicular lymphomas (FL) are indolent (4 patients), while the cohort's aggressive tumors were diagnosed either as the transformed type (tFL, 2 patients) or as diffuse large B cell lymphomas (DLBCL, 2 patients). Understanding the differences between these two groups, aggressive and indolent, is a promising strategy to improve treatment. The first analysis step is to find genes that are differentially expressed between the two conditions.

Classifying CD3E$^+$CD8B$^+$ cells Here, I ask the question how tumor-infiltrating cytotoxic T cells differ between aggressive and indolent lymphomas. Using PCPC on each sample separately, I classify cytotoxic T cells as CD3E$^+$CD8B$^+$ cells. Figure 3.12 shows the marker expression (a,b) and the classification result (c). Patient DLBCL1 had no detectable T cells and was therefore excluded from this analysis. As above, I chose the marker expression thresholds to maximize agreement between the UMAP areas with CD3E$^+$CD8B$^+$ cells (black dots in Figure 3.12c) and co-expression of the two markers (a,b). This resulted in 183 cells per sample on average, with most cells in sample tFL1 (469 cells) and fewest in sample tFL2 (34 cells). Below, I will use these cell labels to test for differential expression between cytotoxic T cells from aggressive and indolent tumors. Before that, I use the classification example to illustrate the strengths of manual thresholding by the user, in preparation of discussing the limitations of automated classification (see section 4.2).

Informed decisions and flexibility with manual thresholding PCPC requires the user to manually select expression thresholds, which in this example are two thresholds per sample (one per gene) in order to classify cytotoxic T cells. While this is more manual labor that running other marker-based methods, such as Garnett [40], CellAssign [141] or Scina [142] (c.f. section 4.5), it provides more flexibility to meet the heterogeneity of patient cohorts. The importance of flexibility is illustrated in the following two examples.

While selecting thresholds for sample DLBCL2, I noticed the CD3E$^+$CD8B$^+$ cells are distributed to two separate groups in UMAP (Figure 3.12c, panel 'DLBCL2'). This situation requires the attention of an investigator, because deciding whether this classification result is desired or not might depend on the research question, and certainly requires more exploration. To this end, I inspected more markers for DLBCL2 and found that both groups of CD3E$^+$CD8B$^+$ cells were also positive for CD8A and negative for both CD4 and CD40LG (not shown), as expected from cytotoxic T cells. The 'upper' CD8B$^+$ cells furthermore highly express ICOS, GZMK, LAG3 and PDCD1, which are described activation markers in tumor-derived cytotoxic T cells [64]. The 'lower' group, in contrast, has a naive phenotype lacking these markers and instead highly expressing CCR7 and IL7R. Thus, exploration suggests that the two groups of CD3E$^+$CD8B$^+$ cells in DLBCL2 are not doublets or other artifacts, but represent two biological entities. For further analysis, an investigator may now decide what makes more sense for his analysis: Include all cytotoxic T cells, or only one of the two subsets? Here, I chose the former since I was interested in all cytotoxic T cells, and not all samples seem to have enough cytotoxic T cells to allow dissecting the activated and naive subpopulations in all samples. This example backs my argument put forward in this thesis that classification for inference is not right or wrong but instead depends on the question, in contrast to exploration. Put differently, if inference is the goal instead of exploration, cell types are discussed rather than discovered.

As a second example of how classification requires flexibility and informed decisions, I note sample FL3 also presented with two CD8B$^+$ subpopulations

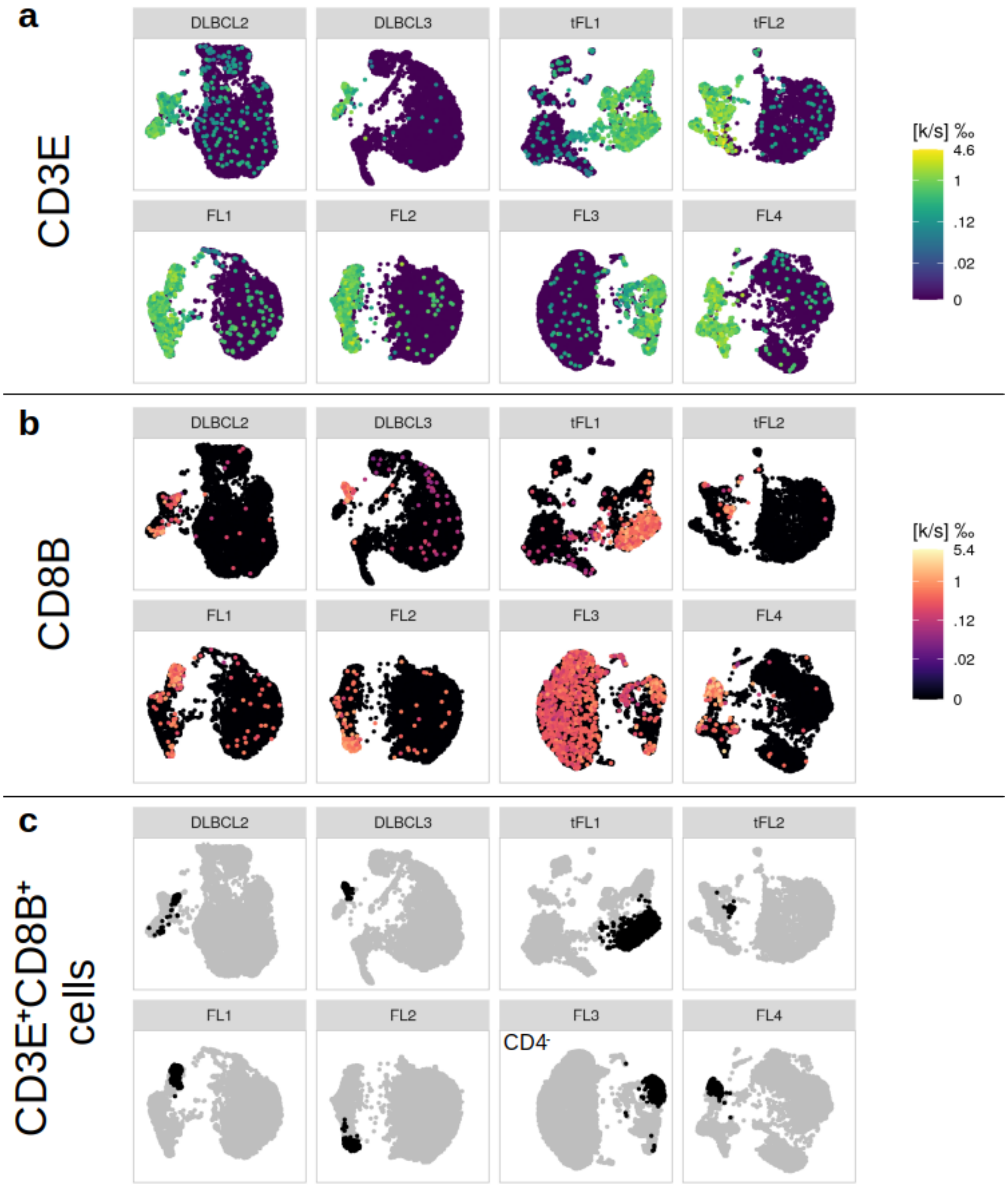


Figure 3.12: CD3E$^+$CD8B$^+$ cells classified with PCPC in the lymphoma cohort. Patients are indicated in grey boxes, feature plots were created as described in methods (section 2.2). For patient FL3 only, cells were additionally required to be CD4$^-$. **a, b** Marker gene feature plots, using different color scales for visual separation. **c** CD3E$^+$CD8B$^+$ cells (black dots) classified with PCPC.

(Figure 3.12b, panel 'FL3'). More exploration showed that the small 'lower' group in the embedding was also highly expressing CD4 (not shown), suggesting these cells represent doublets or a rare CD8B$^+$CD4$^+$ subtype. In contrast to DLBCL2, I decided to exclude this second population of CD3E$^+$ CD8B$^+$ cells by additionally requiring them to be CD4$^-$.

These two examples illustrate that automating marker-based cell type classification is difficult at least, and perhaps not even desirable. For a heterogeneous cancer cohort, classification is best done separately for each patient, taking literature knowledge and the analysis goal into account. Opting out on automation sets PCPC apart from all other marker-based classification tools [40, 141–143], deserving further discussion (see section 4.2).

Multi-group comparison: Aggressive and indolent tumors Having classified cytotoxic T cells as CD3E$^+$CD8B$^+$ in 8 patients (excluding DLBCL1 due to a lack of T cells), I next tested for differential expression (DE) between cells derived from aggressive and indolent tumors. Testing for DE in multi-sample multi-condition data, such as the lymphoma cohort, is still an area of active research [65]. Pseudobulk methods are simple, fast and perform well [65]. Therefore, I formed pseudobulks for the CD3E$^+$CD8B$^+$ cells, one for each patient, and tested for DE between aggressive and indolent tumors [12]. using DESeq2 [166]. As illustrated by the volcano plot in Figure 3.13, DESeq2 found 12 genes with higher expression in T cells from aggressive tumors and 2 genes with higher expression in indolent samples. This illustrates that even with as few as four patients per group, significant differences can be detected with scRNAseq cohorts.

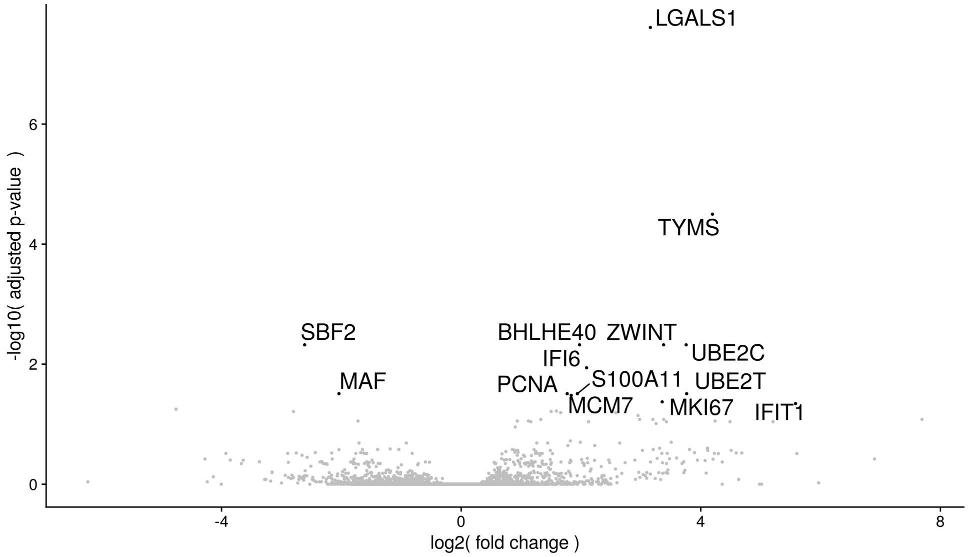


Figure 3.13: Volcano plot for testing differential expression in cytotoxic T cells from aggressive or indolent tumors. CD3E$^+$CD8B$^+$ cells were tested with DESeq2 on pseudobulks. Positive log2 fold changes represent higher expression in cells from aggressive tumors (DLBCL and tFL samples).

[12] I note that this type of multi-sample multi-group comparison has recently been referred to as 'differential state' testing, to discriminate it from comparing clusters within the same sample. I do not follow this convention, as explained in section 1.5

Pitfalls when interpreting inference results I present the DE genes in Figure 3.13 as an illustration of how PCPC can be applied to scRNAseq cohorts for multi-condition comparisons. Given the small number of patients and limitations in using pseudobulks, these results require further validation to avoid over-interpretation. Most prominently, while the genes reported here were DE according to DESeq2's statistical assumptions, there is no guarantee for generalization, i.e. that these genes would also be DE in a larger cohort. Only four patients per group might not be a valid sample from the entire population of all lymphoma samples. Thus, it might be that some of the DE genes obtained from DESeq2 are due to other characteristics than aggressive and indolent. For instance, tumor infiltration might have a certain kinetic, so that the T cell transcriptomes would depend not only on tumor aggressiveness, but also on the time point at which the lymph node was removed and the cells were sequenced [13]. Also, the lymphoma cohort was generated from frozen patient material, so more dead cells and thus contamination with ambiend mRNA are to be expected than for fresh samples. This is important to keep in mind for example when considering IFIT1, the gene with the largest fold change (Figure 3.13). While significant in the pseudobulks, the percentage of CD3E$^+$CD8B$^+$ cells with non-zero UMIs were below 3% in all samples except for DLBCL3 (21%). The inference results from small scRNA-seq cohorts should be considered hypothesis-generating events, followed by further validation experiments. The marker-based cell type definitions used by PCPC are ideal for follow-up experiments with flow cytometry or imaging with the same markers.

LGALS1 in cytotoxic T cells from aggressive lymphoma Above presented case study revealed DE distinguishing cytotoxic T cells in aggressive and indolent lymphoma. The gene with the lowest p-value was LGALS1, for which DESeq2 reported over-expression in the aggressive subgroup. Figure 3.14 shows feature plots of LGALS1 for each patient sample. As described in methods section 2.2, non-zero expression was plotted on top the more abundant zeros, and CD3E$^+$CD8B$^+$ cells are emphasized by making all other cells transparent. Inspecting Figure 3.14, it is evident that LGALS1 is expressed higher in aggressive (top row) than in indolent tumors (bottom row): The color intensity is, overall, higher in the cytotoxic T cells from aggressive tumors (top row). The percentage of CD3E$^+$CD8B$^+$ cells for which LGALS1 was detected was between 17 and 68% for aggressive and between 6 and 10% for indolent tumors [14]. Many non-T cells have high LGALS1 expression in sample DLBCL3 (Figure 3.14, panel 'DLBLCL3'), which could

[13] For resolving such transcriptomic changes in tumor-infiltrating T cells over time, a data set with 4 donors per group is obviously insufficient.

[14] This is no proof for over-expression, because the library sizes in cytotoxic T cells from the aggressive tumors were systematically higher (3098-4608 total UMIs on average) than for indolent tumors (1853-2916 total UMIs on average). I note that in contrast to the detection frequency, DESeq2's negative binomial model adjusts for these differences in library size, so LGALS1 is nevertheless a legitimate DE gene.

