Accurate integration of single cell transcriptome replicates

OSAKA UNIVERSITY

LOZA LOPEZ MARTIN DE JESUS

Graduate School of Frontier Biosciences
Osaka University

This dissertation is submitted for the degree of
Doctor of Philosophy

March 2022

Abstract

Single-cell RNA-sequencing (scRNA-seq) technologies have revolutionized the study of biological systems by capturing gene expression profiles from thousands of cells in the same experiment. One important application of scRNA-seq data is the comparison of two or more samples to describe genetic changes between conditions, e.g. disease or stimulation conditions. In this kind of analysis, replicated samples allow investigating subtle changes in cells composition, improving the understanding and treatment of conditions. However, these analyses are hindered by *technical differences* of samples known as *batch effects*. Batch effects must be addressed in every joint analysis due to their possible correlation with the main biological components. But this task is not trivial, as non-linearities in technical differences of samples might appear in distinct ways on every experiment. In the last three years, different methods have been developed for the integration of scRNA-seq data, allowing to create cell atlases and to perform joint analyses of datasets. However, these methods could *over-correct*, merging cells from different types. This issue is particularly problematic in the analysis of replicated experiments with small batch effects, where cells with subtle changes in gene expression could be masked, affecting the conclusion of the experiment. To address this problem I designed *Canek*, a bioinformatics tool to integrate scRNA-seq data replicates sequenced with the same technology. Assuming a *linear batch effect within a group of similar cells*, Canek uses *linear estimation* and *fuzzy logic* to obtain cell-specific correction vectors to integrate datasets. Using tests specifically designed to assess over-correction, I show that Canek integrates datasets with the smallest amount of over-correction as compared with state-of-the-art methods. To show how to implement Canek within a workflow, I performed a complete analysis using Canek for the characterization of mouse cells from the spleen. In the same analysis, I show the parameter dependence for commonly used tools in the study of scRNA-seq data, which will serve as a comprehensive guide for other researchers. Canek is computationally efficient and can integrate thousands of cells without over-correction, a special characteristic that could lead future experiments to the improvement and/or the design of *gene-specific treatments*.

Acknowledgments

I would like to thank Diego Diez, thanks to him I came to Japan and joined the Osaka University to pursue my doctoral degree. He introduced me to the amazing world of single-cell, and for that, I would be always thankful. Diego always treated me like a collaborator instead of a student, and because of that, I felt free to explore different areas of my interest which eventually led to the development of this dissertation. Thank you for your patience and for believing in this Mexican with no outstanding background but with a great research spirit. I want to thank Shunsuke Teraguchi, a great theoretical research who had a strong impact on this research work. His advice and criticism helped me to perform robust experiments with a well-defined background and purpose. I am profoundly grateful to Daron Standley ”sensei”, a great researcher who had a strong influence on my development as a doctoral student. His advice helped me to follow my research path without losing the big picture of life. Thank you Daron sensei for hosting this weird guy and making me feel like another member of your laboratory. My sincere thanks to Katoh sensei and Songling, even though we didn't talk so much, your daily hard work motivated me to give my best every day. Thank you to all my friends and colleges in the laboratory; my senpai Sun, whose experience motivated me to continue my path as a ”dry researcher”, Aybars, my beer and coffee buddy, Xu, whose dedication inspires everyone in this lab, Floris, the perfect example of research-life balance, and all the other experienced and new members of the systems immunology lab, who always made me feel welcomed.

I want to thank the Ministry of Education, Culture, Sports, Science, and Technology (MEXT) of Japan for the economical support during my doctoral studies. Thank you for the opportunity and helped me to make true this research project. I would like to thank all current and previous staff members of the Immunology Frontier Research Center (IFReC), whose assistance was essential in the language and paperwork adventure during my time in this research center. My research would have been impossible without Machi, whose support was vital to endure all the unknown procedures of living and researching in Japan; thank you for all your patience with this gaijin. My sincere thanks to the staff of the Graduate School of Frontier Biosciences (FBS), whose aid was crucial to finishing my doctoral studies at this department. Special thanks to Hosotani san for all your support and guidance; thank you for all your efforts to connect international students with the Japanese culture.

I am profoundly grateful to my family. To my parents, for taking care of me even the long physical distance and for supporting me since the beginning of my research career. To

my brother and my sister, who encouraged me to search new opportunities every day. To Yotzan, who many times took care of me even though I am the one supposed to parent him. Special thanks to Shiho chan, who gave me invaluable help to merge with Japanese culture and to build my personal life while constructing my life as a researcher. Thank you to my friends in Mexico, who always made me feel proud to be Mexican and made me feel welcome in every visit to my country. Thank you to my friends in Japan, for all the good times and nice chats. Finally, thank you to all the staff working at Osaka University who indirectly helped me in my daily life at this university.

Muchas gracias.

ありがとうございます。

Contents

1

Introduction

Single-cell RNA sequencing (scRNA-seq) data allows the analysis of cell populations from one or more samples with high-resolution [52]. Figure 1.1 shows a typical workflow to perform a *single-batch* scRNA-seq analysis. Currently, there are more than 1,000 bioinformatics tools designed to analyze scRNA-seq data [65], which can be classified as tools for *pre-processing* or tools for *downstream* analysis. The main goal of the pre-processing steps is to prepare the

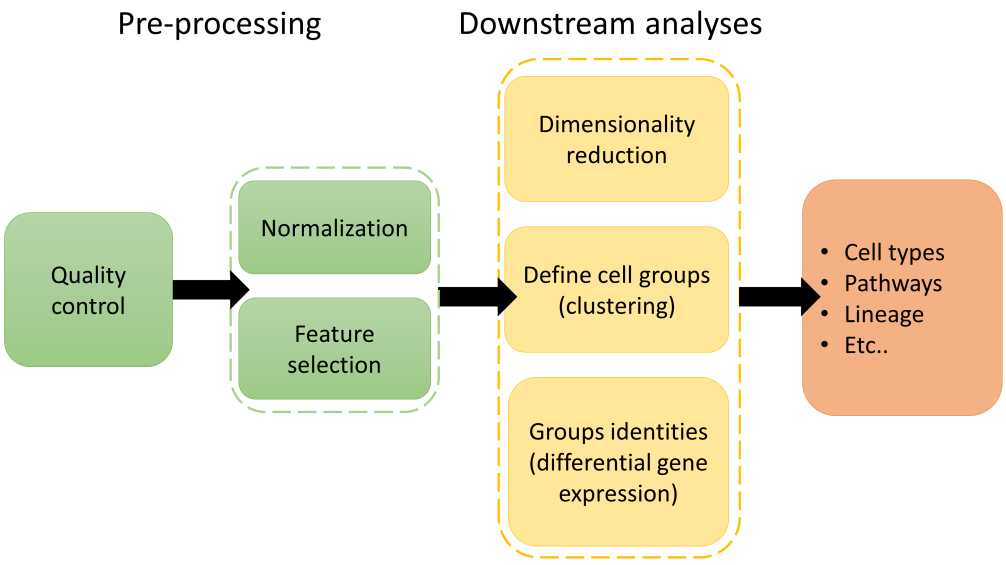

Figure 1.1: Single batch scRNA-seq analysis workflow. The workflow starts with a series of pre-processing steps as quality control, normalization, and features selection. Then, different downstream analyses can be performed to define differences and similarities of samples.

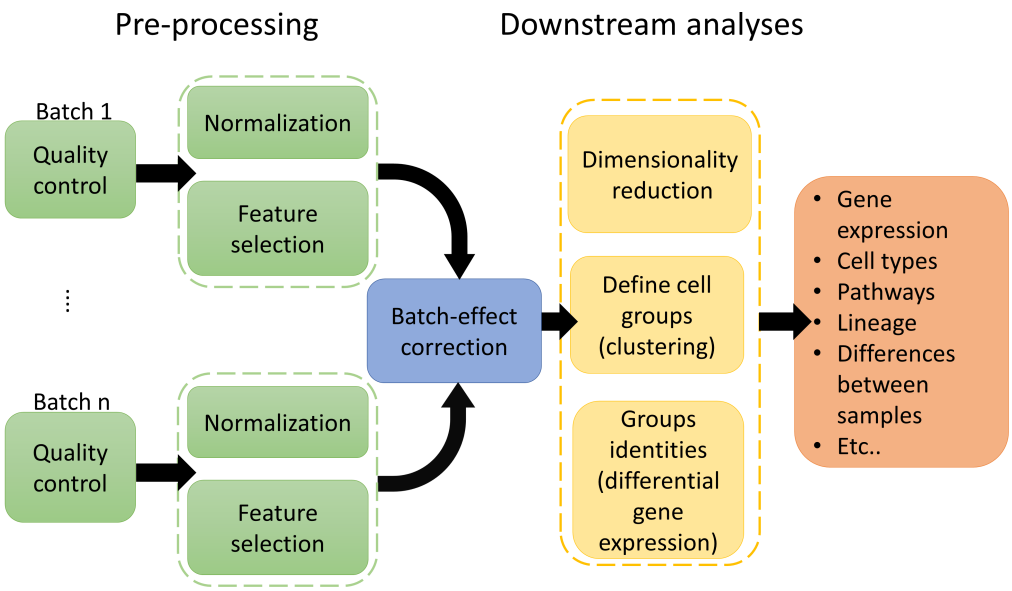

Figure 1.2: Multiple batch scRNA-seq analysis workflow. The pre-processing of each batch is independently performed and an integrative step to correct technical differences between batches is implemented. Then, downstream analysis can be made to investigate the biology of the integrated batches.

data for downstream analysis. The most common pre-processing steps are: *quality control* to filter low-quality cells, *normalization* to remove technical differences of cells, and *feature selection* to select high variable genes. Downstream analysis aims to investigate the biology of samples by comparing the gene expression of cells. *Dimensionality reduction*, *clustering*, and *differential gene expression* are three of the most common steps used in this kind of analysis. These three techniques are often used as part of more complex analyses. e.g. the delineation of the differentiation process of cells within samples. Together, pre-processing and downstream analysis allow the investigation of the underlying biology of thousands of cells at the gene level, an extraordinary improvement as compared wit previous technologies.

Figure 1.2 shows the workflow of a typical *multi-batch* analysis. In this workflow, each batch is independently pre-processed, and a process known as *batch effect correction* is performed to integrate the batches. The integrated batch can be used in downstream analyses with different purposes, like the creation of *cell atlases* [53], the investigation of similarities and/or differences among samples with *distinct conditions* (e.g. wild type and knock-out conditions), the exploration and comparison of *different species*, etc. From the workflow in Figure 1.2, it can be observed that a very important step in a *multi-batch* analysis is batch effect correction, as the results of this step directly affect downstream analyses thus changing the conclusions of the experiment. Batch effects are *technical differences* that appear in every joint analysis due to variations on the preparation and sequencing of samples. Batch effects must be properly addressed in every *multi-batch* analysis [31, 16]. If batch effects are not considered, due to their possible *correlation with biological variables of interest*, they could lead to erroneous conclusions induced by batch differences rather than biological insights of cells. On the other hand, an *over-correction* could mask small populations of cells that could

be key for the conclusions of experiments. Two popular methods to correct batch effects are the one developed at Marioni's laboratory, *MNN* (Mutual Nearest Neighbors) [18], and the one included in the R package *Seurat* [51]. These two methods have the advantage of being easy to implement and presenting simple-to-understand results. Other popular methods are *Harmony* and *Liger* [29, 63], which have the advantage of integrating datasets by using their low-dimensional representation (see section 2.2), an important feature for the integration of dissimilar batches prepared with different protocols. These state-of-the-art methods focus on the integration of batches regardless of their nature, an important feature to correct strong technical differences of datasets (e.g. different technologies), an important feature to create cell atlases, or to compare samples from different species. However, due to their fundamental objective of finding a global integration, they could mask biological differences due to *over-correction* [67]. Over-correction is a significant problem when small changes are expected between samples, e.g. experimental replicates. In this scenario, the methods before mentioned could mask sample differences that could be key for the conclusions of the experiment. Therefore, *methods that integrate scRNA-seq datasets while preserving the underlying structures of cells are essential for the unbiased interpretation of multi-batch analyses*.

As part of this dissertation, I designed *Canek*, a batch effect correction method to integrate *scRNA-seq data replicates sequenced with the same technology*. In the following chapters, I show how Canek can correct technical differences of batches while introducing the least amount of over-correction as compared with state-of-the-art methods. The dissertation is divided as follows. In part I, I introduce the main methods used in the *multi-batch* analysis of scRNA-seq data and in Canek's workflow to correct batch effects. The first chapter of part II, presents the results of a benchmark performed with Canek and state-of-the-art integration techniques. In the same part, I performed an example of *multi-batch* analysis using Canek to characterize spleen cells. In chapters 6 and 7, I discuss the main advantages and limitations of Canek and *multi-batch* analysis. Appendices A and B include supplementary tables and figures from chapters 4 and 5, respectively. The list of my academic accomplishments during my doctoral studies and the references used in this dissertation can be found at the end of the document.

Part I

Methods

2

Single-cell RNA-sequencing analysis

Single-cell RNA sequencing (scRNA-seq) analysis involves a series of steps systematically applied one after the other (Figure 1.1 and 1.2). Currently, there are available numerous bioinformatics tools that can be implemented on each of the steps. Depending on the data, these tools could give different solutions with distinct advantages and disadvantages. Understanding the fundamental objectives of each step helps to decide which tools are appropriate to implement in agreement with the objectives of the experiment. This chapter introduces the main steps and tools for scRNA-seq analysis to establish a basic background for further chapters. This chapter is divided into five sections: *pre-processing, dimensionality reduction, clustering, differential gene expression analysis*, and *batch effects correction*.

2.1 Pre-processing

The objective of the pre-processing steps is to prepare the data for more complex analyses. Typically, data pre-processing involves *quality control* of cells, *normalization* of cell and gene counts, and the selection of high variable genes (also called *feature selection*).

2.1.1 Quality Control

Quality control (QC) aims to filter low-quality cells based on three features (also referred as metrics) from each cell [36, 2, 23]:

- *Number of counts*. Low-quality cells commonly exhibit a lower number of reads. Also, when two or more cells are captured together (doublets), their gene reads presents

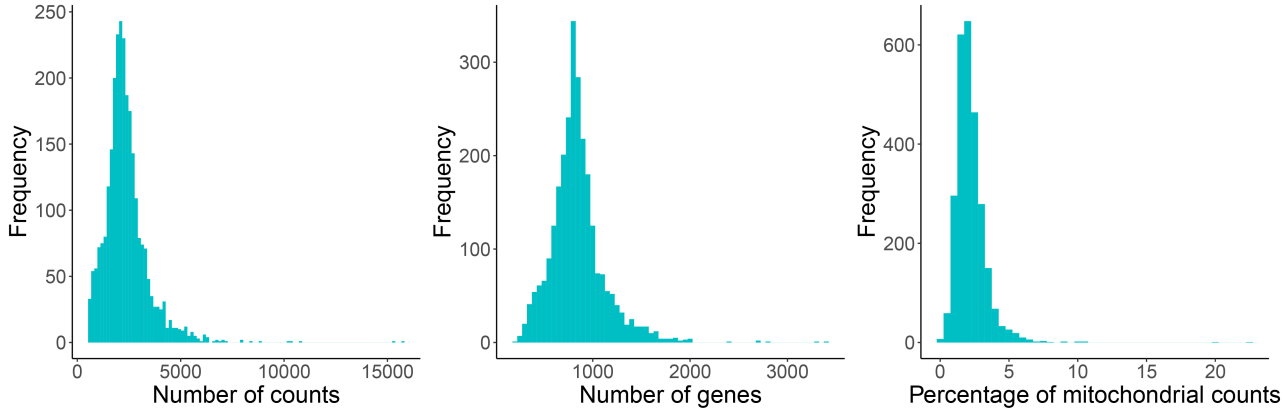

Figure 2.1.1: QC metrics distribution. QC distributions of Peripheral Blood Mononuclear Cells (PBMC). Distributions are relatively normal with outliers.

”excessive” number of counts.

- *Number of genes*. Due to deficiencies during sequencing or library preparation, low-quality cells would present a low number of genes with non-zero reads.

- *Percentage of mitochondrial counts*. Dying cells contain higher proportions of reads from mitochondrial genes.

QC involves setting filtering thresholds to discard cells with *low number or reads*, *low number of genes*, *high number of counts*, *high number of genes*, and *high percentage of mitochondrial counts*. To do so, it is helpful to study the distribution of the three metrics previously described. Figure 2.1.1 shows and example of the distribution of the metrics for a freely available dataset composed by 2700 Peripheral Blood Mononuclear Cells (PBMC) from 10x Genomics. It can be observed that the metrics are relatively normally distributed with outliers. For example, those outlier cells containing more than 15,000 number of counts (Figure 2.1.1 left). Because these three metrics are often correlated with each other, it is useful to set filtering thresholds by inspecting *combined plots*. Figure 2.1.2 shows the interaction of the three metrics from Figure 2.1.1. There is a clear correlation between metrics, where a low number of counts corresponds with a low number of genes and a high percentage of mitochondrial counts. The red dashed lines illustrate *joint thresholds* examples where using only two limiting numbers, those cells with undesired values for the three QC metrics can be filtered.

2.1.2 Normalization

Single-cell RNA sequencing, like any other technology that transforms environmental abundances into quantitative measurements, suffers from technical differences unrelated to the measured process [48]. For instance, the number of counts for each cell in a dataset may extensively differ (e.g. on Figure 2.1.1 left plot, the counts vary from hundreds to thousands).

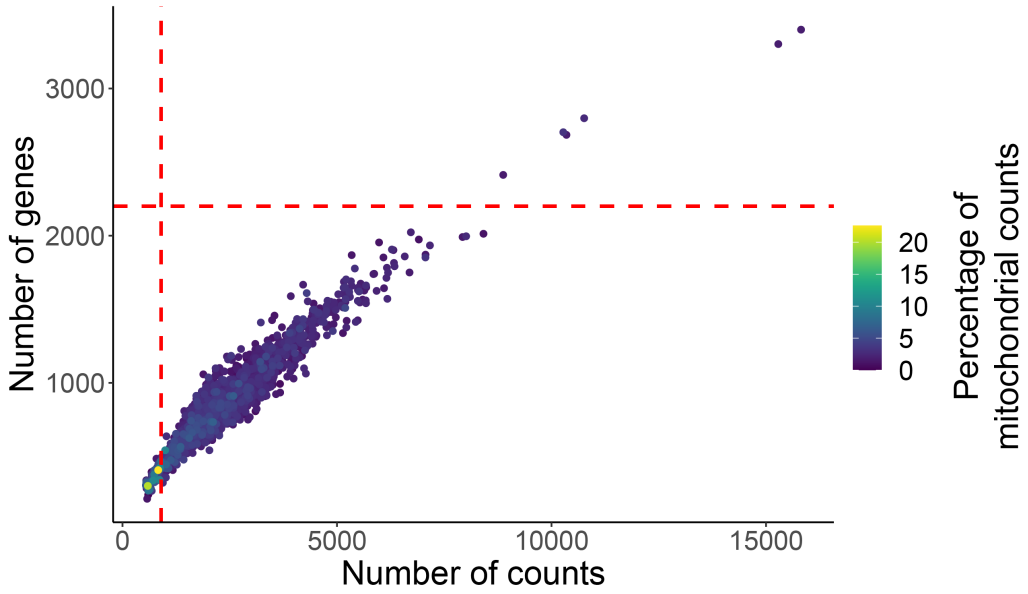

Figure 2.1.2: QC combined plot. It is useful to set filtering thresholds by inspecting a combined plot of the three QC metrics from 2.1.1. There is a correlation between metrics, where a low number of counts corresponds with a low number of genes and a high percentage of mitochondrial counts. Red dashed lines illustrate examples of joint thresholds.

Normalization aims to remove technical differences to obtain comparable measurements between cells and/or genes.

In this thesis, I distinct between normalization and batch effect correction (see section 2.5 and Figure 1.2), where the latter aims to remove technical differences between batches, while cell and/or gene normalization occurs independently of the batch structure [2, 36].

Cell counts normalization.

There exist different linear and non-linear approaches to normalize scRNA-seq data [37, 10]. I will focus on a *global-scaling* normalization method popularized by the R package Seurat[8, 55]. This method assumes that all cells in a dataset contain the same number of counts, and their differences arise from technical variations (e.g. sequencing depth). On this method, each gene from a cell is divided by a *cell-specific scaling factor* [3], obtained by dividing the total counts of each cell by a constant (10,000 in case of Seurat normalization).

Typically, after normalization the resulting counts are $log(x+1)$ transformed. This *logarithmic transformation* (also referred to as log transformation) is useful in further comparison of cells, as distances between log-transformed values represent *log fold changes*, a common way to compare biological samples. Another advantage of transforming the data is the reduction in the skewness of genes distribution that typically appears on scRNA-seq data (Figure 2.1.3). For example, in Figure 2.1.3-left, the distribution is skewed towards the left side of the plot before the log transformation. After transformation (Figure 2.1.3 right), the distribution is approximately normal, a convenient feature for further analysis.

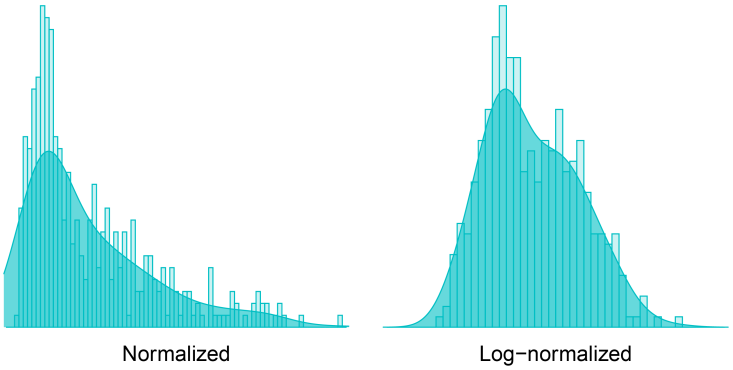

Figure 2.1.3: Effect of the log transformation. Log transforming the normalized data reduces the skewness in the distribution of the genes. Log-normalized data distribution is approximately normal, a useful characteristic for further analysis.

Gene counts normalization.

Gene counts normalization, also referred to as *scaling*, aims to normalize the variability of counts to improve the comparison of genes. Scaling gene counts ensures that all genes are equally weighted, e.g. all genes have mean zero and standard deviation one. This results on the improvement of analysis based on variability, as the principal components analysis (PCA) [42] (see section 2.2) which aims to find a coordinate transformation based on variance maximization.

While gene scaling improves analysis as PCA, for other kind of analysis, the expression magnitude serves as a factor of the importance of the gene [36] (e.g. feature selection. See 2.1.3). In such cases, it is preferred to use the counts without scaling.

2.1.3 Feature selection

Feature selection (FS) refers to the process of sub-setting representative genes (e.g. highly variable genes) to reduce the impact of non-informative noisy genes in downstream analysis (e.g. dimensionality reduction in section 2.2). Moreover, FS reduces the computational effort of scRNA-seq analysis methods by reducing the dimensionality of the data [7].

In this thesis, I focus on the FS method introduced by Stuart et al. (2019) [51], which selects genes that exhibit a high cell-to-cell variation using a *variance-stabilizing transformation* (VST) [39]. VST fits a polynomial model of the *mean-variance* relation of genes using the un-normalized counts (also called raw counts), and transforms them as:

$$z_{ij} = \frac{x_{ij} - \bar{x}_i}{\sigma_i}, \tag{2.1}$$

where z_{ij} and x_{ij} are respectively the standardized value and the un-normalized count of gene i in cell j. $\bar{x}_i$ is the mean value of the raw data of gene i, and σ_i is the predicted variance of gene i. VST computes the variance of the transformed values z_{ij} and selects a set of highly variable genes as those with the highest standardize variance. Figure 2.1.4

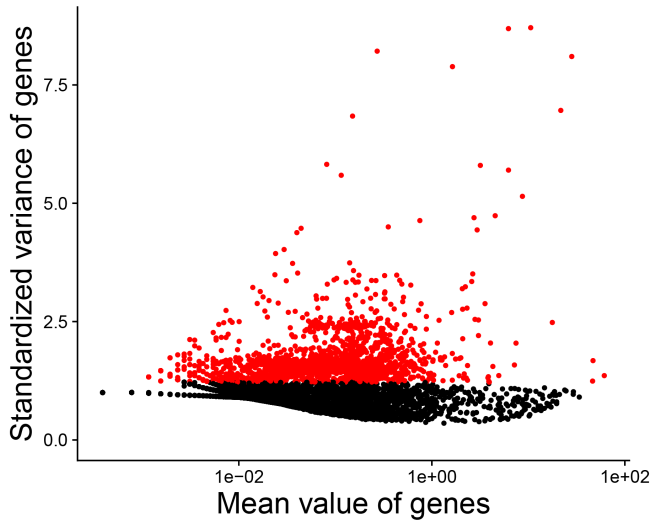

Figure 2.1.4: Mean-standardize variance of genes using the variance-stabilizing transformation (VST). After transforming the raw counts using the predicted variance σ and the mean value $\hat{x}$, VST selects highly variable genes from those with the highest standardize variance.

show an example of the mean-standardize variance plot, where red dots depicts the top 2,000 genes with the highest standardized variance.

2.2 Dimensionality reduction

One of the main goals of single cell RNA sequencing data analysis is to compare samples using their gene expression profiles. In this context, one cell is treated as a sample, and its gene expression data as a *high dimensional* profile. The high dimensional concept comes from treating each gene as an added dimension for each sample. Figure 2.2.5 shows a simple example of this cell-sample, gene-dimension analogy, where two-dimensional gene expression profiles of four samples are represented in a coordinate system. This representation ease the comparison between samples, as the Euclidean distance of cells in the plot can be used as a similarity metric, where more similar cells are closer in the graph (e.g. Figure 2.2.5 cell 1 and cell 3 are more related as compared with the other cells). Unfortunately, there is no a straight way to plot the high-dimensional data from scRNA-seq experiments in a coordinated system. However, genes with high correlated expression levels can be grouped and represented by their fundamental *components* in an *embedded representation* [20, 57]. This process is known as *dimensionality reduction*, and it can be used to *summarize* and/or *visualize* the data [36].

2.2.1 Summarizing the data

Summarizing methods aims to capture the *inherent variability* of the data in a low-dimensional representation, where low-dimensional and high-dimensional distances of cells are related. The low-dimensional coordinates of cells can be used in downstream analysis, e.g. clustering

	Gene 1	Gene 2
Cell 1	1	2
Cell 2	4	1
Cell 3	1	3
Cell 4	4	2

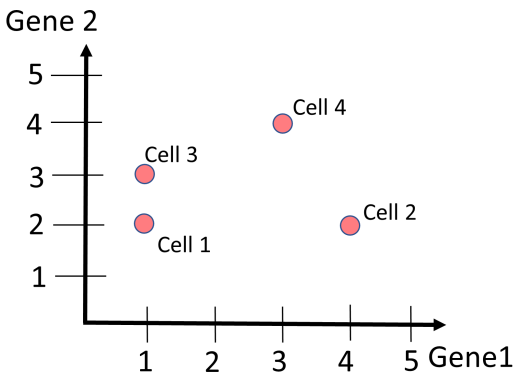

Figure 2.2.5: Example of the gene-dimension relation in scRNA-seq data. Each gene from a scRNA-seq data is treated as an added dimension. Thus, cells can be represented in a coordinate system, and their Euclidean distances can be used as a similarity metric, where more similar cells are closer to each other.

(see section 2.3), reducing noise in the data and reducing computational effort [2]. The most popular method to summarize scRNA-seq data is principal components analysis (PCA) [42], which captures most of the variance of the data by a subset of the top principal components (PC). Choosing the number of PC is not trivial and mostly depends on the structure of the data. The simplest heuristic way is to look at the captured variance of the top PCs in an *elbow plot* (Figure 2.2.6 up-left). The elbow plot shows that the variance of the PC rapidly decreases until certain breaking point from which it remains almost constant, creating a curve similar to an elbow. A good estimation of the number of PCs would be the breaking point, also known as *elbow point*. Figure 2.2.6 up-left shows an example of the elbow plot with the elbow point highlighted in red.