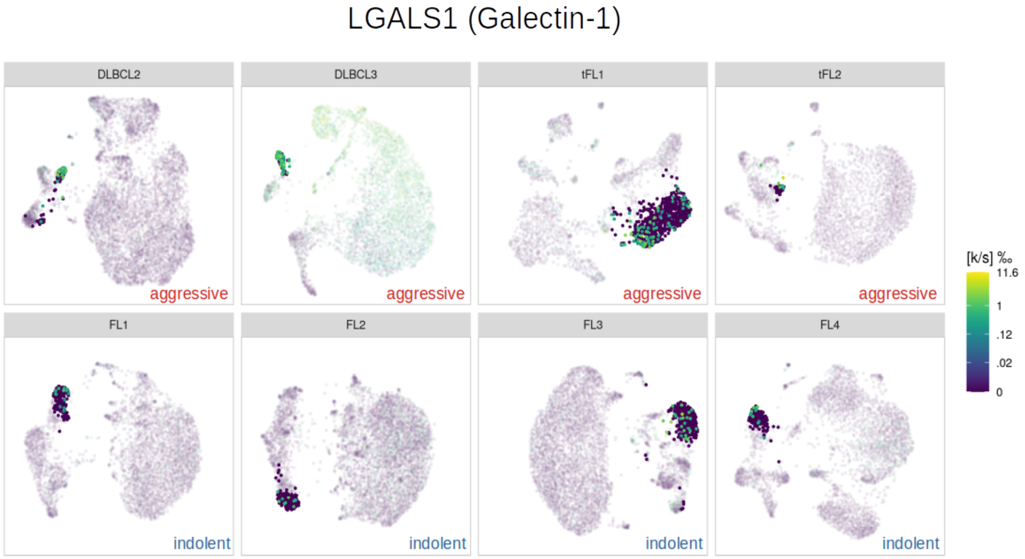


Figure 3.14: Feature plots of LGALS1 in CD3E$^+$CD8B$^+$ cells. Patient name is given within grey boxes, and tumor status (aggressive or indolent) as colored text. LGALS1 expression in these cells was higher for samples from aggressive tumors than for indolent, according to DESeq2 on pseudobulks. For clarity, only CD3E$^+$CD8B$^+$ cells are shown non-transparent, and nonzero values were plotted on top of cells with zero expression.

strongly contaminate a sparsely expressed gene with ambient RNA. In the case of LGALS1, however, high expression was observed in the other three aggressive tumor samples, suggesting this might be a biological property of aggressive lymphomas. LGALS1 (also known as Galectin-1) is known to be secreted by malignant tumor cells, suppressing cytotoxic T cell activity [167]. Of note, it induces apoptosis of activated human T cells [168], which explains why tumor cells would secrete it. Here, I now find that cytotoxic T cells themselves highly express LGALS1 in four samples from aggressive tumors, which is a noteworthy observation. An investigator could now use flow cytometry to sort CD3$^+$CD8$^+$ cells from more lymphoma patients to confirm these findings, and to further investigate if and how LGALS1 influences tumor progression in B-cell lymphoma patients.

In conclusion, PCPC annotates cell types in scRNAseq cohorts. Its flexibility is particularly apt at adjusting cell type definitions to individual patients as required by the inherent heterogeneity of cancer malignancies. The presented case study analyzing the role of cytotoxic T cells in a small lymphoma cohort was only one potential use case. In the future, similar questions might be addressed using larger cohorts (section 5.2 shows how to make this time-efficient for many patients).

Chapter 4

Discussion

In the previous chapter, I showed how Pooled Count Poisson Classification (PCPC) can be used to classify cells in scRNAseq data, giving examples from blood cells (CBMC data), fine T cell subsets (MALT data) and a multi-patient cancer cohort (lymphoma data). I have referenced the existing literature where appropriate in the previous chapter and thus already discussed many aspects of my work and how it relates to the existing state-of-the-art. Here, I focus on a few central aspects that I feel deserve a separate scientific treatment.

This discussion is organized from general to more specific. I start with the general concept of what cell types are, and discuss different classification requirements in exploration and inference (section 4.1). I then discuss the peculiarities of PCPC: Manual thresholding as opposed to automation (section 4.2), nearest neighbor pooling and its implications (section 4.3), using Poisson random variables to model pooled counts (section 4.4), and various limitations that fundamentally arise from how I designed PCPC (section 4.5). Finally I summarize limitations of this thesis, if distinct from those of PCPC as a method, in the next chapter (chapter 5) as an outlook on the next possible steps in this project.

4.1 The cell type concept

The cell type concept is not clearly defined and frequently debated [79]. In this work I argue its requirements depend on whether exploration or inference is the analysis goal. In other words, while scRNAseq has the ability to do both types of analysis, data-driven or hypothesis-driven, it should be separate, specialized algorithms performing these tasks. In this section, I discuss the differences between cell types for exploration (4.1.2) and for inference (4.1.3) in detail. I also provide guidance on how the interplay of exploration and inference can be organized (4.1.4). I start by noting how the cell type concept itself is somewhat subjective (4.1.1), which fits well with my observations from section 3.3.4 that unsupervised algorithms are not necessarily unbiased. For interesting impulses concerining the current cell type debate, I recommend a collection of opinions in 'CellSystem Voices' [81, comment] and the introduction of the muscat preprint [87, preprint] or paper [65].

This section in a nutshell The exploratory aspect of scRNAseq analysis has received much attention in the form of unsupervised algorithms, most prominently clustering after data integration [73, 122, 128–133] and label transfer [84, 126, 127, 136, 138] & [2, preprint] (introduced in section 1.7). The better the data set (multiple modalities, many cells, high transcript capture rates, etc.), the more detailed the cell types become that these methods can discover in high-dimensional space. For inference, on the other hand, the requirements are different. Novel insights gained into a subpopulation of cells (such as genes up-regulated after a certain treatment) should not stand on their own, but are instead to be interpreted in context with existing knowledge from previous studies. Most often, the requirement is thus to connect inference results to 'legacy knowledge' [1] obtained with non-sequencing technologies (e.g. imaging and mass cytometry, or bulk omics and functional assays from sorted cell populations). For this reason, cell type labels found with marker-based methods are ideal for inference. Out of the existing methods that find labels from known marker genes (introduced in section 1.7.4), PCPC is the only one that is simple, transferable as well as useful (as defined in section 3.1). For a given scRNAseq patient cohort, I therefore propose the following analysis strategy. Exploratory algorithms are used on this cohort or, better even, on a large multi-omic data set, to discover relevant cell types and their markers. PCPC or a similarly powerful marker-based approach is used to classify these cell types to the extent that the given cohort data set allows. The resulting insights from inference (such as differentially expressed genes or conclusions drawn from them) are then validated using the same marker genes in independent data sets or experimental platforms.

4.1.1 Subjectivity in cell type definitions

In this thesis, I argue that with the current technology, cell types are not objective fundamental truths to be discovered. Instead, they are the result of a subjective trade-off between how simple a cell type definition is, and how much complex experimental behavior it explains. Here, I further back this argument by reviewing the cell type concept across different disciplines.

That cell types are subjective is evident from the fact that separate biological disciplines disagree already on the first step: Which aspects should be consid-

[1] I ackowledge Wagner *et al.* [80] for coining the term 'legacy knowledge'.

ered and prioritized when defining cell types? For example, ontology has been proposed as the crucial factor to define myeloid cell types (such as dendritic cells, monocytes and macrophages) [85]; in the sense that cells should first and foremost be distinguished by which precursor and developmental lineage they have originated from. The authors note that fate-mapping experiments are required, which are not available in all organisms. In this ontology-based cell type concept, the cell's molecular cell state, such as surface markers or the transcriptome, would then be secondary and optional [85]. This is in contrast to the rather young discipline of scRNAseq, where the priorities are evidently exactly the other way around: It is the single cell's molecular state (transcriptome, proteome, methylation, etc.) that defines its cell type in the field of scRNAseq method development [40, 88, 89, 120, 141, 169], while lineage tracing is an optional read-out requiring specialized protocols [170, review]. Yet another approach to categorize cells is to focus on the evolutionary aspect, such that cell types from one species have counter-parts in related species [86]. Finding such homologous cell types, according to the authors, can be achieved by identifying the combination of transcription factors that define them (referred to as core regulatory complex). Summing up these observations, I conclude that the view on how cell types should be defined appears to depend on a researcher's subjective experimental preferences. Evolutionary biologists put the homology between species first, while researchers who employ single-cell assays prefer to define cell types based on molecular information, which to yet others is inferior to cell ontology defined by fate-mapping experiments.

Next to the lacking agreement on the rules for defining cell types, another rather subjective line to be drawn is the separation between cell type and cell state. In a recent opinion article [82], S. Morris notes that a the cell state (subject to fluctuations due to interactions with the cell's environment) and the cell type (permanent in absence of reprogramming and differentiation) should be kept separate. To this end, Morris proposes to ideally use multiple technologies in concert: Fate-mapping experiments to determine the lineage, scATACseq for the cell type's slowly-changing epigenetic signature, scRNAseq for dynamic transcriptomic cell states, *et cetera.* A similar line of argument, with a special emphasis on transcription factor combinations used by different cell types, is found in [83, opinion article].

Moving forward, I propose it is helpful to distinguish two types of classification. One for exploratory purposes, where ideally multi-omic protocols are used to discover fundamental cell types and cell states, if they exist. And the other for inference, where the limited signal-to-noise ratio is taken into account when defining cell types. I will discuss cell types for exploration and inference in the following two sections.

4.1.2 Cell types for exploration

As single-cell technologies become more powerful, a fundamental question of categorizing cells can be addressed: Does a single, optimal cell type assignment exist, or does the ideal classification depend on the scientific question? Put differently, when single cells are viewed in the high-dimensional feature space that spans the transcriptome, proteome, epigenome and metabolome, do they form disconnected groups (cell types), or rather a continuous spectrum? To discuss this former scenario, I will refer to these disconnected groups of cells as ‘fundamental cell types’ that unsupervised algorithms can discover and characterize. Multimodal analysis tools [110, 120] could be used to annotate a large multi-omic reference data set with these fundamental cell types. Given a new scRNAseq data set (the ‘query’ data), a strong case could then be made for finding a single, ideal annotation using data integration and/or label transfer set [73, 122, 138] based on the most relevant reference data set. In the latter scenario, by contrast, discrete latent cell types do not exist and any categorization of cells is a subjective trade-off between its simplicity and the proportion of experimental observations that it is able to explain (c.f. section 3.3.4). In this scenario, unsupervised methods are poor choices in inference studies, because they would appear unbiased when in reality the bias is just hidden in the methodology, i.e. algorithm parameters and preprocessing choices (c.f. section 3.3.4). Put plainly, this is the crucial open question: Are cell types discovered, or rather discussed? Do they represent fundamental ground truths, or are they mental concepts to explain biological complexity with simple rules?

For current scRNAseq protocols, I argue answering this fundamental question is not feasible yet. Limited to only the transcriptome (and perhaps one other modality [113, 118]), and with transcript capture rates of 30% or lower [15], the signal-to-noise ratio is not sufficient to make unsupervised methods truly unbiased (see section 3.3.4). Taken together with the personal bias discussed in section 4.1.1, I conclude that cell type definitions are subjective, intelligent choices by the researcher in most of PCPC’s use cases. I will describe these ‘inference-oriented’ cell types in the next section.

4.1.3 Cell types for inference

When stating my motivation to develop PCPC (section 3.1), I proposed that inference has different requirements for classification algorithms than exploration. Specifically, I proposed the three requirements for cell type assignments: Simple, transferable and useful. Here (4.1.3 and 4.1.3), I compare PCPC to the existing classification methods introduced in section 1.7 under these three aspects. Furthermore, I point out two other peculiarities of how PCPC finds cell type labels. First, the concept of positive cell types and how it avoids interpretations that are not permitted by the sparsity of scRNAseq data (4.1.3). And second, I discuss whether cell types should be mutually exclusive, as they are in most methods, or whether cell type overlap is acceptable, as with PCPC (4.1.3).

The concept of positive cells PCPC allows researchers to formulate precise, testable claims such as '*FOXP3$^+$ cells in MALT tumors have property XYZ*', by using the concept of positive and negative cells. To this end, I defined marker-positive cells as those with a transcriptome that is indistinguishable from cells with high marker expression. This definition circumvents the issue of smoothing artifacts [67] (introduced in section 1.6), because it is the high similarity to expressing cells, rather than the expression value itself, that is being inferred. In other words, PCPC does not claim that positive cells were necessarily highly-expressing cells themselves. I will discuss the concept of indistinguishable neighbors and its limitations in section 4.3. For now, I stress that my definition of positive cells acknowledges that 'better' data (with more cells/transcripts/modalities) might have led to a different set of cells labeled as positive. The analysis goal, after all, is not to improve the data (which is impossible), but to make claims from it that are beyond all reasonable doubts. Making such scientific claims under uncertainty is enable by my definition of positive and negative cells, because it appreciates the inevitable imperfection of any classification attempt in the presence of high technical noise. This pragmatic interpretation of the resulting cell labels clearly distinguishes PCPC from graph-based clustering (Seurat [37], Scanpy [38], etc.), where the definition of clusters is purely technical, namely groupings of cells that optimize the modularity score in the k nearest neighbors (kNN) graph. Thus, the concept of positive cells defined here introduces more intuition of what the cell types are and are not, which I find crucial for biological interpretation.

One interesting question is how PCPC is impacted when scRNAseq protocols continue to become more sensitive. If the transcript success rate (currently 5 - 30% [15]) improves, the technical noise decreases. Thus, on the one hand, fewer neighbors that are indistinguishable might be available to PCPC in a given data set, since the improved signal-to-noise ratio can measure more subtle cell-cell differences. On the other hand, pooling across fewer neighbors will be necessary to estimate marker gene expression with enough precision to separate positive from negative cells. I therefore speculate the definition of positive cells used by PCPC to remain relevant with 'better' scRNAseq technology.

Classification algorithms are either simple or useful As introduced in section 3.1, a classification method for inference should be simple, transferable and useful. Here, I discuss the two criteria 'simple' and 'useful' again briefly, and point out which ones are fulfilled by the existing classification tools introduced in section 1.7. I discuss 'transferable' in the next section.