2.2.2 Visualizing the data

PCA summarizes the data with just a few dimensions and is the basis of many currently available analysis tools [36]. However, the problem of visualizing the data in a coordinate systems might remain, as most scRNA-seq data analysis require more than 3 PCs to represent them. For this purpose, *non-linear* dimensionality reduction methods are preferred. Visualization methods aim to capture the structure of the data *without the restriction of preserving comparable sample distances* between the low-dimensional representation and the high-dimensional space. Such characteristic makes them suitable for data visualization, but their results *should not be used in downstream analysis* that compares or groups cells based on distance [36]. Two popular visualization methods are the t-distributed Stochastic Neighbour Embbeding (t-SNE) and the Uniform Approximation and Projection (UMAP) methods [40, 59]. Both methods share the objective of *neighborhood preservation* between the low-dimensional representation and the original data, but they focus on different properties. While

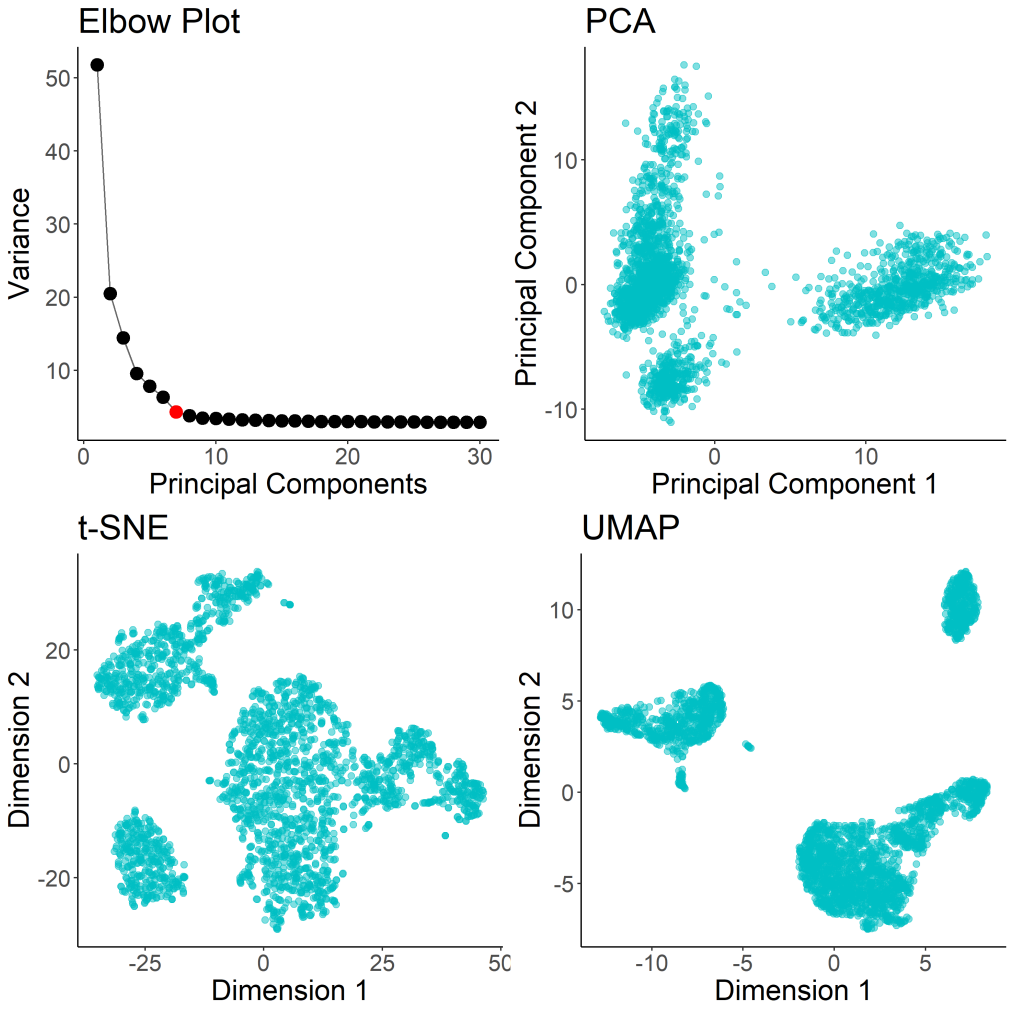

Figure 2.2.6: Dimensionality reduction. The elbow plot is a simple technique to estimate the inherent dimensionality of the data (top-left plot). In this plot, the variance of the top PCs rapidly decrease until a breaking point (also called elbow point) from which it remains almost constant. The elbow point, exemplified by a red point in the upper left figure, can be used as an estimation of the number of PCs. PCA can be used to summarize and visualize the data (top-right plot). However, non-linear techniques, as t-SNE and UMAP in bottom plots, are preferred for visualization as they better capture groups within the data.

t-SNE aims to represent the data from an statistical perspective, UMAP does it from a topological one. The bottom and up-right plots in Figure 2.2.6 show examples of PCA, t-SNE and UMAP visualizations for the 2700 PBMC cells previously used in section 2.1.1. Groups of cells are better delineated on t-SNE and UMAP representations as compared with the first two PCs, a useful characteristics to visualize clustering results (see next section).

2.3 Clustering

In computational analysis, clustering refers to the process of defining *sub-groups* within a dataset [24]. The definition of a sub-group or cluster depends on the context of the data and the objectives of the analysis [12]. In scRNA-seq data analysis, clustering is an *unsupervised*

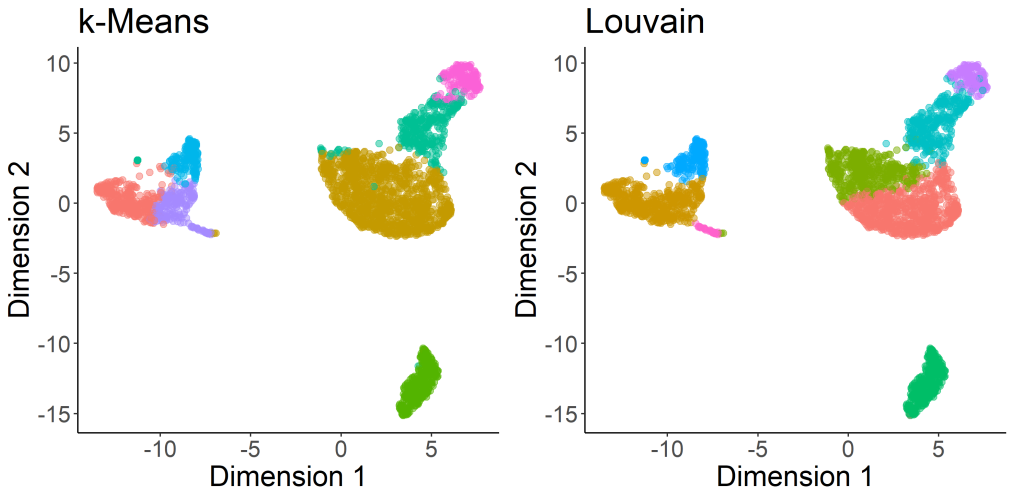

Figure 2.3.7: Clustering with k-Means and Louvain methods. Left plot shows the k-Means clustering results ($k = 7$) in the UMAP representation of 2,700 PBMC cells. Cells from the same cluster are colored equally. The right plot shows the results of the Louvain clustering with a resolution of 0.5. The community detection algorithm formed 8 clusters.

learning procedure to characterize *groups of cells with similar expression profiles*, where cell similarities are often defined by Euclidean distances in the PCA space. Clustering allows to assign discrete and easy-to-understand *labels* to describe cells heterogeneity [2].

One popular clustering method is k-means, which aims to partition the data into k groups of cells [38] (see Figure see Figure 2.3.7). k-means starts by randomly defining k cluster's centroids and obtaining the *cell-to-centroid* distance for each cell. Then, cells are assigned to the closest cluster and centroids are recalculated. The assignment of cells and recalculation of centroids is repeated until convergence. One disadvantage of k-means is that the number of clusters k needs to be decided in advance. The selection is usually performed heuristically by looking at cells embeddings, but some methods estimate the number of clusters using iterative and/or statistical approaches [45, 27, 49, 62]. However, the simplicity and fast implementation of k-means make it a great choice for exploratory analysis.

Another popular clustering method is the *Louvain clustering* [6] (Figure 2.3.7 right). Louvain clustering is based on a k-Nearest Neighbour (kNN) [13] graph representation of the data (do not confuse with the k from k-Means), where cells are represented as nodes in the graph [36]. In this graph, usually obtained in the PCA space, each cell is connected to the k closest cells, and edges are weighted based on cell distance similarities. Then, a *community* detection algorithm is applied to identify dense regions of the graph. Each community can be used as a cluster on downstream analysis.

There isn't a *gold method* for clustering in scRNA-seq analysis [36]. Each method has advantages and disadvantages, and their results vary in every experiment. Therefore, it is recommended to test different clustering techniques with different parameters before deciding the final clusters of cells. Moreover, clustering results should not be only assessed based on how well they match visualization results, but on how well they represent the heterogeneity of the data. Therefore, the inspection of clusters' underlying gene expression profiles and

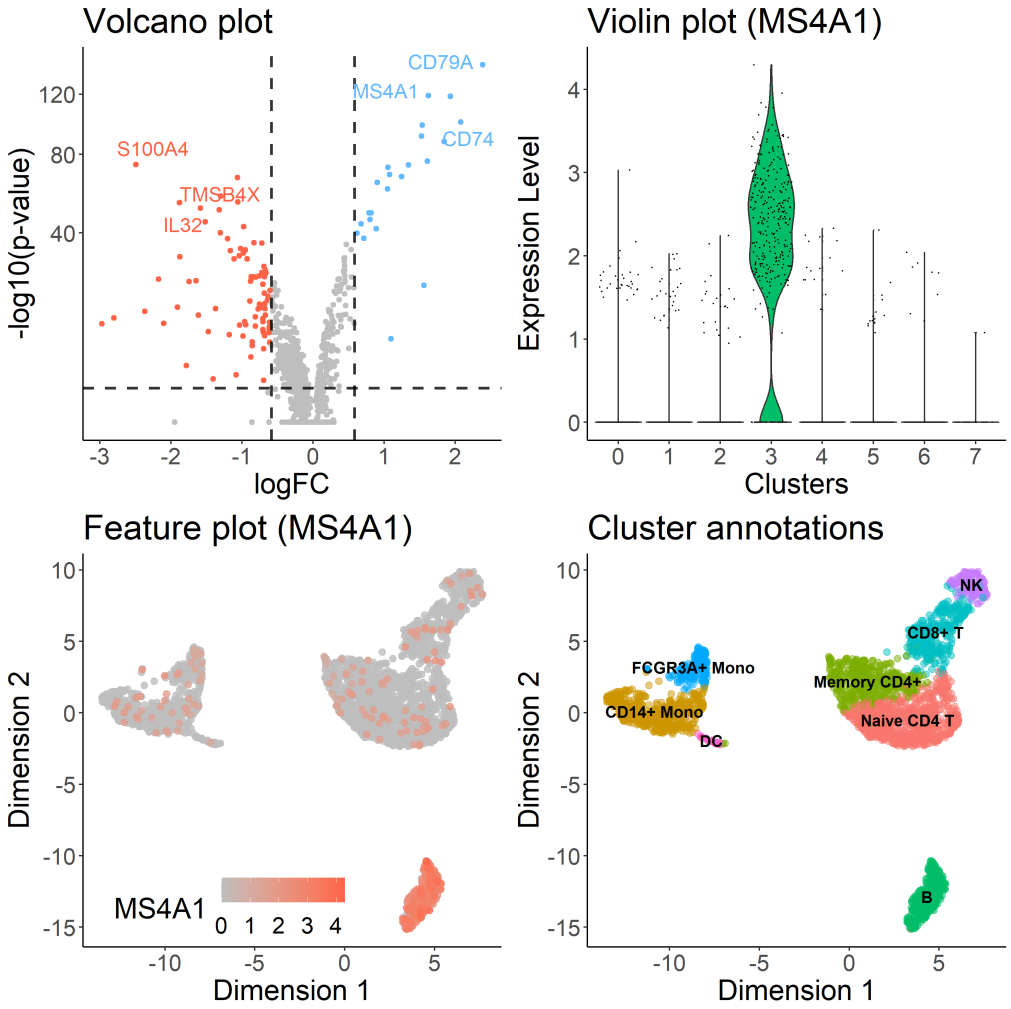

Figure 2.4.8: Differential gene expression (DEG) analysis. Following the Louvain clustering from Figure 2.3.7, the top-left plot displays a volcano plot of the ”one-versus-all” DGE strategy applied on cluster 3. Up and down regulated marker genes are respectively colored in blue and red. Some of the highly differentiated genes are labelled (e.g. *MS4A1* and *CD79*). Top-right plot shows a violin plot of gene *MS4A1*, where the strong change in the distribution of cluster 3 confirms the selection of this gene as a marker of this cluster. This is also validated with a feature plot (bottom-left plot), where cells from cluster 3, located at the bottom-right side, exhibited high expression levels of *MS4A1*. Cluster annotations (bottom-right plot) can be assigned by comparing cluster markers with gene expression profiles of known cell types. In this case, gene markers of cluster 3 correlates with B cells.

their differences is crucial (see differential gene expression analysis in the next section).

2.4 Differential gene expression analysis

In scRNA-seq analysis, differential gene expression (DGE) is widely used to discover *marker genes* that cause separation between clusters [58]. Such markers are used to annotate cluster with meaningful biological labels, typically associated with known cell types [36, 2]. There are different statistical methods and strategies for DGE [47, 58]. One popular methodology is the ”one-versus-all” comparison popularized by the Seurat R package [8, 55], where the

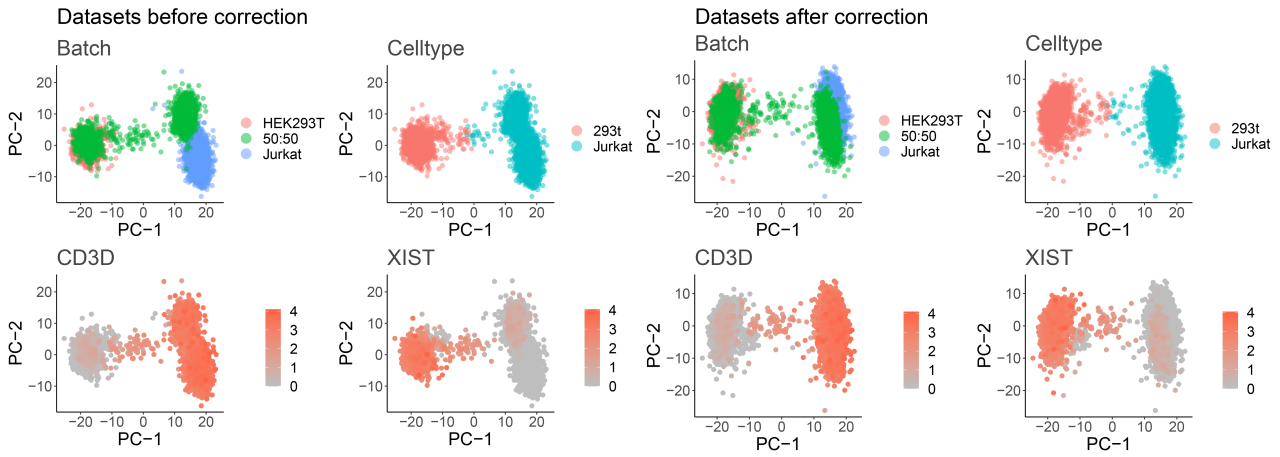

Figure 2.5.9: Batch effects correction. Batch effects are technical differences that must be considered every time we jointly analyze two or more datasets. Before correction (four left plots), three batches, two containing pure Jurkat and HEK293T cells, and one with a 50:50 mix of Jurkat and HEK293T cells, group by batch rather than cell type. This separation is also visible from the expressions of CD3D and XIST genes, two known markers of Jurkat and HEK293T cells, respectively. After batch effect correction (four right plots), cells cluster by cell type allowing joint analysis of biological factors.

gene expression of cells from one cluster are compared with the gene expression of the remaining cells. Then, cluster-specific gene marker are identified with two metrics: a measurement of *gene expression change*, and a measurement of *statistical significance*. This strategy is performed on all the clusters, obtaining a list of differentially expressed genes for each cluster. The results are typically visualized in *volcano plots* (Figure 2.4.8 top-left plot), which help to highlight gene markers based on the tow metrics previously defined. One way to validate the identified markers is by analyzing their distribution in *violin* and *feature plots* (Figure 2.4.8 top-right and bottom-left plots). Finally, gene markers of cluster can be compared with expression profiles of known cell types to *annotate* clusters with biological labels (Figure 2.4.8 bottom-right plot).

2.5 Batch effects correction

A key step in every multi-batch analysis is the *merging* of batches for their *joint analysis* in further steps (Figure 1.2). After merging the batches, it is common to observe differences in cells that are independent of the biology of the samples. For example, cells from the same type but sequenced separately may appear as independent groups in the UMAP visualization. These *technical differences* are known as *batch effects*, and they appear in every multi-batch analysis due to variations during handling and sequencing of samples [31, 36]. Batch effects must be addressed in every joint scRNA-seq analysis because of their possible correlation with biological insights [56]. Figure 2.5.9 exemplifies the batch effects in the joint analysis of three datasets [68]: the first batch contains only 293T Human Embryonic Kidney

cells (HEK293T cells, identified by the expression of XIST gene), the second batch contains only immortalized human T lymphocytes cells (Jurkat cells, identified by the expression of CD3D gene), and the third batch is a mix of 50:50 293T and Jurkat cells. The four left plots in Figure 2.5.9 show the distribution of cells by batch and cell type, and the feature plots of cell type markers. After merging the batches, because the biology of samples is known, two groups of cells are expected: one from HEK293T cells and one from Jurkat cells. However, while HEK293T cells are well mixed between batches, Jurkat cells formed two groups, one from the Jurkat batch and one from the 50:50 batch. Because Jurkat cells from different batches are grouped together, the deviation is attributed to batch effects. There are different methods developed to correct batch effects in scRNA-seq data (see next section) [25, 18, 51, 22, 43, 33, 66, 63, 34]. The four right plots in Figure 2.5.9 show the distribution of batches after batch effect correction. Cells from different batches are well mixed by cell type, and batch differences on Jurkat cells are almost imperceptible. This is confirmed by feature plots, where cells expressing CD3D (HEKT293T cells) and XIST (Jurkat cells) formed cell-specific unique groups.

2.5.1 State-of-the-art methods

Batch effects are not a problem specific to scRNA-seq data. Before single-cell technologies, gene expression of samples was captured using microarrays. In this kind of experiment, cells were sequenced in ”bulk” obtaining only one read for each gene from one sample. A popular method to correct batch effects in microarray data is ComBat [25], which integrates the data by estimating the batch influence in mean and variance of samples. ComBat can also be applied on scRNA-seq data, however, its correction might not be sufficient because of the complex structures of cells in this kind of experiments [36]. One of the first methods developed to correct batch effects on scRNA-seq data is the *Mutual Nearest Neighbors (MNN) correction* [18]. This method corrects batch effects from a pair of batches by calculating correction vectors using MNN pairs (see section 3.2.1). The vectors are used to correct the cells from one of the batches, matching the cells from the other batch. A considerable problem with this methodology is the large computational effort to calculate the MNN pairs and correction vectors. To address this issue, the authors launched a fast version of this algorithm called *fastMNN*, which corrects batch effects from the principal components (PC) space, considerably reducing running time. Another popular method is the one implemented in the *Seurat* R package [51]. Seurat corrects batch effects using correction vectors to match cells from a pair of batches. The correction vectors, obtained using MNN pairs in the canonical correlation analysis (CCA) space, are weighted among neighboring cells to avoid the mixing of dissimilar cells. Another interesting method is *Harmony* [29], which corrects batch effects by joining cells using *soft-clustering* in an iterative way. The correction is performed in the PCA space, which reduces computational effort, getting a merged dataset in a short time. Another method that corrects batch effects from an embedding representation of cells is *LIGER*. LIGER integrates datasets using a *shared factor neighborhood graph* of

cells in a low-dimensional space defined by an *integrative non-negative matrix factorization method*. A different approach to correct batch effects is *Scanorama* [22], which implements a *panorama stitching* algorithm to integrate multiple-batches using MNN pairs. This method is implemented in *Python* [60] and can be used to correct batch effects from the PCA representation cells or the original expression of genes.

Different benchmarks have been performed to assess batch effects correction methods in scRNA-seq data. One of the first comparisons was made by Butler et al. (2018) [8] to introduce their batch effect correction metric named kBET (k-nearest neighbor batch-effect test). The authors assessed different methods in a set of batches containing cells from the same lineage at different stages of differentiation. The integration methods were implemented using data normalized with different approaches. The ComBat method was distinguished from the rest, obtaining the best mixing and cell type preservation scores. A more comprehensive benchmark of 14 batch correction methods was made by Tran et al. (2020) [56]. On this benchmark, the authors compared integration methods on five scenarios: *similar cells with different technologies, dissimilar cells, multiple batches, big data, and simulated data*. The best methods were Harmony, Liger, and Seurat. However, different methods dominated the others in different scenarios. For example, ComBat poorly integrated *similar cells with different technologies*, but it obtained the highest score by recovering gene expression from *simulated data*. Another batch correction benchmark is the one made by Luecken et al. (2020) [35]. On this benchmark, performed at the atlas level, the authors compared 38 methods using different pre-processing combinations of batches. One of the best-scored methods was Scanorama, which outperformed Seurat in complex integration tasks.

2.5.2 Potential problems

From the previous subsection, it can be observed that the performance of batch effect correction methods depends on the nature of the data, with no best method to use in every scenario. Therefore, the *characterization of integration methods* for specific tasks is necessary. Current state-of-the-art methods for batch effect correction focus on solving batch differences regardless of their nature (e.g. different technologies), an important feature to create cell atlases or to compare samples from different species. However, due to their fundamental objective of finding a global integration, they could mask biological differences due to *over-correction* [67]. Over-correction is a significant problem when small changes are expected between samples. In this scenario, over-correction could mask sample differences that could be key for the conclusions of the experiment. To exemplify this problem, the three datasets introduced in section 2.5 (Jurkat, HEK293T, and 50:50 Jurkat:HEK293T cells [68]) were integrated using two correction methods. Figure 2.5.10 shows the datasets before and after integration. While one method led to an *accurate correction* of batches, reducing technical differences in Jurkat cells (Figure 2.5.10 center), the other one *over-corrected*, creating a third group of cells containing a mix of Jurkat and HEK203T cells (Figure 2.5.10 right). If this incorrectly integrated dataset were used in downstream analysis, it would lead to the

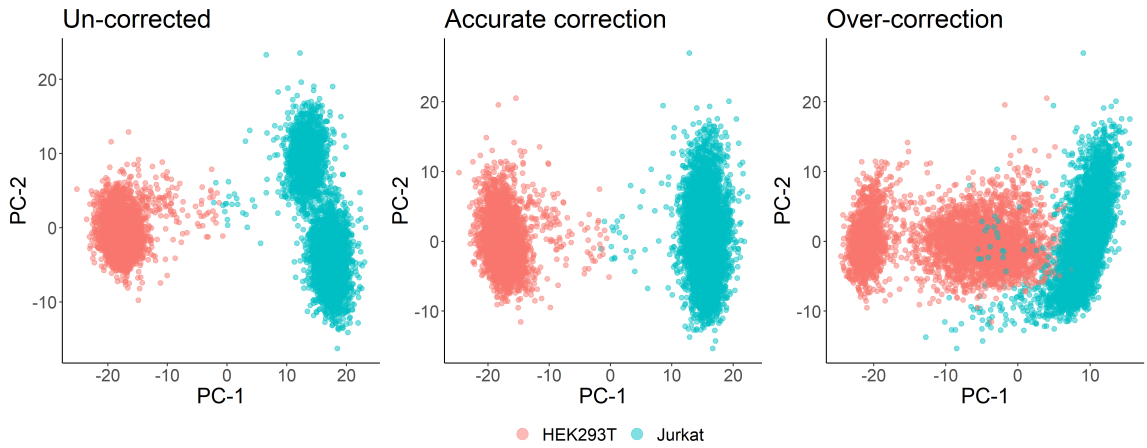

Figure 2.5.10: Over-correction on batch effect correction. One common problem in batch effects correction is *over-correction*. Over-correction masks biological differences which could be key for the conclusions of the experiment. The left plot shows batch effects in the Jurkat, HEK293T example presented in section 2.5. An accurate correction, shown in the center plot, minimized technical differences of samples recovering cell types. However, an over-corrected integration, shown in the right plot, merged different cell types thus affecting further analysis and conclusions.

bias characterization of three cell populations, negatively affecting the conclusions of the experiment.

Over-correction is a common problem in scRNA-seq analysis, however, it hasn't been properly analyzed by state-of-the-art methods and benchmarks. To address these issues, I designed a batch effects correction method for scRNA-seq data replicates and benchmarked it against state-of-the-art methods in tests specifically designed to assess over-correction. The method is named *Canek*, and it's implemented as an R package which is currently available from Github [15]. The next chapter introduces the methodology used in Canek and other useful tools to assess batch effect correction.

3
Canek

This chapter introduces *Canek*, a bioinformatics tool to correct batch effects in scRNA-seq replicates obtained with the same technology. The chapter is divided in three main sections. The first two sections give an *overview of Canek* and explain *Canek's methods* to correct batch effects. The last section presents *metrics* to assess batch effects correction.

3.1 Overview of Canek

Canek corrects batch effects from a pair of batches, denoted as *reference batch* and *query batch*, by combining a *local linear correction* with a *cell-specific non-linear correction*. When Canek is implemented in more than two batches, a hierarchical approach is iteratively used to select the reference and the query batch until all the batches are integrated (see section 3.2.5). Figure 3.1.1 exemplifies the main steps used in Canek to correct batch effects. In this Figure, batch effects caused similar cells (same color dots) to group by batch (Figure 3.1.1a). To correct batch effects, Canek first defines batch effects observation using mutual nearest neighbors pairs (MNN pairs) [18] and identifies groups of similar cells from the query batch with clustering (Figure 3.1.1b). Then, assuming a *mostly linear batch effect within clusters*, Canek estimates a correction vector for each cluster using the gene expression differences of cells identified by MNN pairs (arrows on Figure 3.1.1c). The correction vectors can be used in a *local-linear correction* approach to correct the cells from the query batch. In this approach, cells from the same cluster share the *same correction vector*. To improve the correction of datasets that might not fulfill the linearity assumption, Canek further smooths the transitions between correction vectors using fuzzy logic (Figure 3.1.1d). Thus, in this

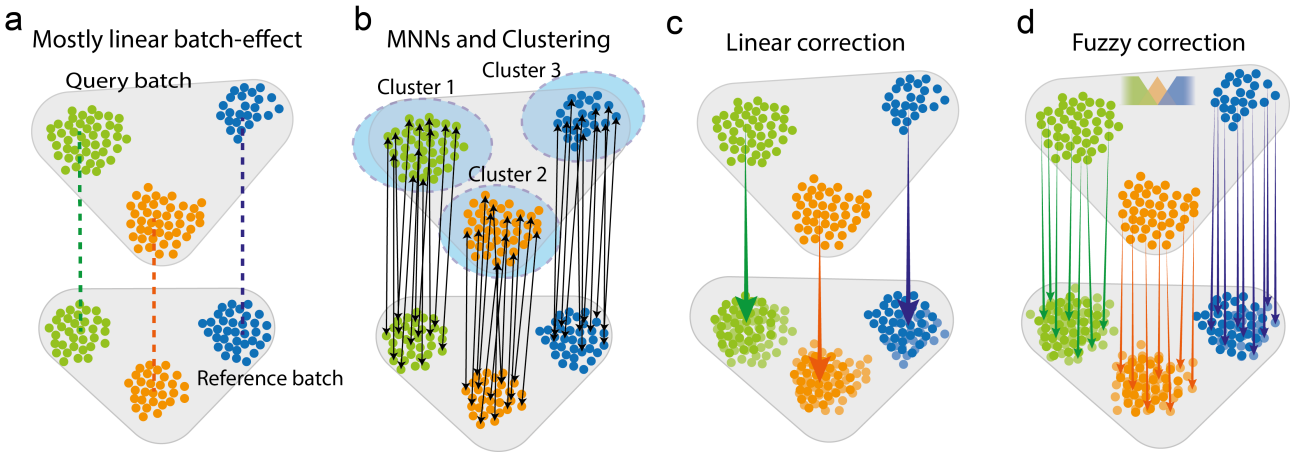

Figure 3.1.1: Overview of Canek workflow. **a)** Canek starts with a reference batch and query batch, assuming a mostly linear batch effect within clusters. **b)** Clusters of cell are defined on the query batch and MNN pairs (arrows) are used to define batch effect observations. **c)** The MNN pairs from each cluster are used to estimate cluster specific correction vectors. These vectors can be used to correct the batch effect or, **d)** a non-linear correction can be applied by calculating cell-specific correction vectors using fuzzy logic.

fuzzy correction, every cell from the query batch is corrected using a *cell-specific* correction vector.