Clusters are not defined in simple terms, instead they are a somewhat abstract result of optimizing the modularity score on a kNN graph (introduced in section 1.7.2). Thus, as I pointed out in section 3.3.4, clustering is a powerful exploratory tool, and therefore complementary to (not competing with) inference-oriented methods such as PCPC. In general, simplicity is an important criterion for cell type labels due to three reasons. First, they provide a clear language for communicating inference results and for scientific debates in general. Second, they facilitate falsification: Not only are marker-based cell type definitions easily testable across many experimental platforms, the derived inference results are also more actionable, i.e. tractable in validation experiments [2]. And third, they explain a complex world with simple rules, which I consider a fundamental goal of science: When two hypothesis can explain the same observations, it is wise to choose the simpler of the two (known as 'law of parsimony' or Occam's razor [171, Wikipedia]). Thus, assigning cell type labels from known marker genes is simple, while clustering and label transfer (introduced in sections 1.7.2 and 1.7.3) are not.

A cell type classification approach is useful when from a given scRNAseq data set, many cells can be retrieved with reasonable confidence. This is of obvious importance, for example because differential expression testing has more power when more cells are included. PCPC fulfills this definition of useful, for example because it was able to find most regulatory T cell (T_{reg}) cells in spite of FOXP3's low expression in the MALT data set (see Figure 3.6 in section 3.3.3). Other marker-based methods may be simple and transferable, namely Garnett, CellAssign, SCINA and scSorter. I argue, however, that they are not as useful as PCPC. For Garnett, I have shown in section 3.3.5 that it fails to label the majority of cells in the MALT data. I expect the same to be true for CellAssign and SCINA, because both tools only use the marker genes themselves, ignoring the rest of the transcriptome. This means that when searching for $CD3E^+CD4^+FOXP3^+$ cells, CellAssign and SCINA would indeed only use the information contained in three genes, which is problematic due to the sparsity of scRNAseq data. Using these methods with more markers, however, results in cell type definitions that are not transferable to some experimental platforms such as flow cytometry or imaging. I note that scSorter, like PCPC, uses the entire transcriptome next to the supplied marker genes (introduced in section 1.7.4). scSorter was published in the final stages of writing this thesis [143], so testing its 'usefulness' (ability to label many cells) on the MALT and lymphoma data is out of scope of this work. If scSorter is indeed able to perform well with few marker genes, the original publication did not show it: Between 3 and 12 markers were used per cell type with a few exceptions [143, supplementary material], for classifying major lineages for which PCPC requires only a single marker. Also, I speculate that scSorter might struggle on some data sets due to its assumptions: scSorter fits only two expression levels across all cells (background expression or over-expression). This means it assumes expression is

[2] Compare '$FOXP3^+$ cells express IL2RA.' with 'Cluster 5 expresses IL2RA.'.

either off (background) or on (over-expression), but not intermediate. I point out violations of this assumption in sections 3.3.5 and 4.2, for example CD4 expression in the MALT data.

Transfer of marker panels The third criterion for inference-oriented cell type annotation is that it should be transferable (c.f. section 3.1). This is relevant embed novel inference results on a certain subpopulation into the existing literature ('legacy knowledge' [80]) and follow-up studies. While feasible with most classification methods, there are fundamental differences in how the transfer is achieved, which I now discuss.

Label transfer between scRNAseq data has received much attention recently (introduced in section 1.7.3). One of the first tools for this, Garnett, uses logistic regression to 'learn' a linear combination of genes that is able to separate cell types in the reference data, and transfers these weights to annotate the query data. Most recently, a similar strategy has been proposed for the deep learning tool scArches [138]. Specifically, scArches can learn neural network weights on a large reference data set. Representing prior knowledge, these weights are then 'fine tuned' on the new data set, finally annotating this 'query' data. The strategy that marker-based methods such as PCPC pursue is fundamentally different. Rather than collecting trained classifiers in the form of weights obtained from a reference, I suggest to curate marker gene lists instead, and use those on new data sets. The two approaches have opposite strengths and weaknesses: Applying it to a new data set can be expected to be less work (either manual or computational) a trained classifier than for a curated marker, but trained weights might fail when the query data is very different from the reference data. Put differently, curating marker genes, in comparison to learned weights, is qualitative rather than quantitative, and I expect it therefore to generalize better to unseen data (different organs, diseases, etc.).

Marker-based cell type definitions have the advantage that, in contrast to clustering and label transfer, they can also be applied in non-sequencing platforms. For example, CD3E$^+$CD4$^+$FOXP3$^+$CCR7$^+$ cells can be further studied with sequencing, flow cytometry or imaging. Before using the same cell type definition across these technologies, however, a researcher first has to ensure that messenger RNA (mRNA) and protein levels are sufficiently correlated. I note that for well established markers, this correlation has often times already been proven (compare for example CD3 protein and mRNA levels in CITEseq data sets). In general, protein-mRNA correlation is low for high abundant house-keeping genes (ribosomes, mRNA splicing), and high for dynamic responses (nucleotide metabolism, acute inflammatory responses) [172]. These dynamic response genes are perfectly suited to define cell type subsets in experiments where cells respond to a treatment condition. Another study found good correlation for most genes in ovarian cancer [172], suggesting the transfer between scRNAseq cohort studies of complex diseases and protein-based technologies (flow cytometry, mass spectrometry, etc.) is realistic.

Cell type overlap and 'cherry picking' Existing classification algorithms (clustering, CellAssign, Garnett, to name a few) are designed to assign exactly one cell type label per cell. In other words, cell types are defined by these tools to be mutually exclusive. Since cell type labels obtained with PCPC are not necessarily mutually exclusive, I discuss the implications of overlapping cell types in this section. Of note, with 'overlap' I mean multiple labels from cell types that are not subset or superset of the other: Naturally, the same cell can at the same time belong to the cell types 'T', 'CD4-positive T' and 'T_{reg}' – cell type overlap instead is when a cell is 'T_{reg}' and 'B' at the same time.

With PCPC, this is possible that a cell is assigned two cell types at once, for example CD3E$^+$FOXP3$^+$CCR7$^+$ and CD3E$^+$FOXP3$^+$ICOS$^+$. At first glance, this overlap may appear undesirable or even nonsensical from a biological perspective. This is because for exploration, cell type overlap would indeed be an obstacle in the way of finding fundamental cell types (fundamental in the sense of section 4.1.2). For inference, in distinction to exploration, mutually exclusive cell types are not *per se* strictly required. For example, one could be interested in asking scientific questions about FOXP3$^+$ICOS$^+$ T cells in a publicly available data set. In such cases, it might be unimportant whether some of these cells overlap with FOXP3$^+$CCR7$^+$ T cells or not. Perhaps the researcher does not even annotate FOXP3$^+$CCR7$^+$ T cells, but instead extracts only the FOXP3$^+$ICOS$^+$ T cells she or he is interested in. This 'cherry picking' of cell types is an ideal use case for PCPC, and particularly appropriate if the scRNAseq data is only one piece of evidence amongst many.

Perhaps PCPC can thus help to increase the re-use of published data sets for novel scientific questions, due to its flexibility to extract any group of cells identified by a unique combination of specific markers. As pointed out in the muscat paper [65], a subpopulation is any group of cell for which it is interesting to ask inference questions. As scRNAseq matures as technology, it will take its place amongst other experimental methods, contributing one more piece of evidence to make a convincing scientific case.

In general, I expect higher degree of overlap at finer hierarchies than at coarser ones. For example, T cells will be distinct from B cells in most if not all data sets, while detailed subsets such as FOXP3$^+$CCR7$^+$ and FOXP3$^+$ ICOS$^+$ T cells are transcriptionally more similar and might not separate in all cases. Evident from UMAP as overlapping CCR7 and ICOS expression, the reasons behind non-exclusivity could either be technical (insufficient signal-to-noise ratio) or biological (double-positive cells exist). Thus, curating marker panels that result in mutually exclusive cell types requires work, but is possible as I have shown for the MALT data (see section 3.3.3). In particular, generalizing a marker panel from one biological setting to another (tissue/disease/patient) may require compromise on top of extensive manual work. Still, it is not without merit to constantly revise cell type definitions, much like all biological hypothesis are constantly revised, falsified and refined.

4.1.4 Cell types in practice

My insights on the cell type concept can be reformulated as best practices on how to find cell types for a given biological setting (disease, organ, species). In particular, I advocate iterating these four steps to derive insights from scRNAseq data:

1. The research community creates a large reference data set, ideally using multi-omics protocols or including lineage tracing. Unsupervised methods can then be used to derive cell type / cell state definitions. These are 'optimal' in the sense that no data set with greater signal-to-noise ratio exists, and so these cell type definitions are expected to be as fundamental (as defined in section 4.1.2) as possible.

2. Marker genes are found that define these cell types. If no mRNA markers exist, scRNAseq might be the wrong technology for the system under study. Instead, other technologies such as imaging mass cytometry or multimodal sequencing protocols (CITEseq [113], scM&Tseq [173], etc.) may be used.

3. The researcher classifies cells in her/his data set using marker-based cell type definitions (e.g. using PCPC), then uses them for inference (hypothesis testing, e.g. molecular pathway activity or differential gene expression testing between conditions).

4. Critical inference results and conclusions derived from them are validated on independent data. This includes cell type definitions themselves: are they useful in other organs / diseases / patients / species?

According to this strategy, PCPC is complementary to (not competing with) unsupervised methods, such as data integration, clustering and label transfer. While these tools provide powerful exploration, PCPC adds falsifiability (step 4.). This is crucial: Only clearly defined cell type definitions are can be proven wrong by showing they do not work well in a certain tissue, disease or species. Only then, a scientific debate can ensue to decide whether this represents an exception to an otherwise helpful cell type rule, or if better marker genes can be found. Likewise, any results derived from the cell type definitions can also be falsified in a clear way, which could improve reproducibility of biological findings.

4.2 Automation and challenges

I have named a few concrete examples of marker genes that make automation difficult in section 3.3.5. Briefly, TTN requires a relative, not absolute expression threshold; and CD4$^+$ cells have bimodal CD4 expression that is not relevant to the classification task. Also, section 3.5 illustrates that annotating heterogeneous cancer cohorts may require human decision making, since different subpopulations may be present in different tumors. Here, I discuss more aspects of automation to consider, and further explain why I adopted manual thresholding instead. The personal bias that manual thresholding might introduce into the analysis is discussed in section 4.5, and interactive code to make it swift and simple is shown in the outlook (section 5.2).

Of note, Köhler *et al.* recently demonstrated that complex algorithms such as deep learning do currently not outperform other machine learning approaches in cell type classification [108]. The authors concluded that since cell type annotation is a simple and linear rule-based approach, non-linear algorithms such as deep learning are unable to use their advantages for superior performance [108]. I would go even further and argue that cell type definitions are, outside the exploratory setting (c.f. section 3.1), subjective rules (see section 4.1.1). Any conceivable algorithm, objective by design, will struggle to provide satisfactory results in matching subjective criteria.

Challenges in automating marker-based classification Next to unexpected marker gene behaviour (TTN and CD4 in section 3.3.5), a further obstacle is that marker expression can be an artifact from the cell capturing procedure. For example, HBB may be detected in all cells due to erythrocyte contamination [174, Fig. 4 therein]. Also, I foresee marker-based cell type definitions to require many 'exceptions' in order to meet biological complexity, and these are best handled by human users instead of automated algorithms. For example, PDCD1 (encoding PD1, an immune checkpoint molecule) marks exhausted (inactive) T cells [175], but in certain cases can also be indicative of T cell activity [64, Fig. 5f therein].

Before I decided to use manual thresholding, I have tested ways to automate marker-based classification as well (not included in this thesis). Briefly, I have tried three approaches: Logistic regression much like Garnett, Naive Bayes on the raw UMI counts and Gamma mixture models fitted to smoothed marker gene expression using Expectation-Maximization. All three approaches require a training set, i.e. a few cells from the data set that are annotated correctly with cell type labels [3]. I have not succeeded at automating this labeling of training cells, and believe this is because an algorithm, unlike a human user, can not make use of prior knowledge from scientific training,

[3] The Gamma mixtures do not require this in theory, but I have found it useful in practice to map the provided cell type classes to the resulting groupings of cells.

the literature and exploratory data analysis. Published algorithms evidently struggle with this as well: Garnett's heuristic to label training cells was misled multiple times in the same data set (as described in sections 3.3.5 and 3.3.5). Briefly, one 'problem' was that TTN is expressed not only by TTN$^+$ cells but also B cells, and another that CD4 showed bimodal expression among the CD4$^+$ cells that was not relevant to the classification task. The 'rule-of-thumb' implemented by Garnett failed to correctly label training cells for these classes, resulting in very poor classification results (section 3.3.5). Perhaps it is still possible to conceive an algorithm that can label training cells robustly. My conclusion at this time is, however, that finding marker gene thresholds is an intelligent task, that does not only depend on the data set at hand, but instead has to integrate prior knowledge from the literature.

It would be interesting to test the other marker-based approaches (CellAssign, SCINA and scSorter) on the MALT and lymphoma data (see outlook in section 5). This could show how much automated algorithms are able to achieve on complex biological samples with the current state-of-the-art, or it could further back my argument that manual labor is well invested here. Towards answering this question, I note the examples provided by the authors in all three papers [141–143] separate clearly defined, major lineages using multiple markers, while PCPC achieves the same task with a single marker per lineage (see section 3.3.3). I discuss more differences between these methods and PCPC in section 4.1.3.

Automation is less flexible than manual thresholding In general, the price of automation is to make stronger assumptions about the data. For example, scSorter assumes that markers are bimodal, i.e. the same two levels (background and overexpression) are used for the entire data set. While CellAssign allows different expression levels for each class, it still models each class as one single negative binomial distribution - this implicitly assumes that for example non-T cells can not have bimodal expression in any T cell marker. Also, all approaches need to solve the 'outgroup' problem, i.e. accommodate the possibility that a group of cells best remains unassigned. This is required, for example, when more cell types are present in the data than defined by the user, and requires assumptions about how this 'outgroup' might look or not look like. I propose that more assumptions make methods more prone to malfunction in unconventional cases, but more research is required to decide whether this is crucial for classification in scRNAseq cohorts.

Finally, automated algorithms such as CellAssign, SCINA, scSorter and Garnett make it somewhat tedious to adapt the marker selection interactively. This is because they all follow the same divide-and-conquer scheme: In a first step, a list of cell types and their marker genes is curated manually by the user. In a second step, the algorithm is provided with this marker list and the raw UMI counts of a data set, and then runs completely unsupervised from there. In contrast, PCPC interactively combines these two steps. Replacing one T cell marker with another, for example, is a matter of seconds with PCPC [4] and the resulting positive cells can immediately be inspected in UMAP (see interactive code example in outlook, section 5.2). In practice, this is valuable when annotating heterogeneous samples, such as different human patients. A researcher may have certain marker genes in mind, but when picking thresholds with PCPC may realize they do not fit a particular sample and thus explore alternative markers. Following this strategy, it might turn out for example that classifying cancer cells in one patient requires different markers than in another patient. Thus, PCPC is more flexible to meet the requirements of multi-sample multi-condition studies, such as heterogeneous cancer cohorts. The manual labor and subjective bias that might arise from this strategy is discussed in section 4.5.

In conclusion, automating marker-based classification is challenging and perhaps not even desirable. If the analysis has inference as main goal (and not exploration, see sections 3.1 and 4.1), then cell types are a rule-based concept, and these rules are made by human researchers. I argue that in this setting, coming up with good cell type definitions has to be an iterative process of exploratory analysis, scientific debate and self-correction. It is intelligent work to be done by researchers, and it is unlikely that algorithms can take this task over completely.

4.3 On the accuracy of nearest neighbor information

PCPC pools marker gene counts from a cell's kNN to assign a cell type label. Throughout this thesis, PCPC uses 50 kNN found by Euclidean distance on PCA embeddings, but in principle all unweighted kNN graphs from any method can readily be used. Here, I discuss how different kNN graphs might affect classification with PCPC. Most importantly, I introduce methods that can identify indistinguishable nearest neighbors as proposed in my definition of positive cells. Furthermore, I discuss the difference between directed and undirected kNN graphs, and how SNN and other weighted graphs can be adapted for PCPC. Finally, I conclude finding the neighbors for pooling with these strategies can result in different numbers of neighbors per cell, which I consider desirable.

As a general note, kNN information is relied on by virtually all scRNAseq

[4] kNN pooling on a single computational thread (without parallelization) computes within milliseconds on a conventional laptop, and plotting the result takes 4 seconds. Using all markers to annotate the MALT data set computes only a few seconds, while Garnett runs roughly 30 minutes.

workflows in one way or another, for example by clustering (Seurat [37], Scanpy [38], Garnett [40]), data integration (Conos [73]) and visualization (UMAP [47], tSNE [48]). As for these methods, I expect 'better' kNN information to improve results obtained with PCPC. In general, I recommend to use algorithms that identify indistinguishable neighbors. I expect this will make cell type labels and the resulting inference results more robust, i.e. more reproducible with 'better' data (more cells / sequencing depth / protocol sensitivity).

Indistinguishable neighbors I now discuss more principled approaches to find kNN. Of note, two recent algorithms have made suggestions how kNN can be selected such that they are indistinguishable from one another, given the data's signal-to-noise ratio. In particular, VarID [146] tests expression values across all kNN and all genes for their likelihood of being 'sampled' from the same negative binomial distributions (one per gene). Only neighbors where this is likely across the entirety of all genes are retained, creating a 'pruned' kNN graph [146]. I note that MetaCell follows a very similar rationale, aiming at identifying cells that "ideally represent re-sampling from the same cellular state" [75]. While these methods were still under development, I have started exploring my own approach to find indistinguishable kNN, based on the simulation of 'Poisson replicates' for each cell (not shown). Briefly, I smoothed the entire transcriptome for a given cell, and sampled many cells from this expression vector using the Poisson distribution. Initial experiments (not shown) indicate that these Poisson replicates tend to have similarly high Euclidean distances to each other than to the closest cells in the data set. Thus, my idea is that the cells that lie as close as a cell's Poisson replicates are indistinguishable from this cell. While this resulted in reasonable numbers of neighbors for most cells in the MALT data and a few data sets I tested (not shown), there are also cell types where no neighbors at all are selected this way (namely the microglia cells in the multiple sclerosis data described in [60], not shown). More research in this direction it thus required to ensure the positive cells found with PCPC have a clear biological interpretation as stated in section 4.1.3.

Directed and undirected neighbor edges PCPC conceptually uses directed neighbor graphs when pooling UMI counts, which I now briefly discuss and compare. Directed edges in a kNN graph mean that if a cell is amongst the kNN of another cell, the inverse is not necessarily true. While simple Louvain clustering does require symmetric kNN graphs (I will return to clustering in the next section), I argue the asymmetric neighbor relationships used by PCPC are desirable. For example, imagine 50 cells form a dense community in expression space, and a 51st cell is drastically different, constituting a lonely outlier. Then, this outlier would not be among the 50 kNN of any other cell, which would avoid potential biases when pooling counts within neighborhoods. This is an advantage of asymmetric kNN graphs as used by PCPC. To stay with this example, one weakness of simple kNN approaches is that this 51st cell would ideally have zero kNN, given its outlier nature, but instead will of course be assigned the same number of neighbors as all other cells. Thus, PCPC is ideally used with kNN found with a principled approach (such as pruned kNN, see previous heading), to ensure existing outliers are not misclassified. To give a specific example for outlier cells encountered in this thesis, I name the doublets and multiplets in the CBMC data, which PCPC was able to discriminate from actual T cells (c.f. Figure 3.2 in section 3.2.3). While PCPC does not specifically safe-guard against outlier cells, it is conceivable that the kNN of such outliers are a noisy, random collection of different cell types, so any lineage marker gene is 'diluted' and the cells remain unassigned.

If simple kNN are used instead of indistinguishable kNN, then the user is required to choose a bandwidth value (the number of nearest neighbors). While Figure 3.11c suggests a wide range of bandwidths is acceptable, a more principled approach is to use cross-validation. Of note, 'molecular cross-validation' has recently proposed as means to find parameter values for scRNAseq gene smoothing [176, preprint]. Still, the most promising approach that I would pursue in future research endeavors is the identification of indistinguishable kNN (see previous heading).

Shared nearest neighbors (SNN) While graph-based clustering relies on nearest neighbors like PCPC does, modern implementations further refine kNN graphs with the shared-nearest neighbors (SNN) principle (introduced in section 1.7.2). SNN are based on the Jaccard index (overlap) between the kNN of two cells. This might prevent random ('noisy') edges from impacting count pooling as much. While I leave a thorough testing to future research, I have run initial tests on how to use SNN information when pooling counts with PCPC. I note that SNN graphs are weighted (continuous values between 0 and 1, instead of binary), and so the pooling loses intuition - after all, what does it mean to pool 25% of one cell together with 14% of another? Also, instead of pooling 50 kNN, the number of cells we pool is not clear. Summing all the edge weights of a cell's SNN gave numbers much smaller than 50 when I tested it on the MALT data, but still information from many more cells was used, so how do we adjust it to a more reasonable number? In initial experiments, it looked promising to convert the weighted SNN edge weights to binary format (1 if Jaccard index is larger than 0.1, 0 otherwise), but more research is necessary before SNN can confidently be applied in PCPC.

Different numbers of kNN per cell There is another appeal to using kNN graphs computed with VarID, MetaCell, Poisson replicates or SNN (see previous headings). Namely, it would allow every cell to be assigned a different number of neighbors, depending on how well the cell is connected.

Assuming the technical noise is constant for all cells (same transcript conversion rate during library preparation), I would expect cells from homogeneous cell types to have many neighbors then, while outliers might receive none at all. Classification with PCPC would take this into account, assigning cells with more information more confidently, while leaving outlier cells unassigned (PCPC assigns cells more confidently when the pooled total UMIs S are larger, see section 3.2.4). While I use a constant number of kNN throughout this thesis, these more principled kNN search methods should thus be explored in future research efforts.

4.4 PCPC's model assumptions

PCPC pools counts from a cell's kNN. It compares the pooled counts to a user-specified expression threshold, using the tails of a Poisson distribution in order to find positive and negative cells. Here, I discuss the Poisson distribution as a model for UMI counts, with a detailed discussion of the literature on the topic.

Poisson distribution to model UMI counts from a single cell In section 3.2.2, I have described a model for how UMI counts are being generated in scRNAseq protocols, and claimed that the UMI counts from a single cell are well described by a Poisson distribution. I discuss this assumption here, but note this is an academic exercise. The assumptions on pooled counts are more relevant to PCPC, and these are discussed under the next heading.

The generative model for UMI counts, as stated in 3.2.2, assumes the count of one gene in a single cell is well described by a Poisson random variable. That is, the technical noise is modeled using a Poisson distribution, while biological variation may introduce over-dispersion and thus requires a negative binomial model (as introduced in section 1.3). This is an important distinction to make: The counts from a single cell can be modeled as a Poisson random variable, while modeling the counts across multiple cells would make a negative binomial random variable more appropriate, as the non-zero dispersion allows the variance to be larger than the mean (while for Poisson, the variance equals the mean). Thus, it is important to realize that the model described in section 3.2.2 uses Poisson random variables to describe the distribution that would be obtained from a hypothetical re-sequencing of the very same cell, i.e. in absence of all biological variation.

A potential limitation in modeling a single cell's UMI counts as Poisson random variable is that highly expressed genes have been reported to be overdispersed even in absence of biological variation [41, 177] & [33, preprint]. Specifically, Grün *et al.* used an early UMI protocol to sequence a homogeneous mRNA solution instead of cells, thereby removing all biological variation. The resulting UMI counts showed over-dispersion for the genes with highest expression [177, Fig. 1c therein]. The same over-dispersion for highly expressed genes is described by the authors of scTransform, which for this reason uses regularized negative binomial (instead of Poisson) regression for normalization [41, Fig. 4D therein]. Furthermore, my own experiments with similar mRNA data showed over-dispersion of the genes with highest expression (not shown [5]), while the study generating this data concludes the Poisson model describes the data well [32]. Townes *et al.* also conclude the Poisson model is sufficient [1], but I note these two last studies ([1, 32]) focused on the fraction of zeros in, rather than the variance of, normalized UMI counts.

If highly expressed genes indeed have over-dispersed technical noise, then it is an interesting question to ask why. It has been speculated the reasons are differences in capture efficiency between droplets [33, preprint] or tubes [177]. Indeed, Grün *et al.* estimated the tube-to-tube variability in capturing efficiency by adding synthetic spike-in molecules of known concentrations before applying their UMI protocol, and found it explains the observed over-dispersion well. Also, this over-dispersion of highly expressed genes is not apparent after capture efficiency has been corrected for by downsampling all cells/droplets to the number of UMIs [1, Fig. 1c/d therein]. I note that beyond these speculations, there might be more mechanisms at play that are less intuitive due to the complexity of the sequencing process. For example, the base composition of UMI barcodes affects the likelihood to observe them, leading to skewed UMI distributions with counter-intuitive properties [20].

In conclusion, modeling the UMI counts of a single gene and from a single cell as Poisson random variable is debated only for highly expressed genes, while it is widely accepted for the vast majority of all genes. In order to model technical noise for these ‘house-keeping’ genes, a negative binomial distribution is thus more appropriate than the Poisson distribution. For non-UMI data, the model on top has to account for zero-inflation as introduced in section 1.3.

Poisson distribution to model pooled counts Under the previous heading, I have noted that for highly expressed genes, counts from individual cells are over-dispersed and require a negative binomial model. Importantly, this does not apply to pooled counts, according to PCPC’s logic, irrespective of their expression level. Instead, UMIs pooled across a cell’s kNN are

[5] Briefly, I obtained UMI counts from soluble mRNA from Valentine Svensson upon personal request. I divided raw UMI counts by the total UMI for all genes and cells and computed gene variances and means of these normalized values across all droplets (containing mRNA instead of cells). A scatter plot showing the variance-to-mean ratio (VMR) and the mean (both axis logarithmic) shows a constant VMR (consistent with Poisson) for all genes except the 20 or so with highest expression. This investigation is unfinished, which is why I consider it anectdotal evidence that belongs into a footnote rather than the main text.

modeled as Poisson random variable, which I discuss in the following. For simplicity, I will discuss how PCPC finds positive cells, but the same considerations apply accordingly to negative cells.

Conceptually, finding positive cells with PCPC asks whether a hypothetical re-sequencing of the cell's 50 kNN could have given pooled counts as low as expected from a Poisson distribution around the user-defined threshold (c.f. section 3.2.4). If the negative binomial distribution were used, the expression strengths of the kNN would be treated as gamma random variable. By using the Poisson distribution, PCPC instead considers these expression strengths as fixed [6]. Consider hypothetical re-sequencing experiments, where the exact same cells (i.e. the kNN) with the same amount of mRNA could again be profiled with the same scRNAseq protocol. Then, 'fixed' means that the kNN's expression strengths would have the exact same k unknown values, and the UMI counts observed in each re-sequencing experiment would correspond to k Poisson random variables (according to the model specified in section 3.2.2, for discussion see previous heading). Then, pooling UMI counts across a cell's kNN corresponds to summing up Poisson random variables. The sum of Poisson random variables, even with different Poisson rates, again is a Poisson random variable, which is why PCPC's logic does not require a negative binomial model.

Of note, using the Poisson model does not address biological variation between kNN at all, because PCPC includes this variation in its definition of positive cells. As detailed in section 4.1.3, this definition puts it on the researcher to correctly interpret what positive cells are (namely, cells with highly-expressing neighbors) and, importantly, what they are not (cells that can be guaranteed to have had high mRNA content themselves). When using PCPC, the crucial question is whether the neighbors of a given cell are relevant in deciding on that cell's cell type label, and this will depend on the cell type granularity. In most data sets, the kNN will be appropriate to separate major cell types (B cells from T cells), and many subtle T cell subsets (see e.g. MALT data in section 3.3.3). With PCPC, it thus is the researcher who decides which granularity of cell types can still be resolved with the given data, not the statistical model.