Canek expects input datasets to be log normalized and the output dataset retains the same dimensionality (number of genes) as the input batches. To speed up computations, the first 50 principal components (PC) [42] are calculated using the *prcom_irlba* function from the irlba R package [32]. This low-dimensional space is used to identify MNN pairs, to cluster the query batch, and to obtain the fuzzy correction vectors. The input log normalized counts are used to calculate the correction vectors for each cluster.

3.2 Canek's methods

This section explains the methods used by Canek to correct batch effects: *batch effect observations*, *clustering*, *correction vector*, *fuzzy correction*, *hierarchical integration*, and *pairs filtering*.

3.2.1 Batch effect Observations

Following Haghverdi et al. (2018) idea [18], Canek uses *mutual nearest neighbors* (MNN) pairs as *batch effect observations*. Assuming that *at least one cell is shared between batches* [18], MNN pairs allow the quantification of batch effects as the gene expression differences of a set of *mutually related* cells. Figure 3.2.2 depicts the process to obtain MNN pairs from a pair of batches. In this process, relationships of cells from different batches are defined by obtaining the *crossed* k-nearest neighbors (kNN) among batches, then, MNN pairs are formed from cells with a double (mutual) relationship. For example, suppose that there is only

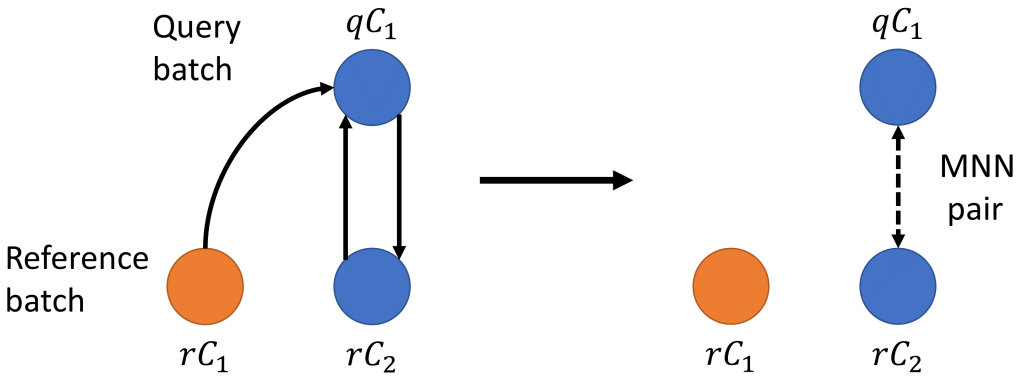

Figure 3.2.2: Mutual Nearest Neighbors (MNN). MNN pairs are defined by the intersection of the crossed k nearest neighbors of two input batches. Arrows in the left plot represents the crossed $k = 1$ nearest neighbors for a query and a reference batch. In this example, the query batch contains only one cell qC_1, while the reference batch contains two, rC_1 and rC_2. The dashed arrow in the right plot represents a MNN pairs formed by the mutual relation between qC_1 and rC_2.

one cell in the query batch qC_1, and two cells in the reference batch, rC_1 and rC_2 (Figure 3.2.2). When $k = 1$, cells rC_1 and rC_2 share a common kNN qC_1, but qC_1 kNN is only rC_2. Because of their mutual relationship, a MNN pair is formed between qC_1 and rC_2 (Figure 3.2.2). On the other side, qC_1 and rC_1 do not have a mutual relationship, thus no pair is formed between these cells.

In Canek's workflow, the crossed 30 nearest neighbors are found using the first 50 PC of the reference and query batches. Then, MNN pairs are formed between cells that fulfill the mutual relation criterion.

3.2.2 Clustering

Assuming that *the biological insights of cells are stronger than technical differences of batches*, Haghverdi et al. (2018) [18] used MNN pairs to characterize an *almost orthogonal* batch effect between batches. In this orthogonal assumption, small variations in the batch effect are caused by noise or by non-linearities (see supplementary material of [18] for a deeper discussion of these assumptions). One common way to analyze a system with *unknown* non-linearities, is to *partition* the analysis into bounded regions where the system presents a *linear behavior* [50]. In this "divide-to-conquer" approach, the *local linear* approximations of the system have the advantage of being *easier to solve* than the original problem, and the independent solutions can be *joined back* to form a *global non-linear* solution. Following this approach, in Canek's workflow, bounded regions of related cells are defined by clustering the query batch (Figure 3.1.1). The clusters and the MNN pairs are used in further steps to find independent batch effect correction vectors, which will be joined back using fuzzy logic to obtain a non-linear solution to the batch effect problem (see sections 3.2.3 and 3.2.4).

In Canek's workflow, clustering of the first 10 PC of the query batch is performed using the *Louvain* algorithm from the *igraph* R package [11].

3.2.3 Correction Vector

Following the assumption of a *local orthogonal batch effect*, for each cluster, the next linear relation is stated:

$$g^i_{Q_k} = g^i_{R_k} + g^i_{BE_k} + \epsilon \tag{3.1}$$

where $g^i, i = 1, ..., n$, is the log-normalized gene expression level of the n genes from the input batches. The batch effect g_{BE_k} is represented as an additive value in the query batch g_{Q_k} in terms of the same gene in the reference batch $g_{R_k}, k = 1, \cdots, p$, being p the number of MNN pairs from the cluster under analysis. Finally, ϵ represents a normally distributed random error term with mean zero and standard deviation σ, which is assumed to be independent of g^i on each cluster. Thus, using

$$g_{Q_k} - g_{R_k} = g_{BE_k} + \epsilon \tag{3.2}$$

on each gene i, assuming that the log-normalized expression levels of g_{Q_k} and g_{R_k} are normally distributed (see section 2.1.2), the term $g_{BE_k} + \epsilon$ would follow a normal distribution with mean $\mu = g_{BE}$ and standard deviation σ. Accordingly, the *maximum likelihood estimator* of the batch effect would be the mean of the gene expression subtraction between MNN cells pairs (e.g. $\hat{g}_{BE} = \frac{1}{n}\sum_{k=1}^{p}(g_{Q_k} - g_{R_k})$). However, there is a complication with this approach, since *erroneous pairs* between cells from distinct but related cell types could be formed, resulting in the incorrect integration of dissimilar sub populations [51]. To avoid this problem, reasoning that abnormal pairs would appear as outliers to the normal distribution of $g_{BE_k} + \epsilon$, the correction vector is estimated as

$$CV = -\begin{pmatrix} \hat{g}^1_{BE_k} = Med(g^1_{Q_k} - g^1_{R_k}) \\ \vdots \\ \hat{g}^n_{BE_k} = Med(g^n_{Q_k} - g^n_{R_k}) \end{pmatrix} \tag{3.3}$$

where the function Med represents the statistical *median*, which is less affected by outliers than the mean. Canek uses this approach by default to reduce the impact from outliers, but it is possible to perform an optional filtering step (with extra computational cost) based on the *interquartile range* to detect MNN outliers (see section 3.2.6).

3.2.4 Fuzzy correction

In the previous section, *linear batch effect correction vectors* (CV) were calculated for each cluster using MNN pairs and linear estimation. This CV could be used to integrate batches where all the cells belonging to the same cluster would be equally corrected. However, because of the inherent *inflexibility* of this approach, e.g. the number of *degrees of freedom* (the number of clusters) is low as compared with the number of cells, this linear correction might be insufficient to *properly* mix cells from different batches. This problem is especially visible

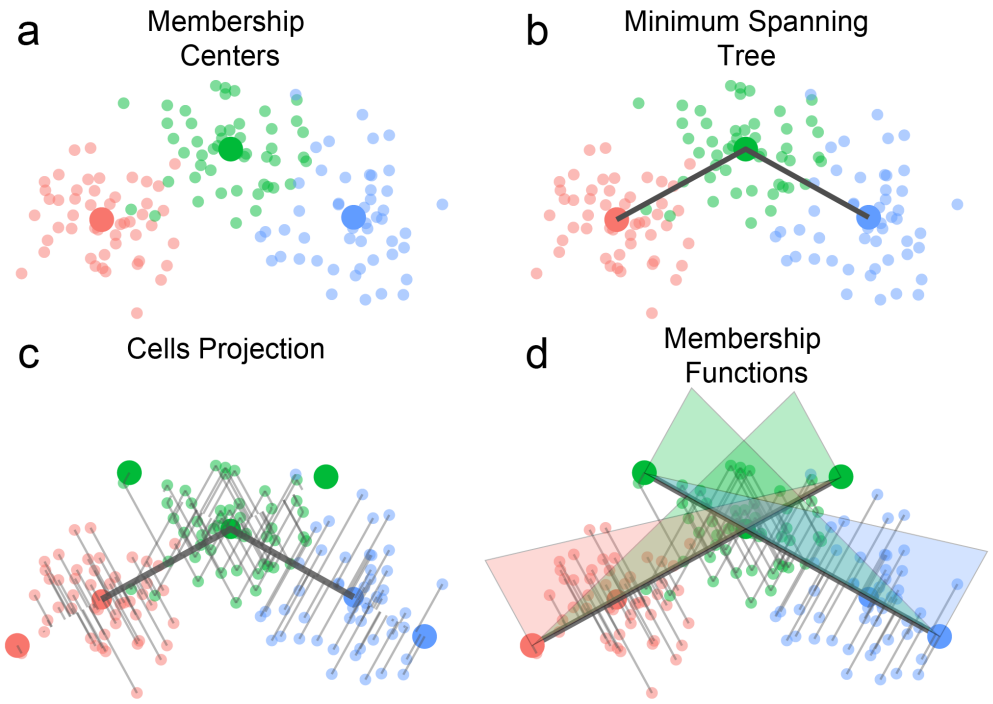

Figure 3.2.3: Fuzzy workflow. (a-b) A Minimum Spanning Tree (MST) is constructed using the membership center points as nodes. **(c)** For each edge, the cells that belong to the memberships are scalar projected and the maximum and minimum scalars are found. **(d)** Using the maximum and minimum scalars, a pair of membership functions (MF) are constructed for each edge of the MST. The membership functions score each of the cells.

in cases where batch effects have ”strong” non-linearities, e.g. samples sequenced with different technologies. To reduce this problem, Canek implements a *fuzzy logic* procedure to *smooth* the transitions of CV, obtaining a *specific correction vector for each cell* of the query batch. In the following description, according to the fuzzy logic framework, the clusters identified in section 3.2.2 will be considered as *memberships*. Canek uses the *Takagi-Sugeno* fuzzy model [54] to smooth CV transitions between memberships. In this model, one or more *premise variables*, the variables used to support conclusions, are used to describe a fuzzy subspace of input-output relations. The fuzzy subspace is obtained using a set of linear functions called *membership functions*, which can be seen as *scoring functions* to associate the inputs with the premise variables. Canek uses the *distance between memberships* as the premise variable to describe a fuzzy subspace of the batch-CV relation. Accordingly, using the PC of the query batch, a minimum spanning tree (MST) is created over the memberships' center points (MC) using the *mst* function from the R package igraph [11] (Figure 3.2.3a,b). For each edge of the MST, a pair of membership functions (MF) are constructed. These MF are used to calculate a *fuzzy score* for the cells (Figure 3.2.3c,d, and Figure 3.2.4). For example, let us consider an edge that joins the centers of memberships number 1 (MC_1) and 2 (MC_2). For each cell j that belongs to memberships 1 or 2, V_j is the vector for cell j from MC_1 in the PCs embeddings. Similarly, let V_{MC_2} be the vector corresponding to MC_2.