One limitation of PCPC's assumptions becomes apparent when we restate the above as follows: PCPC models the technical noise with the Poisson distribution, while the researcher is charged with the biological uncertainty when interpreting what positive cells are [7]. The problem is that the technical noise

[6] Note that the negative binomial distribution, also called gamma-Poisson distribution, is a hierarchical model in which gamma random variables are used as rate parameters in Poisson random variables, conceptually. The width of this gamma distribution corresponds to the over-dispersion, and the Poisson distribution is its special case when over-dispersion is zero.

[7] For example, in order to interpret FOXP3$^+$ cells as T$_{\mathrm{reg}}$ cells, a researcher has to answer questions such as: How likely is it that a T$_{\mathrm{reg}}$ cell was assigned a B cell as neighbor? How about a CD8 T cell, or an even more related CD4 T cell subset?

is not always appropriately modeled by the Poisson distribution. As pointed out in the previous paragraph, there is a non-negligible over-dispersion for highly-expressed genes (often refered to as 'house-keeping genes'). Using the negative binomial distribution instead of Poisson would solve this issue, but at the same time require dispersion estimates for each neighborhood [8]. Throughout this thesis, I therefore chose to use the Poisson distribution instead, but note this is a limitation of my work. It is a task for future research to understand why the technical noise of highly expressed genes is overdispersed, and to test whether this means that PCPC is over-confident in assigning positive cells for house-keeping genes [9].

In conclusion, PCPC assumes that the Poisson model suffices to describe pooled counts, by moving considerations of biological variation into the definition of positive cells. This separation of statistical model and the biological variation between neighbors may improve interpretation of the cell types found in scRNAseq data, simultaneously empowering and obliging the researcher (subjective bias is discussed in section 4.5). Given that the kNN are found with rigorous analysis methods (c.f. section 4.3), the FOXP3$^+$ cells for example can be interpreted as cells that are (with the given data) indistinguishable from FOXP3-expressing cells. This acknowledges that better data (more cells, more omic modalities or higher protocol efficiency) may always revise the current knowledge on cell type.

Similarity and differences to scTransform I want to acknowledge scTransform [41] by noting that the Poisson tails used by PCPC are conceptually similar to the Pearson residuals computed by scTransform. The goals are of course very different: PCPC models pooled counts with the goal of classification, while scTransform models individual counts for gene expression normalization. Also, scTransform uses the regularized negative binomial distribution, while PCPC uses the Poisson distribution for reasons pointed out in section 4.4. Thus, both methods are very dissimilar, but it was scTransform's Pearson residuals that made me realize the Poisson tails can be used to define positive and negative cells.

4.5 Limitations

I next discuss limitations of PCPC in general. For limitations of this thesis, see the outlook in chapter 5.

[8] A neighborhood is made up of a cell's kNN, including the cell itself.

[9] This case has not occured yet in my work, because these highly expressed genes (typically MALAT1, ribosomal proteins, etc.) are not marker genes for any cell type I have worked with. This does, however, not mean that case does not exist.

UMI protocols only Our method can only be applied to scRNAseq data generated with UMI protocols. Pooling counts from a cell's kNN results in a Poisson random variable for UMI data (c.f. 4.4), while it is statistically less tractable for non-UMI data due to the amplification bias of polymerase chain reaction (introduced in section 1.2). I observe that 10x technology has spread widely within the research community, and so I assume that a large proportion of scRNAseq patient cohorts that are being generated right now will generate UMI data. Also, I point out how SAVER's smoothing estimates could be used to extent PCPC to non-UMI data in the future, which could also make it orthogonal to UMAP (see section 4.5). As of today, however, requiring UMI data is a limitation that might prevent some researchers from using PCPC.

PCPC requires suitable marker genes Grabski et al. recently showed that the majority of all genes show bimodal expression across 218 cell types in PanglaoDB [2, preprint, Fig. 3 therein]. In other words, most genes are either 'on' or 'off' in a given cell type, suggesting that specific markers exist in the vast majority of cases. Still, it is conceivable that a cluster does not have any good markers, even if it is evident as distinct population in UMAP. It could, for example, be defined by a combination of dozens of genes with a weak tendencies towards over- or under-expression in that particular cell type, but without one single clear gene. In such cases, PCPC might struggle to select positive cells for this cell type. To my knowledge, this is a hypothetical scenario and inference results for such a cell type would anyways be of limited for use since it would be inaccessible to many experimental platforms such as flow cytometry and imaging (c.f. 'actionable' in section 3.1 and [145, blog post]).

It is a requirement that the user has curated a panel of markers for the cell types under study. As a side-note in their paper, Abdelaal *et al.* [89] mention that selecting marker genes for 'deeply annotated datasets' (fine cell type hierarchies) becomes infeasible. I agree it is a challenging task, but have shown for the MALT data that indeed it is feasible to subdivide T cells into finer and finer subtypes. This is especially true for patient cohorts and other multi-sample multi-condition studies: Sequencing more patients (samples) rarely introduces new cell types, so once a suitable marker set has been found for a certain tissue, disease and organism, it can be applied again and again. Furthermore, curating marker genes can have value in itself, because simple marker-based cell type definitions that describe the complex biology well are helpful for science. Thus, curating marker sets that apply in many different experimental settings, tedious as it might be, should be considered an achievement which requires integration of exploratory data analysis with prior knowledge from the literature spanning sequencing and non-sequencing methods. Finally, I note that with PCPC, it is no longer necessary to decide on the final panel before running the algorithm: In contrast to CellAssign, Garnett, SCINA and scSorter, PCPC makes it easy to experiment with different markers interactively (discussed further in section 4.2, see also interactive code in outlook section 5.2).

For lowly expressed marker genes, PCPC is furthermore unable to find positive and negative cells at the same time. For example, if FOXP3 is used to mark T_{regs}, we can not use the same threshold value to find FOXP3$^-$ cells (c.f. section 3.3.2). The sparse FOXP3 expression simply provides not enough information, so absence of UMI detection does not prove low expression in this neighborhood due to high technical noise. In general, I advise to use positive markers wherever possible (for instance, ANXA1$^+$ instead of FOXP3$^-$) and to select markers with high expression wherever possible. This way, inference results deduced from the cell type definitions are more likely to generalize to data sets generated with less sequencing depth or imaging methods such as RNA fluorescent *in situ* hybridization.

Subjective bias PCPC is highly flexible (c.f. section 4.2), but this is prone to introducing personal biases. By freely chosing marker genes and the corresonding thresholds, a researcher has few constraints in choosing any group of cells he or she pleases. In principle, generating nonsensical classification results is thus possible. Also, PCPC may enable so called 'p-hacking', i.e. the slight adjustment of algorithm parameters until the desired results are obtained. For example, by changing thresholds or the marker panel time and time again, the list of differentially expressed genes may be influenced until a certain gene of interest is contained in it – defeating the purpose of multiple testing correction and other scientific principles. Thus, classification with PCPC is only as rigorous as the investigator using it.

The most obvious personal bias, however, is the selection of marker genes. With PCPC, a single marker can be enough to classify certain cell types, but this raises the question which marker gene to use. For example, if FOXP3 and IL2RA are both specific to regulatory T cell cells in a given data set, the choice which one to use can be a highly subjective one. In flow cytometry or imaging, researchers are often forced to select a single or few markers, while scRNAseq offers the entire transcriptome – Is it thus not an unnecessary restriction to choose between FOXP3 and IL2RA? The crucial difference, I argue, is that whenever necessary, the analysis can be repeated with another set of markers. For example, if it is crucial to the scientific question, a rigorous researcher could show to his scientific peers that the reported results do not change in essence when IL2RA is used instead of FOXP3. Thus, during the process of peer review, the personal bias introduced by selecting marker genes can well be controlled.

Before pointing out more counter-measures guarding against subjective biases, I note that unsupervised algorithms such as clustering also have their biases. As demonstrated in section 3.3.4, clustering results depend largely on pre-processing choices. Another subjective researcher choice in clustering is the resolution parameter implemented in modern versions of graph-based clustering algorithms (introduced in section 1.7.2). When applied to more than one sample, clustering furthermore requires batch effect removal, introducing more subjectivity. This is because data integration has a trade-off between cell type resolution and the mixing of samples [135], which the user decides indirectly by algorithm choice [135] or directly (e.g. alignment strength in Conos [73]). Thus, unsupervised algorithms are not 'unbiased', but have more subtle and hidden biases. Not only do they not prevent 'p-hacking', they may even hide it within the complexity of these algorithms. In this context, it is important to note that PCPC clearly states marker genes and thresholds used, which are more accessible to interpretation and revision by scientific peers than the hyperparameters of unsupervised algorithms.

Two important mechanisms against subjective biases are scientific debate and falsification. As introduced in section 3.1 and pointed out above, the simple marker-based cell type definitions used by PCPC make scientific debates clearer and can thus also make peer review more effective and efficient. At the same time, they bring falsification to scRNAseq inference findings: If the claim is that FOXP3$^+$ cells upregulate certain genes under certain conditions, it is easy to test and falsify this inference result.

In summary, the personal bias introduced in classification with PCPC is not necessarily more severe than that introduced by unsupervised methods (making it misleading to call them 'unbiased', c.f. section 3.3.4), but it is more accessible to scientific peers.

Manual labor It is clear that running PCPC once on a patient cohort is more laborious than running unsupervised methods (clustering after data integration or label transfer, introduced in sections 1.7.2 and 1.7.3). This does not mean that unsupervised methods do not require a lot of work when applied rigorously – testing different parameters or even different algorithms for data integration and clustering can also be highly time-consuming. At any rate, there are differences between PCPC and unsupervised methods in how a researcher invests time, which I discuss in more detail here.

PCPC is time-consuming in two ways. First, picking thresholds for each gene requires some time, and my personal opinion is that this is feasible for dozens of patients but perhaps not hundreds (see section 5.2 for interactive code to make threshold picking convenient). Second, finding mutually exclusive markers is tedious, and in my experience takes more time than picking thresholds. As argued in section 4.5, however, this may only be necessary once for a given disease and be of value on its own to drive biological understanding onward. Thus, PCPC splits the classification task and the time invested into two apects: Finding good markers (intelligent work), and finding good thresholds (repetitive work). This enables iterative data analysis: A preliminary marker panel can be used for a first end-to-end analysis, using for example only major cell types in differential expression testing between experimental conditions. After more exploratory data analysis, this marker panel may be refined, but only new markers now require threshold picking. If pre-processing changes (doublet removal, filtering genes and cells, data integration, ...) or more patients are coming in, it is conceivable that the list of differentially expressed genes becomes more precise, but the top hits might not even change. Thus, experiments to confirm these preliminary results and novel hypotheses generated from them can already be started, even if the final cell type labels have not been reached. This uncoupling of validation experiments and the tedious progression towards ideal cell type labels is particularly useful in scRNAseq cohort studies, which typically generate data in multiple 'waves', i.e. once every few months when more patient materials has been acquired.

For unsupervised methods such as clustering after data integration and label transfer, only little amounts of human labor seems to be required at first glance. I argue, however, that previous time investments are mostly wasted once the pre-processing changes. In large cohort studies that span multiple years of effort, this may happen several times. Specifically, data integration and clustering have to be recomputed and re-annotated every time a new samples arrives or algorithm parameters change for quality control, doublet removal, normalization or data integration. When using PCPC, in contrast, once appropriate marker genes have been found they remain the same if preprocessing changes – the user can supply the updated kNN and re-run previous code in seconds. Whether thresholds can be re-used in such cases, and whether the manual labor is indeed manageable, will become clear once PCPC is applied to a novel scRNAseq cohort study and remains to be seen.

Dependence on UMAP embeddings The user picks thresholds interactively with PCPC by trial-and-error, comparing marker expression in UMAP embeddings [10] to the positive cells identified by the current threshold value. This has implications that I discuss in the following.

One limitation of our approach is that if cells from a given subpopulation are not somewhat grouped together in the embedding, they can not be confidently classified using PCPC. The user simply does not know how to pick a reasonable threshold, and it is uncertain if the resulting classification is reliable. Thus, as a rule of thumb, if UMAP can not resolve a subpopulation, PCPC also can not (in particular if they operate on the exact same kNN graph). In contrast to this, CellAssign [141] and SCINA [142] might be able to resolve such a subpopulation due to their architecture. Since both algorithms are designed to work with many marker genes (typically dozens, as introduced in section 1.7.4), the user might use a large marker panel ob-

[10] Any visualization tool next to UMAP can be used, for example tSNE. Throughout this thesis, I use UMAP because weak evidence exists that it outperforms tSNE on scRNAseq data [49], but note that this has been contended [50].

tained for example from bulk RNAseq after cell sorting. In any case, however, researchers may find it difficult in trusting any findings concerning a subpopulation that is scattered across the embedding. Also, if the data does not provide enough signal for UMAP [11], it is questionable whether this data set is rich enough to add striking new insights into this evasive subpopulation.

The tight coupling of UMAP embeddings and PCPC has another potential drawback. They do not provide independent perspectives on the data, since both rely on the same kNN graph. Thus, any artifacts introduced into the kNN graph, for example by technical noise or the approximate kNN algorithm (see methods), is expected to manifest in UMAP and PCPC alike. A more elegant approach would be to find positive cells without kNN information. This way, PCPC and UMAP would be more likely to disagree whenever UMAP is 'wrong', i.e. placing similar cells far apart or dissimilar cells into close proximity to each other. In particular, a future direction of PCPC could be to find positive cells based on smoothed values and their uncertainties computed by SAVER [35] (introduced in section 1.6). Briefly, SAVER regresses marker gene UMI counts on the expression of all other genes, not using kNN information at all. PCPC could thus replace its Poisson tails approach by asking whether the smoothed values found by SAVER lies clearly above a given threshold (for example, by 3 standard errors or more), using the estimates of uncertainty that SAVER also computes. This would make PCPC more orthogonal to UMAP and other kNN approaches, and I would welcome future research in this direction.

As a final note, I stress that PCPC does not rely on UMAP beyond visualization. This is because reducing high dimensional information (such as 50 principal components representing thousands of genes) to two dimensions is a simplified view on the data that should not be expected to capture all relevant information. Instead, PCPC bases its annotations on careful statistical assumptions derived from the Poisson distribution as described in section 3.2 and discussed in section 4.4.

[11] The interplay between a data set's 'signal-to-noise ratio' and the ability to resolve highly related subpopulations are discussed in section 4.1.

Chapter 5

Outlook

This thesis describes the ongoing development of Pooled Count Poisson Classification (PCPC), and I now discuss aspects that I find are relevant to address in the near future. Automating PCPC is not discussed here, see section 4.2.

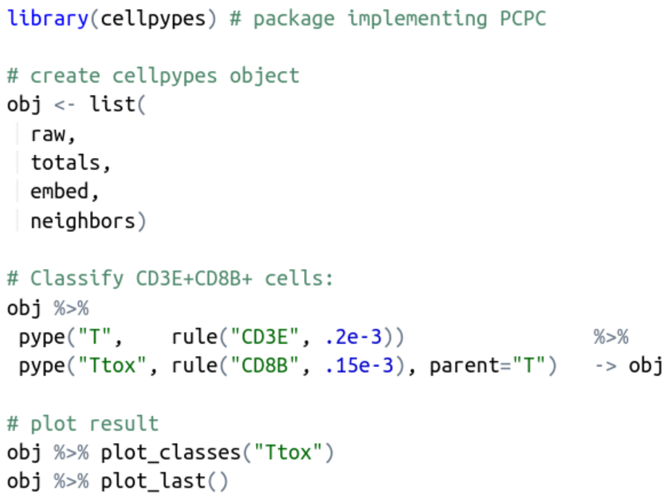
5.1 Large cohort, all subpopulations

I have demonstrated that PCPC is able to annotate all cell types in complex tumor microenvironments (section 3.3.3) and a single cell type in a multi-sample multi-condition lymphoma cohort (section 3.5). I consider this a first proof of principle, but stress that the goal is to annotate all subpopulations in dozens of patients. UMI-protocols have spread widely throughout the research community and I assume that scRNAseq cohorts will keep growing in the future, so I believe PCPC will have plenty of opportunities to be applied once it is available as R package (see section 5.2). In particular, testing PCPC on a large, novel scRNAseq cohort will answer three questions. First, does manual thresholding scale reasonably with increasing patient numbers, and can mutually exclusive marker genes be found in all cases (c.f. sections 4.5 and 4.5)? Second, how does pre-processing (normalization, cell and gene filtering, etc.) influence classification performance, and more importantly the resulting inference insights (e.g.differentially gene expression)? I would assume that providing refined k nearest neighbors (kNN) information and re-using the same genes and thresholds with PCPC is a convenient and fast way to test the influence of such analysis choices, highlighting their relative importance to other aspects of data analysis. And third, does PCPC prove useful beyond immune cells? Specifically, the classification examples I present here use three data sets (CBMC, MALT and the lymphoma cohort) that mainly consist of immune cells, which express highly specific markers that are required for their function. The next exciting step is to apply PCPC to a non-immune cell cohort, to test its applicability for a wide range of scientific fields. In initial experiments, I observed that PCPC performs well on stem cells and their progeny from the fly gut (not shown, unpublished) and two recent brain disease cohorts [60, 61] (not shown), but this is anecdotal evidence until this part of the project is finished as well.

5.2 Interactive programming with *cellpypes*

In order to apply it to larger cohorts, PCPC requires well-structured code that makes interactive usage fast and clear. To this end, I started developing the *cellpypes* R package, which I briefly introduce now.

When developing PCPC and writing this thesis, I have found that the most time consuming part of classification is exploration, not manual thresholding. Specifically, the user of PCPC has to answer these questions in order to annotate the entire data set: Which cell types are present, and which markers result in mutually exclusive subpopulations? This requires code which makes changing marker genes and cell type definitions not much more difficult than adjusting threshold values. To this end, I currently develop the *cellpypes* package, which implements PCPC using the highly interactive R pipes from the magrittr package [178].



```
library(cellpypes) # package implementing PCPC

# create cellpypes object
obj <- list(
  raw,
  totals,
  embed,
  neighbors)

# Classify CD3E+CD8B+ cells:
obj %>%
 pype("T",    rule("CD3E", .2e-3))                %>%
 pype("Ttox", rule("CD8B", .15e-3), parent="T")   -> obj

# plot result
obj %>% plot_classes("Ttox")
obj %>% plot_last()
```

Figure 5.1: Code example for using the *cellpypes* package (under development), a highly interactive implementation of PCPC using magrittr's pipe operator "%>%". Object creation, classification and plotting is shown, for explanations see main text.

Figure 5.1 shows how cytotoxic T cells can be classified with the *cellpypes* package providing clear and flexible structure. After loading the *cellpypes* package (first line in Figure 5.1), the user creates an R object with the UMI count matrix (*raw*), the total UMI of each cell (*totals*, i.e. the sum from all columns in *raw*), a two-dimensional embedding computed for example with UMAP (*embed*) and an unweighted kNN graph (*neighbors*). The kNN graph can either be supplied as binary adjacency matrix (square matrix with 1 if neighbor and 0 otherwise) or as a matrix with the indices of each cell's kNN (with k columns and as many rows as there are cells). I note that these four inputs are readily available from scRNAseq data objects processed with any pipeline (e.g. Seurat [37], scater/scran [179], scanpy [38], monocle2 [39]), because the neighbor information is computed by UMAP and clustering algorithms – I therefore intend that *cellpypes* can wrap objects from all of these pipelines for the user's convenience. I note that weighted graphs can in principle also be used, as discussed in section 4.3.

In order to classify CD3E$^+$CD8B$^+$ cells, the user specifies classes and rules with the *pype* command (*pype* is short for 'cell type pipes'). The example in Figure 5.1 classifies T cells as those cells whose CD3E expression was estimated to be above 0.2 ‰. Internally, the *pype* command selects cells for which the pooled counts were higher than expected from a Poisson distribution around 0.2 ‰ of the pooled totals (c.f. 3.2.4). The next line again uses the *pype* command, to define Ttox (short for cytotoxic T cells) as a subset of T cells, requiring CD8B expression above 0.15 ‰. As a further refinement, the user could attempt to exclude doublets formed from co-capturing a T with a B cells by demanding that his CD3E$^+$CD8B$^+$ cells are on top negative for a B cell marker, such as CD79B or MS4A1 (not shown). For this negative thresholding, the *rule* in the *pype* command can also accept an additional operator argument ("<" instead of ">", not shown in this example).

I have used this implementation to classify cytotoxic T cells in the lymphoma cohort (c.f. Figure 3.12). In order to decide on the exact genes and thresholds, the user can for example replace CD3E with CD3G or 0.2 ‰ with 0.3 ‰, and inspect the new result with the *plot_classes* or *plot_last* functions (last lines in Figure 5.1). The *plot_classes* function combines all rules that apply for this class, so in this example would show CD3E$^+$CD8B$^+$ cells in a UMAP embedding (effectively this generates Figure 3.12c). In contrast, *plot_last* only displays the rule in the most recent *pype*, which in this case would be CD8B$^+$ cells. Conveniently, *plot_last* on top shows a feature plot of the relevant gene (CD8B here), much like in Figure 3.12a and b, so that the user may compare the currently selected cells in UMAP with the marker gene expression.

Thus, cytotoxic T cells are classified with two lines of code, and the user can adjust thresholds (0.2 ‰ and 0.15 ‰) and genes (CD3E and CD8B) interactively. Once the user is satisfied with the resulting cell type labels, these can be used to form pseudobulk samples from this patient, or simply be exported and saved for later (not shown). In order to annotate a new sample (other patient, other treatment, etc.), these few lines of code can simply be copied and pasted below a different *cellpypes* object, and the threshold be adjusted where necessary.

I expect this way of annotating multi-sample multi-condition cohorts to scale quite well to dozens of patients. One strength of this approach is that advances in analysis can directly be fed in. If exploration and literature reveal after some time that a different marker panel might be more appropriate, the user can swiftly experiment with it. Conveniently, the clear and concise code represents a compact documentation of how the cell types were defined. Finally, the user perhaps wants to compare whether data integration (c.f. next section) improves the nearest neighbor information or not. After updating *neighbors* in the *cellpypes* object, the code with the same genes (perhaps even same thresholds) can be re-run and the result immediately inspected to see if the novel neighbor information leaves fewer cells unassigned, for example.

5.3 Analyze samples together instead of separately

In the lymphoma cohort, I have chosen to process each sample separately. As an alternative, data from different patients could first be integrated, with the hope of refining kNN information this way (as explained in 2.1). Numerous methods for integration have been proposed [73, 122, 128–133], and each can be used to compute kNN graphs. It would be interesting to test how much any potential differences in kNN information will influence PCPC classification. Also, it may turn out that batch effect removal is not necessary: I expect that kNN information improves when patients are pooled when computing PCA embeddings. A batch effect separating patients in this embedding is not necessarily a nuisance for classification with PCPC – cells would be assigned nearest neighbors from the same patient sample, which might be desirable anyways. While clustering strictly requires the removal of batch effects using data integration methods, PCPC could thus work without it while still profiting from more patients (by making gene-gene correlations less noisy, improving the PCA embedding). This is desirable since removing batch effects comes at the price of removing some biological variation as well, blurring subtle biological cell type differences [135]. On the same note, PCPC might profit from using scTransform for normalization, or GLM-PCA [1] (a method that adapts principal component analysis specifically to UMI count data, not requiring normalization).

Finally, pooling data across several patients might enable the detection of rare cell types. This is because in a single sample, a handful of cells are unlikely to separate in UMAP due to the low signal-to-noise ratio. When pooled across many samples, however, the rare subpopulation might reach a critical abundance and separate.

5.4 Future work concerning nearest neighbors

In this work, I use the 50 nearest neighbors of each cell. As discussed in section 4.3, it might be more robust to prune the kNN graph and only retain indistinguishable neighbors. In any case, this would ensure straight forward interpretability of positive cells (as described in section 4.1.3). As discussed in section 4.5, it would also be interesting to use gene smoothing with SAVER to find positive and negative cells. SAVER was shown to smooth more conservatively than other methods [67] and is independent from kNN information. This could make PCPC orthogonal to UMAP and clustering, in the sense that artifacts specific to either the kNN search or gene-wise smoothing would show as disagreement between UMAP and PCPC.

Along a similar line of reasoning, I would like to see another question answered in the future: If the data set at hand does not have enough information to resolve a given subpopulation, would PCPC help the user to realize this? A key difference between PCPC and exploratory tools such as clustering is that cell type definitions are used as prior knowledge, so it is not only interesting which cell types we find, but also which cell types we do not find with the given data. From PCPC's architecture (c.f. section 3.2.4), I would expect that PCPC adapts to the available signal-to-noise ratio, leaving questionable cells unassigned rather than accepting misclassifications. In order to show this, one could classify FOXP3$^+$ cells in the MALT data with increasingly poorer kNN information (by exchanging each cell's 5, 10, 20 and 50 kNN with randomly selected cells). This is work to be done in the future.

Chapter 6

References

1. Townes, F. W., Hicks, S. C., Aryee, M. J. & Irizarry, R. A. Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. *Genome biology* **20,** 1–16 (2019).

2. Grabski, I. N. & Irizarry, R. A. Probabilistic gene expression signatures identify cell-types from single cell RNA-seq data. *bioRxiv.* doi:10.1101/2020.01.05.895441. eprint: https://www.biorxiv.org/content/early/2020/01/23/2020.01.05.895441.full.pdf. https://www.biorxiv.org/content/early/2020/01/23/2020.01.05.895441 (2020).

3. Rogers, S. *Data are or data is?* Accessed on 27th Apr 2021. https://www.theguardian.com/news/datablog/2010/jul/16/data-plural-singular.

4. Lao, K. Q. *et al.* mRNA-Sequencing Whole Transcriptome Analysis of a Single Cell on the SOLiD™ System. *Journal of biomolecular techniques: JBT* **20,** 266 (2009).

5. Picelli, S. *et al.* Full-length RNA-seq from single cells using Smart-seq2. *Nature protocols* **9,** 171 (2014).

6. Macosko, E. Z. *et al.* Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* **161,** 1202–1214 (2015).

7. Peng, A., Mao, X., Zhong, J., Fan, S. & Hu, Y. Single-Cell Multi-Omics and Its Prospective Application in Cancer Biology. *Proteomics* **20,** 1900271 (2020).

8. Han, X. *et al.* Mapping the mouse cell atlas by microwell-seq. *Cell* **172,** 1091–1107 (2018).

9. Gierahn, T. M. *et al.* Seq-Well: portable, low-cost RNA sequencing of single cells at high throughput. *Nature methods* **14,** 395–398 (2017).

10. Verboom, K. *et al.* SMARTer single cell total RNA sequencing. *Nucleic acids research* **47,** e93–e93 (2019).

11. Goetz, J. J. & Trimarchi, J. M. Transcriptome sequencing of single cells with Smart-Seq. *Nature biotechnology* **30,** 763–765 (2012).

12. Hagemann-Jensen, M. *et al.* Single-cell RNA counting at allele and isoform resolution using Smart-seq3. *Nature Biotechnology* **38,** 708–714 (2020).

13. Hashimshony, T. *et al.* CEL-Seq2: sensitive highly-multiplexed single-cell RNA-Seq. *Genome biology* **17,** 1–7 (2016).

14. Rosenberg, A. B. *et al.* Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science* **360,** 176–182 (2018).

15. 10x Genomics. *What fraction of mRNA transcripts are captured per cell?* Accessed on 23rd Nov 2020. https://kb.10xgenomics.com/hc/en-us/articles/360001539051-What-fraction-of-mRNA-transcripts-are-captured-per-cell-.

16. Zheng, G. X. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nature communications* **8,** 1–12 (2017).

17. Picelli, S. *et al.* Tn5 transposase and tagmentation procedures for massively scaled sequencing projects. *Genome research* **24,** 2033–2040 (2014).

18. 10x Genomics. *What is Cell Ranger?* Accessed on 23rd Nov 2020. https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger.

19. Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29,** 15–21 (2013).

20. Petukhov, V. *et al.* dropEst: pipeline for accurate estimation of molecular counts in droplet-based single-cell RNA-seq experiments. *Genome biology* **19,** 78 (2018).