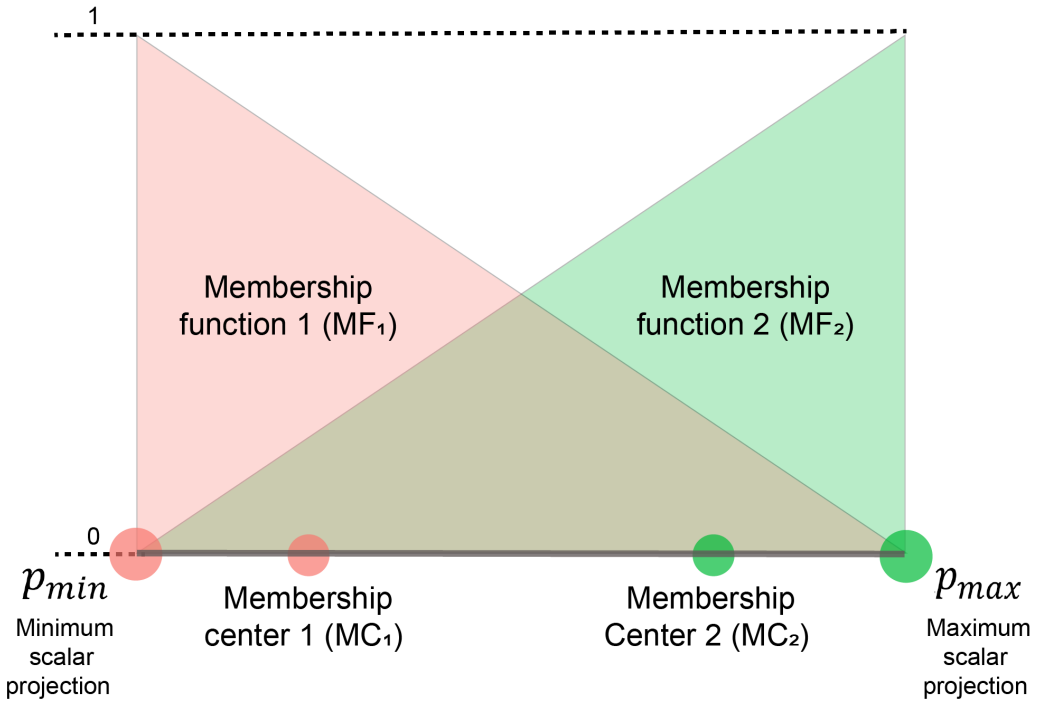

Figure 3.2.4: Construction of a pair of membership functions (MFs). The MFs serve as nonlinear mappings from a membership-specific correction to a cell-specific correction. For a given edge, using the vector that joins the membership centers 1 and 2, a scalar projection is found for each of the cells from the two memberships. The MF_1 and MF_2 are defined such that their minimum and maximum values correspond with the minimum and maximum scalar projections, and their intersection is located at the midpoint of the line that joins these scalar projections.

Then, each cell j is scalar projected p_j onto the line connecting MC_1 and MC_2 as:

$$p_j = V_j \cdot \frac{V_{MC_2}}{\|V_{MC_2}\|} \tag{3.4}$$

where the operator $\cdot$ denotes the dot product, and $\|V_{MC_2}\|$ is the length of V_{MC_2}. Then, membership functions are constructed (i.e., MF_1 and MF_2) as

$$MF_1(j) = 1 - \frac{p_j - p_{min}}{p_{max} - p_{min}} \tag{3.5}$$

$$MF_2(j) = \frac{p_j - p_{min}}{p_{max} - p_{min}} \tag{3.6}$$

Here, p_{max} and p_{min} are the maximum and the minimum of the scalar projections of the cells in the memberships ($p_{max} = \max_j p_j$ and $p_{min} = \min_j p_j$). In this way, the membership function MF_1 (MF_2) takes the maximum value 1 (the minimum value 0) for p_{min} and the minimum value 0 (the maximum value 1) for p_{max}, respectively, and linearly interpolates for the other values of the projections. (Figure 3.2.4). Then, cell specific correction vectors are calculated CV_j to combine the membership's correction vector $CV^{(l)}$ (see section 3.2.3) with the membership functions:

$$CV_j = \frac{\sum_l MF_l(j) CV^{(l)}}{\sum_l MF_l(j)} \tag{3.7}$$

here, it is worth noticing that a cell that belongs to a membership connected to more than one edge would obtain multiple scores from its membership. In this case, the mean of the multiple scores is used as the score from this cell's membership, and the scores from the other memberships remain unchanged. Even though the fuzzy scheme is applied in a low dimensional representation, the final output is in the original dimensionality of the input datasets. In Canek's R package, the fuzzy step can be skipped by setting the boolean parameter fuzzy to FALSE. In this case, all the cells from the same cluster will be identically corrected using the cluster-specific CV found in previous steps.

3.2.5 Hierarchical integration

When Canek is applied to more than two input batches, a hierarchical integration is used to correct batch effects in all the input datasets. First, the batches are sorted by cell number in descending order and the batch with the highest number of cells is used as the first reference batch. Then, the query batch is defined as the batch which the highest number of MNN pairs with the reference batch. Once the reference and the selected query batch are integrated, the integrated batch becomes the new reference, and the new query batch is selected following the previously explained procedure. This process is performed until all the input batches are integrated.

3.2.6 Pairs filtering

Assuming that erroneous MNN pairs would appear as outliers from the normal distribution of $g_{BE_k} + \epsilon = g_{Q_k} - g_{R_k}$ (see section 3.2.3), the median function is used to reduce the impact of outliers on the correction vector estimation. In addition, the user can select an extra filtering step based on the interquartile range:

$$IQR = Q_{75} - Q_{25} \tag{3.8}$$

where Q_{75} and Q_{25} are the $75th$ and $25th$ percentiles of the distribution of the p MNN pairs' Euclidean distance $d(k)$, $k = 1, \cdots, p$. Therefore, any outlier MNN pair is selected and filtered as:

$$MNN_k \textbf{ IS } \text{outlier } \textbf{IF } d(MNN_k) < (Q_{25} - 1.5 * IQR) \textbf{ OR } d(MNN_k) > (Q_{75} + 1.5 * IQR). \tag{3.9}$$

3.3 Metrics

Different metrics have been developed to assess batch-effects in scRNA-seq data [9, 18, 51, 29]. In this section, two popular metrics are introduced: the k-*nearest-neighbor batch effect test* (kBET) [9], and the the *average Silhouette width* (Silhouette) [42], used to assess *mixing of batches* and *cell type preservation*, respectively.

3.3.1 k-Nearest Neighbour Batch-Effect test

The k-Nearest Neighbour Batch-Effect test (kBET) is a bioinformatics tool specifically designed to assess batch effects in scRNA-seq data [9]. kBET compares the local and the global distribution of batches using a k-nearest neighbor matrix of cells. The comparison of distributions is performed using the *Pearson's* χ^2*-test* under the null hypothesis: *all batches are well-mixed*. The results of the analysis are provided as a *rejection rate* of the null hypothesis with scores between 0 and 1, where lower scores mean a better mixing of batches. One parameter that directly affects kBET results is the choice of the number of nearest neighbors k. Therefore, following the idea of Tran et al. (2020) [56], to objectively assess corrections in chapter 5, k was fixed as the 5%, 15%, and 30% of the mean cell number of the datasets. To ease the interpretation of this metric, an "*acceptance rate*" was calculated by subtracting the rejection rate from 1.

3.3.2 Silhouette width score

Silhouette width score assesses the separation among cells from the same cluster as compared with cells from other clusters. Let $a(i)$ be the average Euclidean distance of cell i to all other cells from the same cluster as i, then the Silhouette width $s(i)$ is defined as:

$$s(i) = \frac{b(i) - a(i))}{\max\left(a(i), b(i)\right)} \tag{3.10}$$

where b(i) is calculated as

$$b(i) = \min d(i, C) \tag{3.11}$$

being $d(i, C)$ the average distance of cell i to all the other cells assigned to different clusters C. A higher silhouette score means a longer separation between clusters.

Part II

Results

4 Benchmark

Part I introduced the main methods for single-cell RNA-sequencing (scRNA-seq) data and Canek, a batch effect correction algorithm for the integration of scRNA-seq experimental replicates performed with the same technology. The main goal of this chapter is to assess Canek and eight state-of-the-art methods (Combat, ComBat-seq, Harmony, Liger, MNN, Scanorama, scMerge, and Seurat [25, 66, 29, 63, 18, 22, 33, 51]) on different scenarios. Table 7.2 describes the methods and the versions used on the tests. The first set of tests compares methods in the situation of *known conditions*, which are useful to investigate different *biases* that integration methods could introduce. Next, methods are evaluated under *unknown conditions* scenarios with *sample* and/or *technology differences*. Then, Canek was implemented on a *human lung dataset* [1] with more than 300,000 cells, demonstrating its applicability on *large data*. The last section of this chapter presents a *running time benchmark* of methods using simulated data with varying numbers of cells. A description of the public datasets used on these tests can be found in Table 7.1.

4.1 Known conditions

In this section, batch correction methods are assessed under *known condition* scenarios, *known cell types*, *known zero batch effects*, and *known non-zero batch effects*, where the simplicity of the tests and used datasets allows the investigation of *biases* introduced by integration methods, as *over-correction*. The assessment includes the *qualitative* and *quantitative analysis* of samples using *visualization methods* and *batch effect correction metrics* (see sections 2.2 and 3.3).

4.1.1 Known cell types

The purpose of this test is to compare batch effect correction methods in a scenario where *the biology of the data is known*. The datasets used on this test are the three batches introduced in sections 2.5 and 2.5.2 [68]: Jurkat, 293T, and 50:50 Jurkat:293T cells. Figure 4.1.1a shows the principal component analysis (PCA) [42] of the data before correction with cells colored by batch and by cell type. While 293T cells from different batches are well-mixed, technical differences have caused Jurkat cells to form two groups. The problem to solve is simple: to correct batch effects in the population of Jurkat cells without disturbing the 293T cells. Two metrics were calculated to quantitatively assess the integration methods, the k-nearest neighbors batch effect test (kBET) and the Silhouette width score to weigh batch mixing and cell type preservation, respectively (4.1.1f). Figure 4.1.1b-e show the integration done by Canek, ComBat, MNN, and Seurat. The complete set of corrections can be observed in Appendix-A Figures A1 for PCA and A2 for the uniform manifold approximation and projection (UMAP) [40]. Correction methods like Canek, Scanorama, and scMerge resulted in a good mixing of Jurkat cells while preserving 293T cells. On the other hand, correction methods like ComBat, Liger, and Seurat, incorrectly mixed different cell type populations. These results are in agreement with the metrics in Figure 4.1.1f, where the Silhouette score penalized the integration done by Seurat, ComBat, and Liger, while giving higher scores to Canek, Scanorama, and scMerge. These results show that Canek can correct technical differences while preserving the biological differences of samples.

4.1.2 Known batch effects

The previous example presented a simple scenario where batch effect correction methods can incorrectly merge different cell type populations, introducing *biases* that disturb the underlying biology of datasets [36, 4]. The following two tests are designed to quantify the level of biases introduced by integration methods in the presence of a *known solution*, also referred to as a *gold standard*.

Zero batch effects.

The purpose of this test is to analyze the bias introduced by integration methods in the scenario with *known zero batch effects*. Zero batch effects refer to a situation where samples were equally prepared, handled, and sequenced, thus *no technical differences exist between batches*. Unfortunately, it is not possible to obtain two real biological replicates with such characteristics. However, assuming that cells from a *single batch have no technical differences*, artificial *pseudo-batches* can be created. The process to generate a pair of pseudo-batches is depicted in Figure 4.1.2a, where two batches are created from a single batch by *sampling* cells at equal proportions *without replacement*. Therefore, the initial dataset becomes the true solution or *gold standard* (GS) with *known zero batch effects*. Then, the pseudo-batches are integrated and compared with the GS metrics in two levels:

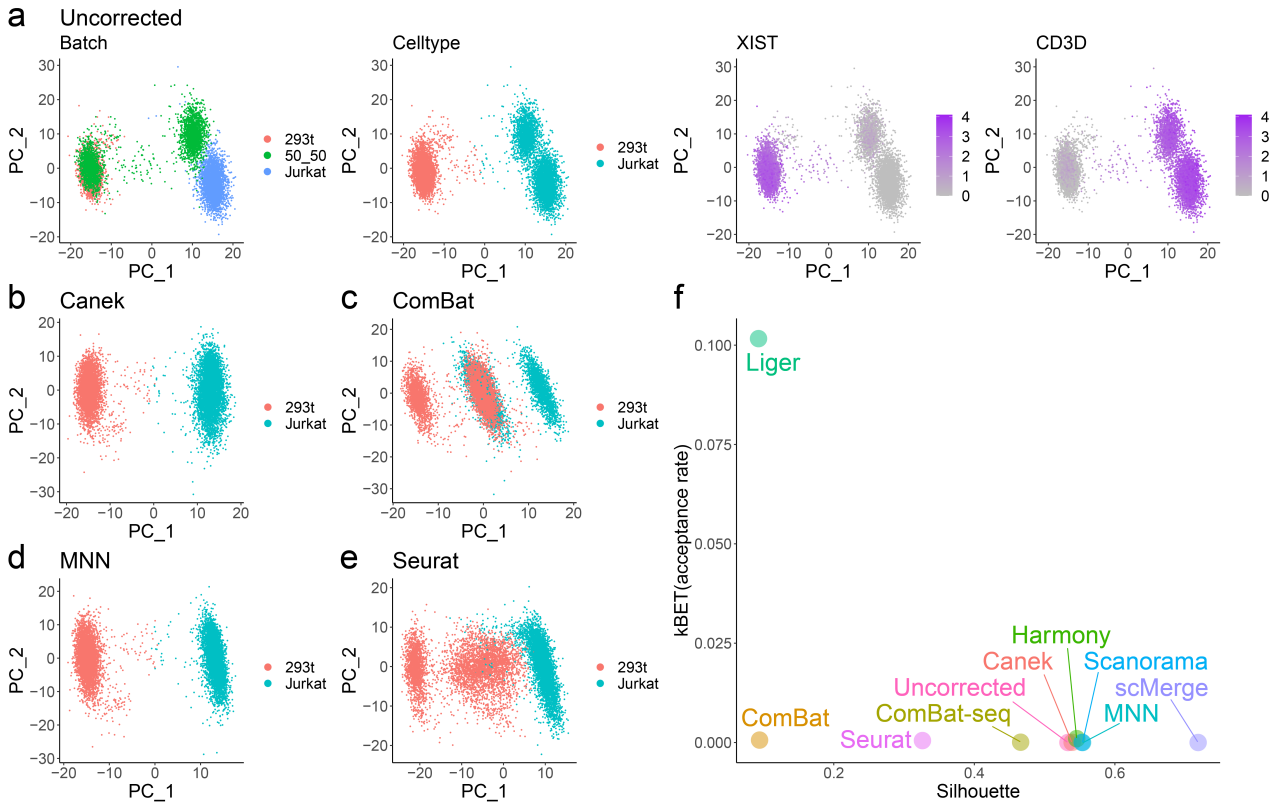

Figure 4.1.1: Integration of batches with known cell types. Batch effect correction of three batches, two containing pure Jurkat (CD3D) and HEK293T cells (XIST), and one with a 50:50 mix of Jurkat and HEK293T cells. **a)** While 293T cells from different batches are well-mixed, technical differences have caused Jurkat cells to form two groups. **b-e)** Results of batch effect correction using Canek, Combat, MNN, and Seurat. While Canek and MNN correctly integrated the Jurkat cells, Combat and Seurat incorrectly mixed different cell types. **f)** Scatter plot with kBET and Silhouette metrics calculated for the Uncorrected dataset and after correction with Canek and 8 other methods. Higher kBET (acceptance rate) and Silhouette scores mean better mixing of batches and a better delineation of cell types respectively.

mixing of cells and the *preservation of original clusters*. Accordingly, correction methods can be labeled as *biased* or *unbiased*, as those methods which distort or preserve the elemental biology of the GS, respectively. This strategy was applied to the droplet spleen dataset from Tabula Muris [53] with clusters identified using the Louvain algorithm (see section 2.3). Figures 4.1.2b-d show the clusters and pseudo-batches distribution of the GS (uncorrected data), and the corrections performed by Canek and MNN; the complete set of plots can be found in Appendix-A Figure A3. While Canek's unbiased-correction resembles the cells and clustering distribution of the GS, MNN's biased correction modified the data creating batch-specific cell populations. To quantitatively measure deviations from the GS, the kBET and Silhouette scores were calculated using the pseudo-batch and clustering labels, respectively. The results are displayed in Figure 4.1.2e with the gray dashed lines indicating the scores of the GS (e.g. the objective scores). Therefore, in a known zero batch effect scenario, Canek, ComBat, and Harmony stand out as the best-unbiased methods, MNN and Seurat are biased methods that distorted the mixing of batches, and scMerge and Liger are biased methods that deformed the distribution of the clusters. This experiment was repeated ten times to

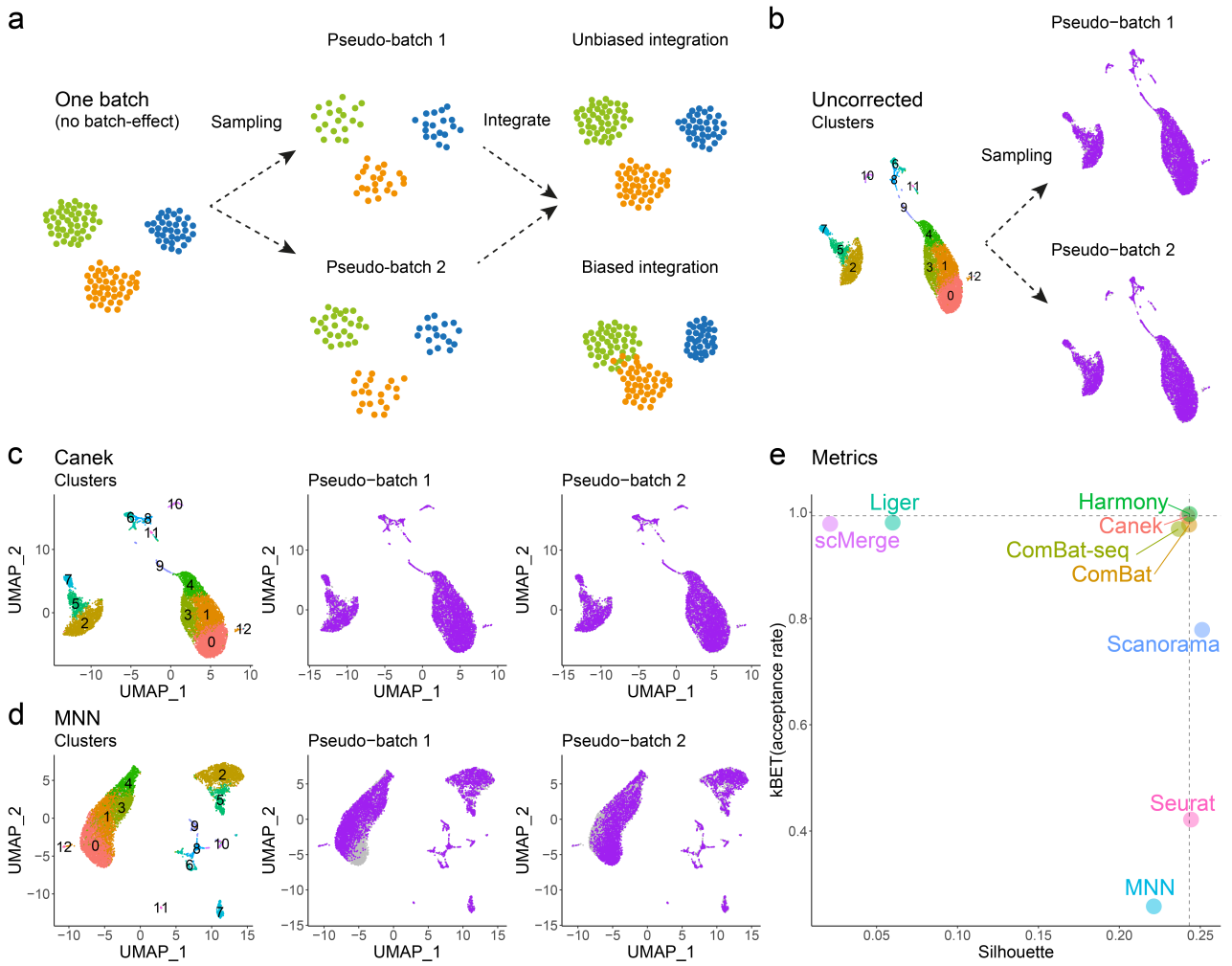

Figure 4.1.2: Integration of batches with known zero batch effects. a) Assuming that cells from a *single batch have no technical differences*, artificial *pseudo-batches* can be created by sampling cells from a given batch at equal proportions without replacement. In this scenario, the uncorrected data is the gold standard (GS). Therefore, pseudo-batches are integrated and compared with the initial dataset at two levels: mixing of cells and clustering preservation. **b)** Pseudo-batch generation using the spleen dataset from Tabula Muris. **c)** Canek's unbiased correction resembles the cells and clustering distribution of the GS. **d)** MNN biased correction modified the data creating batch-specific cell populations. **e)**kBET and Silhouette scores. The gray dashed lines indicate the GS scores and the scores from the correction methods are indicated as colored points. Canek, ComBat, and Harmony presented unbiased corrections, while MNN, Seurat, scMerge, and Liger distorted the data at different levels.

estimate the variability of the results due to pseudo-batch sampling. Appendix-A Figure A4a shows the metric results of the ten tests, where the gray dashed lines represent the kBET and Silhouette scores means of the GS, and the ovals depict the variability of the GS and the correction methods. The method with the highest variability was MNN, which, depending on the initial conditions (e.g. pseudo-batch initial distribution), distorted batch mixing or clusters distribution at different levels. Figure A4b zooms into those methods closer to the GS. Canek presents the distribution of the metrics that best resemble the ones of the GS. This demonstrates that Canek preserves the underlying biology of batches without distorting the

	Methods				
	Path cells		Group cells		
	Cell-1	Cell-2	Cell-3	Cell-4	Cell-5
Batch-1	✓	✓	✓	-	-
Batch-2	-	✓	-	✓	-
Batch-3	✓	-	-	-	✓

Table 4.1: Cell type distribution on simulated data. Paths and groups were separately simulated using splatter [64] and merged. Cells were manually removed such that the batches shared only one cell type. Then, Batch 1 is composed of two shared and one unique cell type, whereas batch 2 and 3 have one shared and one unique cell type. After removing the cell types, the number of cells per batch is: 1,671 cells for Batch 1,975 cells for Batch 2 and 964 cells for Batch 3, all of them with the same 2,000 genes.

structures of cells in the null scenario with no batch effects.

Non-zero batch effects

This test analyzes integration methods in the scenario where the *non-zero batch effects are known*. Unfortunately, due to the inherent non-linear behavior of batch effects [56], there are no public real scRNA-seq batches where technical differences are *a priori* known. However, synthetic batches that mimic statistical characteristics of scRNA-seq data can be simulated and used to assess methods performance against a known-truth [64]. Therefore, using *Splatter* [2], a package from *Bioconductor* [14] to simulate scRNA-seq data, a simulation of three batches was performed with *shared* and *batch-specific* cell types at different proportions. In Splatter, cells can be produced as *groups* or *paths*, where gene expression of cells resembles those of *different cell types* or *cells in a differentiation process*, respectively. To simulate a diverse scenario with independent and correlated cell types, a mixed approach was implemented where cells were first produced whether as groups or as paths, and then merged into a single simulation. Then, pre-selected cells were manually removed to obtain batch-specific cell types. Table 4.1 presents the distribution of cells in the batches. This process to simulate cells where implemented *with* and *without* batch effects, obtaining uncorrected and gold standard (GS) datasets, respectively. Figures 4.1.3a,b show the UMAP of the simulated GS and uncorrected datasets, respectively, with cells colored by batch and cell type. The simulated batch effects have caused the Cell-1 and Cell-2 cell types to group by batch. Then, an appropriate correction is that which corrects the technical differences in these cells, while preserving the independent Cell-3, Cell-4, and Cell-5 groups. As in the previous test (see section 4.1.2), the batches were integrated using different correction methods, and the KBEt and Silhouette metrics were calculated for the GS, uncorrected, and integrated datasets. The results are shown in Figure 4.1.3c, where the GS scores are represented by the gray dashed lines, and the metrics for the uncorrected and corrected datasets are represented by colored dots. *The best corrections are expected to be close to the GS metrics*. Figures 4.1.3b,d show the correction performed by Canek and Harmony; the complete set of corrections can be seen in Appendix-A Figure A5. While methods like Seurat, Liger, and scMerge incorrectly

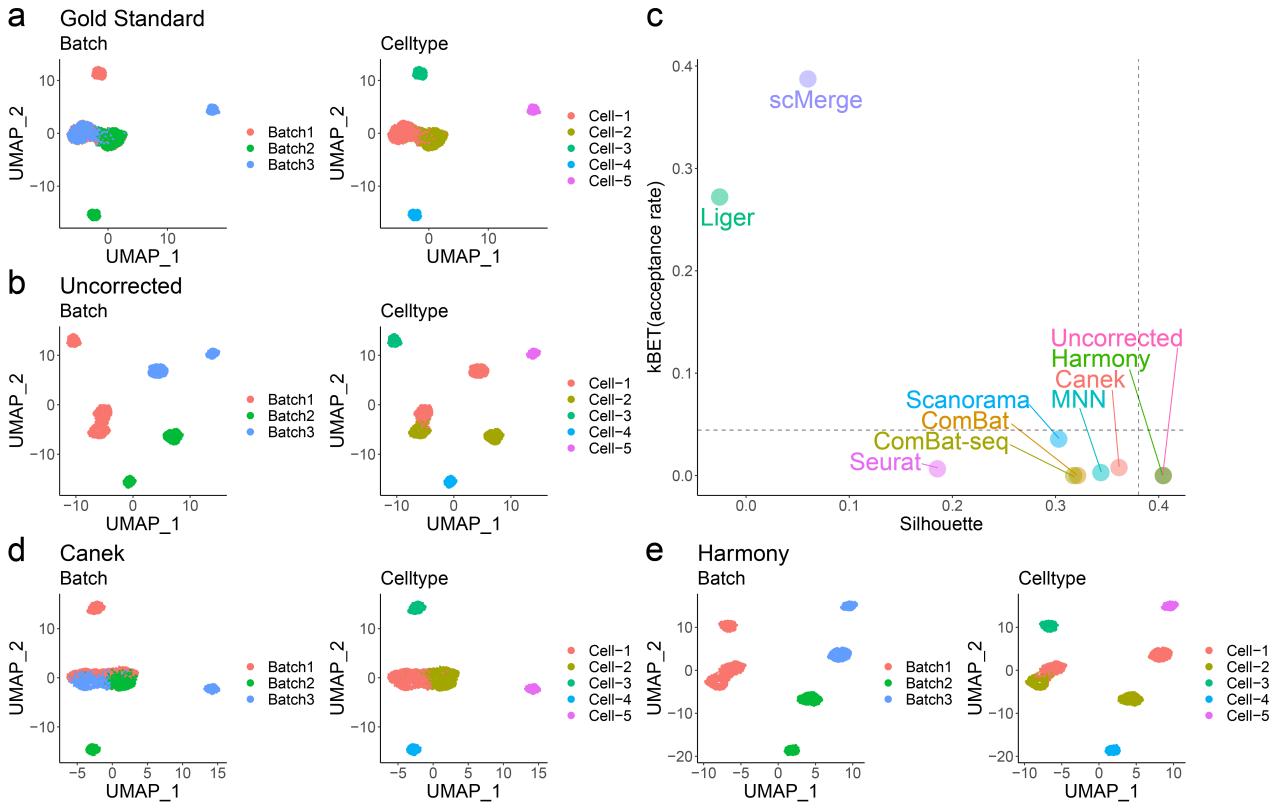

Figure 4.1.3: Integration of batches with known non-zero batch effects. Three batches were simulated using *Splatter* [2] with and without batch effects. **a,b)** Shared Cell-1 and Cell-2 cell types are well-mixed in the UMAP plot of the simulated GS, while batch effects in the Uncorrected data have caused cells from these cell types to group by batch. **c)** Batch mixing and cell type preservation of integration methods are compared with the GS using the kBET and Silhouette metrics, respectively. Good corrections are expected to be close to the GS metrics, represented by gray dashed lines in the plot. **d,e)** Harmony failed to integrate the batches where cells from the same cell type still cluster by batch. The integration performed by Canek, correctly preserved the group cells while correcting the batch effects from Cell-1 and Cell-2 cell types. This is supported by its metrics in Figure 4.1.3c, where Canek obtained the closest scores to the GS.

mixed cells from different cell types, other methods like Harmony and ComBat-seq failed to integrate the batches, thus cells from the same cell type still cluster by batch. The integration performed by Canek corrected the batch effects from Cell-1 and Cell-2 while preserving the group cell types. This is supported by the kBET and Silhouette metrics in Figure 4.1.3c, where *Canek obtained the closest scores to the GS*. This shows that *Canek best preserved the biological insights of simulated batches in the presence of known non-zero batch effects*.