21. Melsted, P., Ntranos, V. & Pachter, L. The barcode, UMI, set format and BUStools. *Bioinformatics* **35,** 4472–4473. ISSN: 1367-4803 (May 2019).

22. Zheng, G. X. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nature communications* **8,** 1–12 (2017).

23. Jaitin, D. A. *et al.* Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. *Science* **343,** 776–779 (2014).

24. Islam, S. *et al.* Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome research* **21,** 1160–1167 (2011).

25. Brennecke, P. *et al.* Accounting for technical noise in single-cell RNA-seq experiments. *Nature methods* **10,** 1093–1095 (2013).

26. Kharchenko, P. V., Silberstein, L. & Scadden, D. T. Bayesian approach to single-cell differential expression analysis. *Nature methods* **11,** 740–742 (2014).

27. Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S. & Vert, J.-P. ZINB-WaVE: A general and flexible method for signal extraction from single-cell RNA-seq data. *BioRxiv,* 125112 (2017).

28. Grün, D., Kester, L. & Van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. *Nature methods* **11,** 637–640 (2014).

29. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology* **15,** 1–21 (2014).

30. Wang, J. *et al.* Data denoising with transfer learning in single-cell transcriptomics. *Nature methods* **16,** 875–878 (2019).

31. Gong, W., Kwak, I.-Y., Pota, P., Koyano-Nakagawa, N. & Garry, D. J. DrImpute: imputing dropout events in single cell RNA sequencing data. *BMC bioinformatics* **19,** 220 (2018).

32. Svensson, V. Droplet scRNA-seq is not zero-inflated. *Nature Biotechnology,* 1–4 (2020).

33. Wagner, F., Yan, Y. & Yanai, I. K-nearest neighbor smoothing for high-throughput single-cell RNA-Seq data. *bioRxiv.* doi:10.1101/217737. eprint: https://www.biorxiv.org/content/early/2018/04/09/217737.full.pdf. https://www.biorxiv.org/content/early/2018/04/09/217737 (2018).

34. Wang, J. *et al.* Gene expression distribution deconvolution in single-cell RNA sequencing. *Proceedings of the National Academy of Sciences* **115,** E6437–E6446 (2018).

35. Huang, M. *et al.* SAVER: gene expression recovery for single-cell RNA sequencing. *Nature methods* **15,** 539–542 (2018).

36. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications* **10,** 1–14 (2019).

37. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology* **36,** 411–420 (2018).

38. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology* **19,** 1–5 (2018).

39. Qiu, X. *et al.* Reversed graph embedding resolves complex single-cell trajectories. *Nature methods* **14,** 979 (2017).

40. Pliner, H. A., Shendure, J. & Trapnell, C. Supervised classification enables rapid annotation of cell atlases. *Nature methods* **16,** 983–986 (2019).

41. Hafemeister, C. & Satija, R. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome biology* **20,** 1–15 (2019).

42. Satija Lab. *Seurat - Guided Clustering Tutorial [v3.0]* satijalab.org/seurat/archive/v3.0/pbmc3k_tutorial.html. Accessed: 2020-02-29. June 2019.

43. Andrews, T. S. & Hemberg, M. M3Drop: dropout-based feature selection for scRNASeq. *Bioinformatics* **35,** 2865–2867. ISSN: 1367-4803 (Dec. 2018).

44. Kim, T. H., Zhou, X. & Chen, M. Demystifying 'drop-outs' in single cell UMI data. *Genome biology* **21** (2020).

45. Lun, A. Overcoming systematic errors caused by log-transformation of normalized single-cell RNA sequencing data. *BioRxiv,* 404962 (2018).

46. Kiselev, V. Y., Andrews, T. S. & Hemberg, M. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics* **20,** 273–282 (2019).

47. McInnes, L., Healy, J. & Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

48. Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **9** (2008).

49. Becht, E. *et al.* Dimensionality reduction for visualizing single-cell data using UMAP. *Nature biotechnology* **37,** 38–44 (2019).

50. Kobak, D. & Berens, P. The art of using t-SNE for single-cell transcriptomics. *Nature communications* **10,** 1–14 (2019).

51. Linderman, G. C., Rachh, M., Hoskins, J. G., Steinerberger, S. & Kluger, Y. Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. *Nature methods* **16,** 243–245 (2019).

52. Consortium, T. M. *et al.* Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature* **562,** 367 (2018).

53. Zeisel, A. *et al.* Molecular architecture of the mouse nervous system. *Cell* **174,** 999–1014 (2018).

54. Reid, A. J. *et al.* Single-cell RNA-seq reveals hidden transcriptional variation in malaria parasites. *Elife* **7,** e33105 (2018).

55. Davie, K. *et al.* A single-cell transcriptome atlas of the aging Drosophila brain. *Cell* **174,** 982–998 (2018).

56. Rozenblatt-Rosen, O., Stubbington, M. J., Regev, A. & Teichmann, S. A. The Human Cell Atlas: from vision to reality. *Nature News* **550,** 451 (2017).

57. Montoro, D. T. *et al.* A revised airway epithelial hierarchy includes CFTR-expressing ionocytes. *Nature* **560,** 319–324 (2018).

58. Pal, B. *et al.* Construction of developmental lineage relationships in the mouse mammary gland by single-cell RNA profiling. *Nature communications* **8,** 1–14 (2017).

59. Villani, A.-C. *et al.* Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science* **356** (2017).

60. Schirmer, L. *et al.* Neuronal vulnerability and multilineage diversity in multiple sclerosis. *Nature* **573,** 75–82 (2019).

61. Velmeshev, D. *et al.* Single-cell genomics identifies cell type–specific molecular changes in autism. *Science* **364,** 685–689 (2019).

62. Roider, T. *et al.* Dissecting intratumour heterogeneity of nodal B-cell lymphomas at the transcriptional, genetic and drug-response levels. *Nature Cell Biology* **22,** 896–906 (2020).

63. Azizi, E. *et al.* Single-cell map of diverse immune phenotypes in the breast tumor microenvironment. *Cell* **174,** 1293–1308 (2018).

64. Lambrechts, D. *et al.* Phenotype molding of stromal cells in the lung tumor microenvironment. *Nature medicine* **24,** 1277–1289 (2018).

65. Crowell, H. L. *et al.* muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nature communications* **11,** 1–12 (2020).

66. Zhang, J. M., Kamath, G. M. & David, N. T. Valid post-clustering differential analysis for single-cell RNA-Seq. *Cell systems* **9,** 383–392 (2019).

67. Andrews, T. S. & Hemberg, M. False signals induced by single-cell imputation. *F1000Research* **7** (2018).

68. Van Dijk, D. *et al.* Recovering gene interactions from single-cell data using data diffusion. *Cell* **174,** 716–729 (2018).

69. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature methods* **15,** 1053–1058 (2018).

70. Lin, P., Troup, M. & Ho, J. W. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome biology* **18,** 1–11 (2017).

71. Li, W. V. & Li, J. J. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature communications* **9,** 1–9 (2018).

72. Chi, W. & Deng, M. Sparsity-Penalized Stacked Denoising Autoencoders for Imputing Single-Cell RNA-seq Data. *Genes* **11.** ISSN: 2073-4425. doi:10.3390/genes11050532. https://www.mdpi.com/2073-4425/11/5/532 (2020).

73. Barkas, N. *et al.* Joint analysis of heterogeneous single-cell RNA-seq dataset collections. *Nature methods* **16,** 695–698 (2019).

74. Haghverdi, L., Büttner, M., Wolf, F. A., Buettner, F. & Theis, F. J. Diffusion pseudotime robustly reconstructs lineage branching. *Nature methods* **13,** 845 (2016).

75. Baran, Y. *et al.* MetaCell: analysis of single-cell RNA-seq data using K-nn graph partitions. *Genome biology* **20,** 1–19 (2019).

76. Azizi, E., Prabhakaran, S., Carr, A. & Pe'er, D. Bayesian inference for single-cell clustering and imputing. *Genomics and Computational Biology* **3,** e46–e46 (2017).

77. Prabhakaran, S., Azizi, E., Carr, A. & Pe'er, D. *Dirichlet process mixture model for correcting technical variation in single-cell gene expression data* in *International Conference on Machine Learning* (2016), 1070–1079.

78. Wagner, F. & Yanai, I. Moana: A robust and scalable cell type classification framework for single-cell RNA-Seq data. *bioRxiv.* doi:10.1101/456129. eprint: https://www.biorxiv.org/content/early/2018/10/30/456129.full.pdf. https://www.biorxiv.org/content/early/2018/10/30/456129 (2018).

79. Trapnell, C. Defining cell types and states with single-cell genomics. *Genome research* **25,** 1491–1498 (2015).

80. Wagner, A., Regev, A. & Yosef, N. Revealing the vectors of cellular identity with single-cell genomics. *Nature biotechnology* **34,** 1145–1160 (2016).

81. Clevers, H. *et al.* What is your conceptual definition of "cell type" in the context of a mature organism? *Cell Systems* **4,** 255–259 (2017).

82. Morris, S. A. The evolving concept of cell identity in the single cell era. *Development* **146,** dev169748 (2019).

83. Xia, B. & Yanai, I. A periodic table of cell types. *Development* **146,** dev169854 (2019).

84. Song, Q., Su, J. & Zhang, W. scGCN: a Graph Convolutional Networks Algorithm for Knowledge Transfer in Single Cell Omics. *bioRxiv.* doi:10.1101/2020.09.13.295535. eprint: https://www.biorxiv.org/content/early/2020/09/14/2020.09.13.295535.full.pdf. https://www.biorxiv.org/content/early/2020/09/14/2020.09.13.295535 (2020).

85. Guilliams, M. & van de Laar, L. A Hitchhiker's guide to myeloid cell subsets: practical implementation of a novel mononuclear phagocyte classification system. *Frontiers in immunology* **6,** 406 (2015).

86. Arendt, D. *et al.* The origin and evolution of cell types. *Nature Reviews Genetics* **17,** 744–757 (2016).

87. Crowell, H. L. *et al.* On the discovery of subpopulation-specific state transitions from multi-sample multi-condition single-cell RNA sequencing data. *bioRxiv.* doi:10.1101/713412. eprint: https://www.biorxiv.org/content/early/2020/08/27/713412.full.pdf. https://www.biorxiv.org/content/early/2020/08/27/713412 (2020).

88. Diaz-Mejia, J. J. *et al.* Evaluation of methods to assign cell type labels to cell clusters from single-cell RNA-sequencing data. *F1000Research* **8** (2019).

89. Abdelaal, T. *et al.* A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome biology* **20,** 194 (2019).

90. Kiselev, V. Y. *et al.* SC3: consensus clustering of single-cell RNA-seq data. *Nature methods* **14,** 483–486 (2017).

91. Grün, D. *et al.* Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature* **525,** 251–255 (2015).

92. Wang, B., Zhu, J., Pierson, E., Ramazzotti, D. & Batzoglou, S. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nature methods* **14,** 414–416 (2017).

93. Guo, M., Wang, H., Potter, S. S., Whitsett, J. A. & Xu, Y. SINCERA: a pipeline for single-cell RNA-Seq profiling analysis. *PLoS computational biology* **11,** e1004575 (2015).

94. Yau, C. *et al.* pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC bioinformatics* **17,** 1–11 (2016).

95. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* **2008,** P10008 (2008).

96. Traag, V. A., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Scientific reports* **9,** 1–12 (2019).

97. Waltman, L. & Van Eck, N. J. A smart local moving algorithm for large-scale modularity-based community detection. *The European physical journal B* **86,** 1–14 (2013).

98. Reichardt, J. & Bornholdt, S. Statistical mechanics of community detection. *Physical review E* **74,** 016110 (2006).

99. Lambiotte, R., Delvenne, J.-C. & Barahona, M. Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint* (2008).

100. Levine, J. H. *et al.* Data-driven phenotypic dissection of AML reveals progenitor-like cells that correlate with prognosis. *Cell* **162,** 184–197 (2015).

101. Xu, C. & Su, Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics* **31,** 1974–1980 (2015).

102. Miao, Z. *et al.* Putative cell type discovery from single-cell gene expression data. *Nature Methods* **17,** 621–628 (2020).

103. Aibar, S. *et al.* SCENIC: single-cell regulatory network inference and clustering. *Nature methods* **14,** 1083–1086 (2017).

104. Fiers, M. W. E. J. *et al.* Mapping gene regulatory networks from single-cell omics data. *Briefings in Functional Genomics* **17,** 246–254. ISSN: 2041-2657 (Jan. 2018).

105. Holland, C. H. *et al.* Robustness and applicability of transcription factor and pathway analysis tools on single-cell RNA-seq data. *Genome biology* **21,** 36 (2020).

106. Pratapa, A., Jalihal, A. P., Law, J. N., Bharadwaj, A. & Murali, T. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods* **17,** 147–154 (2020).

107. Zeng, Y., Zhou, X., Rao, J., Lu, Y. & Yang, Y. Accurately Clustering Single-cell RNA-seq data by Capturing Structural Relations between Cells through Graph Convolutional Network. *bioRxiv.* doi:10.1101/2020.09.02.278804. eprint: https://www.biorxiv.org/content/early/2020/09/03/2020.09.02.278804.full.pdf. https://www.biorxiv.org/content/early/2020/09/03/2020.09.02.278804 (2020).

108. Köhler, N. D., Büttner, M. & Theis, F. J. Deep learning does not outperform classical machine learning for cell-type annotation. *bioRxiv.* doi:10.1101/653907. eprint: https://www.biorxiv.org/content/early/2019/05/31/653907.full.pdf. https://www.biorxiv.org/content/early/2019/05/31/653907 (2019).

109. Hao, Y. *et al.* Integrated analysis of multimodal single-cell data. *bioRxiv.* doi:10.1101/2020.10.12.335331. eprint: https://www.biorxiv.org/content/early/2020/10/12/2020.10.12.335331.full.pdf. https://www.biorxiv.org/content/early/2020/10/12/2020.10.12.335331 (2020).

110. Argelaguet, R. *et al.* MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. *Genome Biology* **21,** 1–17 (2020).

111. Welch, J. D., Hartemink, A. J. & Prins, J. F. MATCHER: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics. *Genome biology* **18,** 1–19 (2017).

112. Welch, J. D. *et al.* Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell* **177,** 1873–1887 (2019).

113. Stoeckius, M. *et al.* Simultaneous epitope and transcriptome measurement in single cells. *Nature methods* **14,** 865 (2017).

114. Peterson, V. M. *et al.* Multiplexed quantification of proteins and transcripts in single cells. *Nature biotechnology* **35,** 936–939 (2017).

115. BioLegend. *Multiomics and TotalSeq Reagents* Accessed on 11th Mar 2021. https://www.biolegend.com/en-us/totalseq.

116. Cao, J. *et al.* Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science* **361,** 1380–1385 (2018).

117. Chen, S., Lake, B. B. & Zhang, K. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nature biotechnology* **37,** 1452–1457 (2019).

118. 10x Genomics. *Single Cell Multiome ATAC + Gene Expression* Accessed on 11th Mar 2021. https://www.10xgenomics.com/products/single-cell-multiome-atac-plus-gene-expression.

119. Clark, S. J. *et al.* scNMT-seq enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells. *Nature communications* **9,** 1–9 (2018).

120. Hao, Y. *et al.* Integrated analysis of multimodal single-cell data. *bioRxiv.* doi:10.1101/2020.10.12.335331. eprint: https://www.biorxiv.org/content/early/2020/10/12/2020.10.12.335331.full.pdf. https://www.biorxiv.org/content/early/2020/10/12/2020.10.12.335331 (2020).

121. Fan, J. *Stability testing: How do you know whether your single-cell clusters are 'real'?* Feb. 2018. https://jef.works/blog/2018/02/28/stability-testing/.

122. Stuart, T. *et al.* Comprehensive integration of single-cell data. *Cell* **177,** 1888–1902 (2019).

123. Aran, D. *et al.* Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nature immunology* **20,** 163–172 (2019).

124. Hou, R., Denisenko, E. & Forrest, A. R. scMatch: a single-cell gene expression profile annotation tool using reference datasets. *Bioinformatics* **35,** 4688–4695 (2019).

125. Satija Lab. *Azimuth - App for reference-based single-cell analysis* Accessed on 24th Nov 2020. https://satijalab.org/azimuth/.

126. De Kanter, J. K., Lijnzaad, P., Candelli, T., Margaritis, T. & Holstege, F. C. CHETAH: a selective, hierarchical cell type identification method for single-cell RNA sequencing. *Nucleic acids research* **47,** e95–e95 (2019).

127. Alquicira-Hernandez, J., Sathe, A., Ji, H. P., Nguyen, Q. & Powell, J. E. scPred: accurate supervised method for cell-type classification from single-cell RNA-seq data. *Genome biology* **20,** 1–17 (2019).

128. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nature methods* **15,** 1053–1058 (2018).

129. Hie, B., Bryson, B. & Berger, B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. *Nature biotechnology* **37,** 685–691 (2019).

130. Polański, K. *et al.* BBKNN: fast batch alignment of single cell transcriptomes. *Bioinformatics* **36,** 964–965. ISSN: 1367-4803 (Aug. 2019).

131. Korsunsky, I. *et al.* Fast, sensitive, and flexible integration of single cell data with Harmony. *Biorxiv,* 461954 (2018).

132. Lin, Y. *et al.* scMerge: Integration of multiple single-cell transcriptomics datasets leveraging stable expression and pseudo-replication. *bioRxiv,* 393280 (2018).

133. Liu, J. *et al.* Jointly defining cell types from multiple single-cell datasets using LIGER. *Nature Protocols* **15,** 3632–3662 (2020).

134. Haghverdi, L., Lun, A. T., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology* **36,** 421–427 (2018).

135. Luecken, M. D. *et al.* Benchmarking atlas-level data integration in single-cell genomics. *BioRxiv* (2020).

136. Xu, C. *et al.* Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular systems biology* **17,** e9620 (2021).

137. Svensson, V., Gayoso, A., Yosef, N. & Pachter, L. Interpretable factor models of single-cell RNA-seq via variational autoencoders. *Bioinformatics* **36,** 3418–3421. ISSN: 1367-4803 (Mar. 2020).

138. Lotfollahi, M. *et al.* Query to reference single-cell integration with transfer learning. *bioRxiv* (2020).

139. Boufea, K., Seth, S. & Batada, N. N. scID uses discriminant analysis to identify transcriptionally equivalent cell types across single-cell RNA-seq data with batch effect. *Iscience* **23,** 100914 (2020).

140. Akira, C., Loredana, M., Emmanuelle, S. & Antonio, R. Cell-ID: gene signature extraction and cell identity recognition at individual cell level. *bioRxiv.* doi:10.1101/2020.07.23.215525. eprint: https://www.biorxiv.org/content/early/2020/08/11/2020.07.23.215525.full.pdf. https://www.biorxiv.org/content/early/2020/08/11/2020.07.23.215525 (2020).

141. Zhang, A. W. *et al.* Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling. *Nature methods* **16,** 1007–1015 (2019).

142. Zhang, Z. *et al.* SCINA: a semi-supervised subtyping algorithm of single cells and bulk samples. *Genes* **10,** 531 (2019).

143. Guo, H. & Li, J. scSorter: assigning cells to known cell types according to marker genes. *Genome biology* **22,** 1–18 (2021).

144. Lee, J. T. H. & Hemberg, M. Supervised clustering for single-cell analysis. *Nature methods* **16,** 965–966 (2019).

145. Svensson, V. *Actionable scRNA-seq clusters* Mar. 2018. http://www.nxn.se/valent/2018/3/5/actionable-scrna-seq-clusters.

146. Grün, D. Revealing dynamics of gene expression variability in cell state space. *Nature methods* **17,** 45–49 (2020).

147. Smillie, C. S. *et al.* Intra-and inter-cellular rewiring of the human colon during ulcerative colitis. *Cell* **178,** 714–730 (2019).

148. 10x Genomics. *10k Cells from a MALT Tumor - Gene Expression and Cell Surface Protein* Accessed on 19th December 2020. https://support.10xgenomics.com/single-cell-gene-expression/datasets/3.0.0/malt_10k_protein_v3?.

149. Yang, S. *et al.* Decontamination of ambient RNA in single-cell RNA-seq with DecontX. *Genome biology* **21,** 1–15 (2020).

150. Torre, E. *et al.* Rare cell detection by single-cell RNA sequencing as guided by single-molecule RNA FISH. *Cell systems* **6,** 171–179 (2018).

151. Bais, A. S. & Kostka, D. scds: computational annotation of doublets in single-cell RNA sequencing data. *Bioinformatics* **36,** 1150–1158 (2020).

152. Bernstein, N. J. *et al.* Solo: doublet identification in single-cell RNA-Seq via semi-supervised deep learning. *Cell Systems* **11,** 95–101 (2020).

153. DePasquale, E. A. *et al.* DoubletDecon: deconvoluting doublets from single-cell RNA-sequencing data. *Cell reports* **29,** 1718–1727 (2019).

154. McGinnis, C. S., Murrow, L. M. & Gartner, Z. J. DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors. *Cell systems* **8,** 329–337 (2019).

155. Wolock, S. L., Lopez, R. & Klein, A. M. Scrublet: computational identification of cell doublets in single-cell transcriptomic data. *Cell systems* **8,** 281–291 (2019).

156. Hartung, J., Knapp, G. & Sinha, B. K. *Statistical meta-analysis with applications* (John Wiley & Sons, 2011).

157. Wikipedia. *Variance, section 'Basic properties'* Accessed on 24th Nov 2020. https://en.wikipedia.org/wiki/Variance#Properties.

158. Elyada, E. *et al.* Cross-species single-cell analysis of pancreatic ductal adenocarcinoma reveals antigen-presenting cancer-associated fibroblasts. *Cancer discovery* **9,** 1102–1123 (2019).

159. Lavin, Y. *et al.* Innate immune landscape in early lung adenocarcinoma by paired single-cell analyses. *Cell* **169,** 750–765 (2017).

160. Schulz, D. *et al.* Simultaneous multiplexed imaging of mRNA and proteins with subcellular resolution in breast cancer tissue samples by mass cytometry. *Cell systems* **6,** 25–36 (2018).

161. Satija Lab. *Seurat Multimodal vignette, v3.1* https://satijalab.org/seurat/v3.1/multimodal_vignette.html. Accessed: 2020-01-30. Oct. 2019.

162. Mathas, S. *et al.* Aberrantly expressed c-Jun and JunB are a hallmark of Hodgkin lymphoma cells, stimulate proliferation and synergize with NF-κB. *The EMBO journal* **21,** 4104–4113 (2002).

163. Zhang, J. *et al.* The c-Jun and JunB transcription factors facilitate the transit of classical Hodgkin lymphoma tumour cells through G 1. *Scientific reports* **8,** 1–14 (2018).

164. Förster, R., Davalos-Misslitz, A. C. & Rot, A. CCR7 and its ligands: balancing immunity and tolerance. *Nature Reviews Immunology* **8,** 362–371 (2008).

165. Lieberman, Y., Rokach, L. & Shay, T. CaSTLe–classification of single cells by transfer learning: harnessing the power of publicly available single cell RNA sequencing experiments to annotate new experiments. *PloS one* **13,** e0205499 (2018).

166. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome biology* **15,** 1–21 (2014).

167. Ito, K. *et al.* Galectin-1 as a potent target for cancer therapy: role in the tumor microenvironment. *Cancer and Metastasis Reviews* **31,** 763–778 (2012).

168. Perillo, N. L., Pace, K. E., Seilhamer, J. J. & Baum, L. G. Apoptosis of T cells mediated by galectin-1. *Nature* **378,** 736–739 (1995).

169. Ianevski, A., Giri, A. K. & Aittokallio, T. Fully-automated cell-type identification with specific markers extracted from single-cell transcriptomic data. *bioRxiv.* doi:10.1101/812131. eprint: https://www.biorxiv.org/content/early/2019/10/21/812131.full.pdf. https://www.biorxiv.org/content/early/2019/10/21/812131 (2019).

170. Wagner, D. E. & Klein, A. M. Lineage tracing meets single-cell omics: opportunities and challenges. *Nature Reviews Genetics* **21,** 410–427 (2020).

171. Wikipedia. *Occam's razor* Accessed on 12th Apr 2021. https://en.wikipedia.org/wiki/Occam's_razor.

172. Zhang, H. *et al.* Integrated proteogenomic characterization of human high-grade serous ovarian cancer. *Cell* **166,** 755–765 (2016).

173. Angermueller, C. *et al.* Parallel single-cell sequencing links transcriptional and epigenetic heterogeneity. *Nature methods* **13,** 229–232 (2016).

174. Young, M. D. & Behjati, S. SoupX removes ambient RNA contamination from droplet-based single-cell RNA sequencing data. *GigaScience* **9,** giaa151 (2020).

175. Wherry, E. J. & Kurachi, M. Molecular and cellular insights into T cell exhaustion. *Nature Reviews Immunology* **15,** 486–499 (2015).

176. Batson, J., Royer, L. & Webber, J. Molecular Cross-Validation for Single-Cell RNA-seq. *bioRxiv.* doi:10.1101/786269. eprint: https://www.biorxiv.org/content/early/2019/09/30/786269.full.pdf. https://www.biorxiv.org/content/early/2019/09/30/786269 (2019).

177. Grün, D., Kester, L. & Van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. *Nature methods* **11,** 637–640 (2014).

178. Henry, L. *magrittr: A Forward-Pipe Operator for R* Accessed on 14th Apr 2021. https://cran.r-project.org/web/packages/magrittr/index.html.

179. McCarthy, D. J., Campbell, K. R., Lun, A. T. & Wills, Q. F. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics* **33,** 1179–1186 (2017).

Appendix A

Supplement: Garnett marker files

I now paste the content of the three marker files used to train Garnett. This relates to Figure 3.9, where the resulting cell type labels assigned by Garnett are shown.

A.1 Coarse hierarchy

```
>TTN+
expressed: TTN

>LYZ+
expressed: LYZ

>TOP2A+
expressed: TOP2A

>CD3E+
expressed: CD3E

>CD8+
expressed: CD8B
subtype of: CD3E+

>CD4+
expressed: CD4
subtype of: CD3E+

>CD19+
expressed: CD19
```

A.2 Intermediate hierarchy

```
>TTN+
expressed: TTN

>LYZ+
expressed: LYZ

>TOP2A+
expressed: TOP2A

>CD3E+
expressed: CD3E

>CD8+
expressed: CD8B
subtype of: CD3E+

>CD4+
expressed: CD4
subtype of: CD3E+

>FOXP3+
expressed: FOXP3
subtype of: CD4+

>CD4+CXCL13+ T cells
expressed: CXCL13
subtype of: CD4+

>CD4+ANXA1+ T cells
expressed: ANXA1
subtype of: CD4+

>CD19+
expressed: CD19
```

A.3 Fine hierarchy

```
>TTN+
expressed: TTN

>LYZ+
expressed: LYZ

>TOP2A+
expressed: TOP2A

>CD3E+
expressed: CD3E

>CD8+
expressed: CD8B
subtype of: CD3E+

>CD8+ICOS+
expressed: ICOS
subtype of: CD8+

>CD8+CCR7+
expressed: CCR7
subtype of: CD8+

>CD4+
expressed: CD4
subtype of: CD3E+

>FOXP3+
expressed: FOXP3
subtype of: CD4+

>FOXP3+ICOS+
expressed: ICOS
subtype of: FOXP3+

>FOXP3+CCR7+
expressed: CCR7
subtype of: FOXP3+

>CD4+CXCL13+ T cells
expressed: CXCL13
subtype of: CD4+

>CD4+ANXA1+ T cells
expressed: ANXA1
subtype of: CD4+
>CD19+
expressed: CD19

>IGKC+
expressed: IGKC
subtype of: CD19+

>SPN+
expressed: SPN
subtype of: CD19+

>LINC+
expressed: LINC01781
subtype of: CD19+

>IGHD+JUN+
expressed: IGHD, JUN
subtype of: CD19+

>IGHD+JUN-
expressed: IGHD
not expressed: JUN
subtype of: CD19+
```