4.2 Unknown conditions

In this section, Canek and eight integration methods were tested in scenarios where the biology of the integrated samples is *unknown*. It is worth noticing that even though batch mixing and cell type preservation metrics were calculated, they *lack of interpretability* due to the *ab-*

sence of a gold standard. However, the metrics can be used as quantitative measurements to support PCA and UMAP representations.

4.2.1 Same sample, different technologies

This test evaluates integration methods in a scenario where the *same sample was sequenced with different technologies*. Therefore, batch differences are expected to be driven mainly by technical variations during sample handling and sequencing. The data used in this test are the droplet and FACS batches from the Tabula Muris spleen datasets [53]. The cell type annotations from Tabula Muris were used in UMAP plots and in the calculation of Silhouette scores. Appendix-A Figure A6a shows UMAP plots for the Uncorrected data, where most cells group by batch rather than cell type. Figures A6b-j show UMAP plots after batch effect correction. Except for scMerge, which merged natural killer and T cells, and Liger, which created two populations of T cells, all the methods successfully integrated the datasets while preserving cell type annotations. The observed results are also supported by the kBET and the Silhouette metrics in Figure A6k, where the latter penalized scMerge and Liger insufficient preservation of cell types. It is worth noticing that even though Canek correctly preserved biological annotations, small-batch differences remained. This is also observed from the kBET acceptance rate, where Canek obtained a lower score as compared with other methods. However, mixing differences are negligible as compared with the strong deviations observed in the Uncorrected datasets, and cell types annotations can be accurately recovered after correcting with Canek. This demonstrates that Canek can integrate datasets from different technologies.

4.2.2 Same tissue, different technologies

In this test, batch effect correction methods were evaluated in a scenario where *different samples from the same tissue were sequenced with different technologies*. The evaluation is performed using Canek and eight methods to integrate a set of eight human pancreatic islet datasets sequenced at different laboratories with five different technologies. The cell type annotations provided in the *SeuratData* R package [44] were used to color cells in the UMAP plots and to calculate the Silhouette scores. Figure 4.2.4a shows the Uncorrected data, where the strong technical variations have caused cells from the same type to group by batch and/or technology. This Figure also contains the Canek and Seurat corrections, but the complete set of results can be found in the Appendix-A Figure A7. Seurat, Liger, and Harmony minimized batch differences, causing cells from the same type to form well-delineated unique groups. This is supported by the metrics in Figure 4.2.4d, where the top kBET scores correspond to these methods. The remaining methods, including Canek, integrated the batches with different mixing levels but batch differences remained. Because of the lack of a gold standard, it is difficult to assess which methods properly removed technical differences while retaining the underlying biology of the samples. Moreover, as samples were

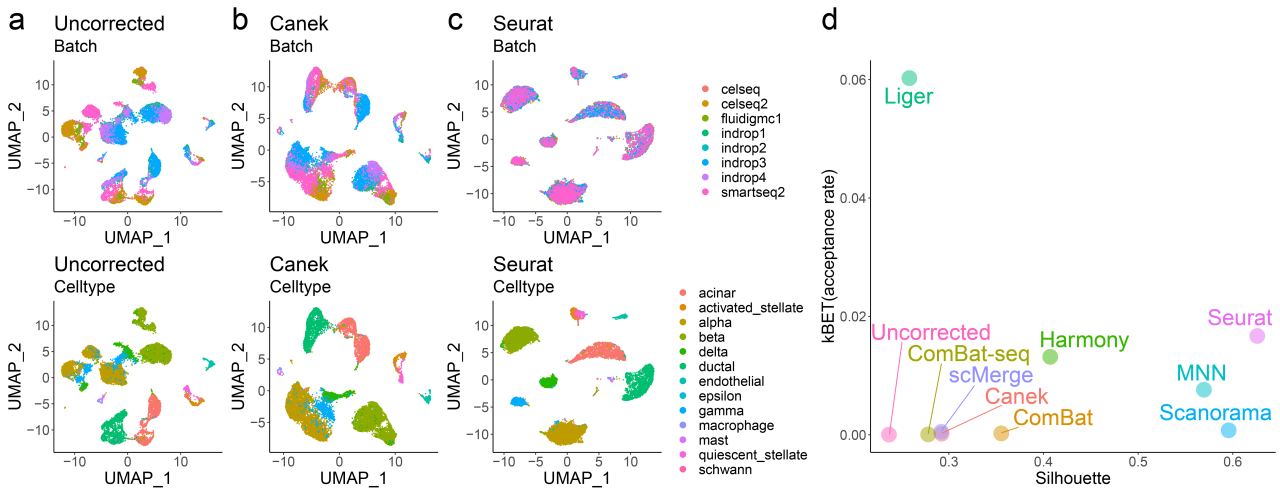

Figure 4.2.4: Integration of batches obtained from the same tissue but sequenced with different technologies. Eight pancreatic datasets obtained using different technologies were corrected. **a)** Batch effects caused the cells to cluster by batch instead of by cell type. **b)** Canek integrated the batches but some differences remained. **c)** Contrastingly, after integrating with Seurat batch differences are almost imperceptible. This is supported by the metrics in **d)**, where Seurat is within the top best-scored methods.

obtained from different donors, batch differences could be caused by distinctions in the individual heterogeneity of samples. Therefore, the diversity of the donors was further examined to get a better insight into biological conditions that could drive batch differences, finding that batches contain samples from healthy and type 2 diabetes (T2D) donors. Appendix-A Figure A8 shows the same UMAP representations as Figure A7 but coloring cells by the found disease condition (e.g. healthy and T2D). The batch differences are correlated with disease state after correcting with methods like Canek and Scanorama (Figure A8b,h), facilitating comparisons between cells from different conditions. Contrastingly, samples differences are difficult to distinguish from methods like Harmony and Seurat (Figure A8e,j), complicating further identification of samples by condition. This suggests that Canek can integrate batches while retaining biological differences even when strong technical differences are present.

4.2.3 Different conditions, same technology

In this test, batch correction methods were assessed on the integration of two PBMCs datasets with and without interferon-beta (IFN-β) stimulation [26]. In this case, small technical differences are expected but strong biological changes of samples are anticipated. The cell annotations provided in the *SeuratData* R package were used to label cells in UMAP visualizations and to calculate Silhouette scores. Appendix-A Figure A9 shows the data before and after batch correction with Canek and eight integration methods, with control and stimulated cells labeled as CTRL and STIM, respectively. The effect of IFN-β on PBMC cells has been previously studied [21], showing that IFN-β stronger stimulates monocytes as compared with B and T cells. The same effect is observed in the UMAP before correction (Figure A9a), where the clear separation between stimulated and un-stimulated cells suggests stronger IFN-β

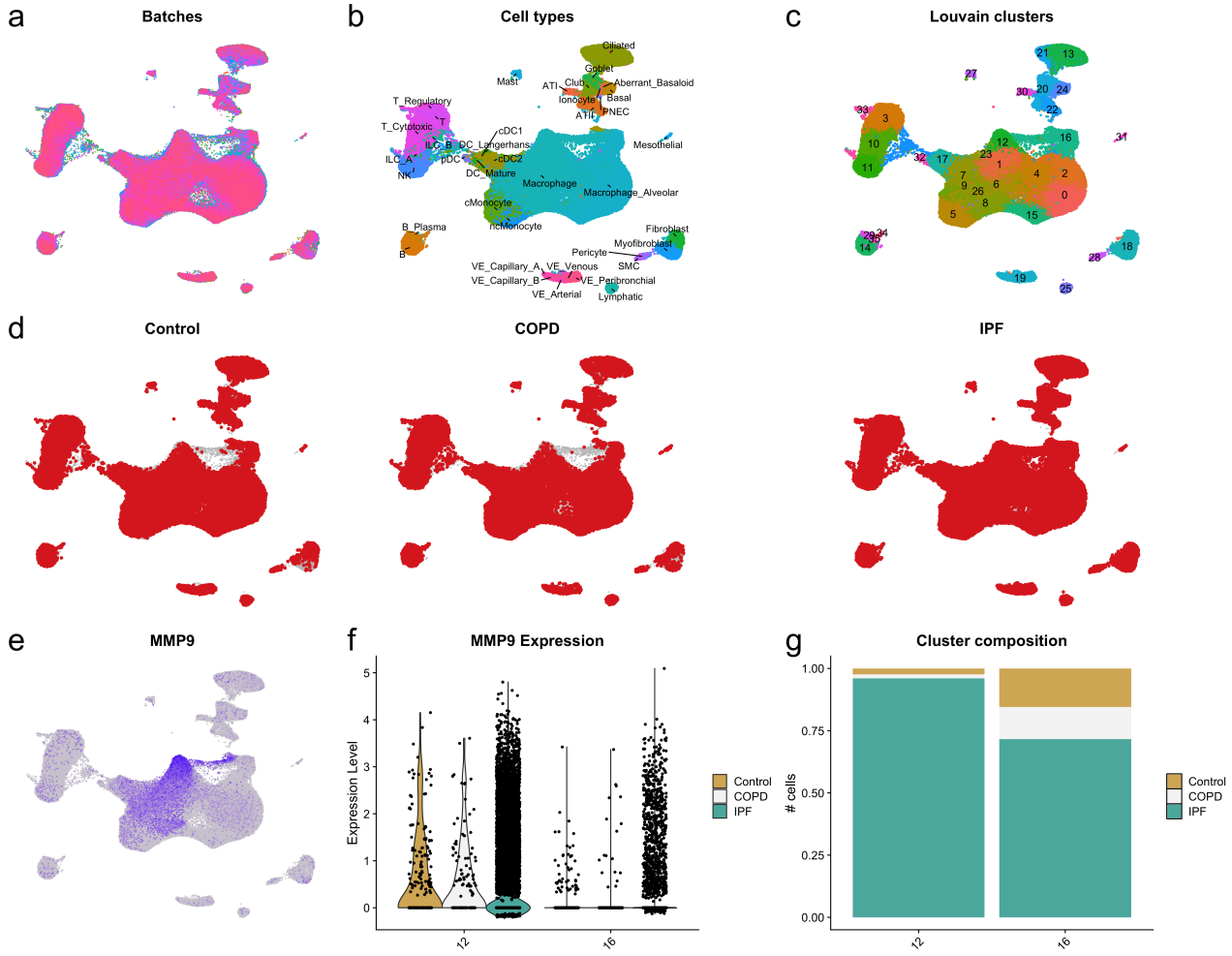

Figure 4.2.5: Integration of large datasets. A dataset containing 312,928 cells was integrated with Canek. **a)** UMAP after correction with libraries well-mixed. **b)** Cell type annotations from the original publication. **c)** Clustering based on Canek's integration. **d)** Distribution of the integrated batches by disease condition. Cells are well-mixed, except for a bridge formed from IPF cells in clusters 12 and 16 (macrophages). **e-g** The bridge is characterized by the over-expression of matrix metallopeptidase 9 (MMP9) from IPF cells.

effects on monocytes. Because of the inherent correlation of batch labels with stimulation labels, this dataset presents a difficult situation where technical differences between batches are correlated with biological conditions in the samples. Then, an appropriate integration is such that minimizes differences between batches while maintaining IFN-β stimulation effects following the results of previous studies. Figures A9b-j show UMAP plots after correction with Canek and eight integration methods, and Figure A9k displays the corresponding kBET and Silhouette scores. Seurat, Liger, scMerge, and Harmony perfectly integrated the batches. However, this highly contradicts the expected results, suggesting over-correction of these methods. Correction methods like Canek, Combat, and MNN, reduced batch differences while maintaining stronger stimulation difference in monocytes than T and B cells, as expected [26]. These results show that Canek can correct batch effects while preserving stimulation differences in samples.

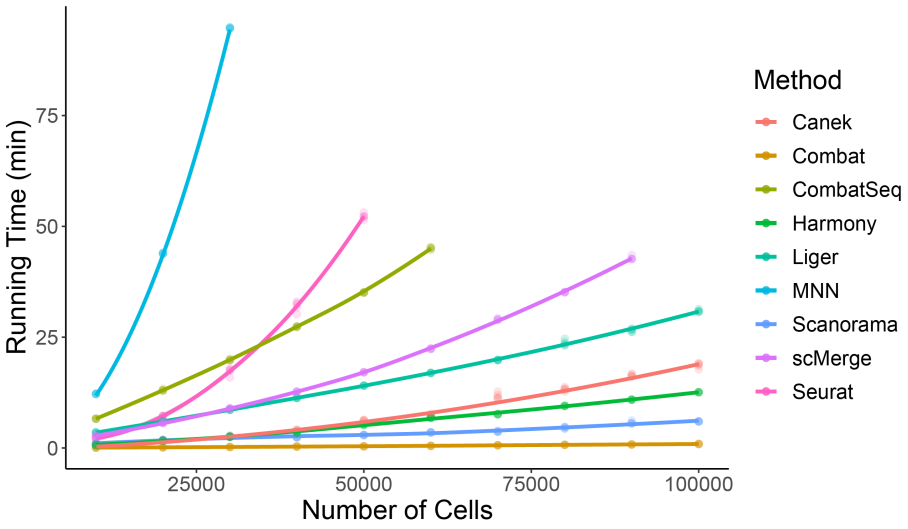

Figure 4.3.6: Time benchmark. Time benchmark using simulated datasets with the number of cells varying from 10,000 to 100,000 and a fixed number of 2000 genes. Canek and other eight methods were used five times to integrated the simulated batches and the elapsed times were recorded. Dots represent the correction time for each dataset colored by method, and lines depict estimated time vs. number-of-cells dependencies. Canek displayed a linear time increase being able to integrate 100,000 of cells in less than 20 min.

4.2.4 Large datasets

This test evaluates the performance of Canek in the scenario where the batches are composed of a *large number of cells*. The batches to integrate consist of a human lung dataset with 312,928 cells obtained from 78 donors with different conditions: idiopathic pulmonary fibrosis (IPF), chronic obstructive pulmonary disease (COPD), and healthy disease condition [1]. The dataset is divided into 107 libraries which were treated as batches. Figure 4.2.5a shows the UMAP representation of the cells after correction, where the libraries present a well-mixed distribution in most parts of the plot. Figure 4.2.5 shows the same UMAP plot with cells colored using the annotations from the original publication, which highly correlate with the clusters found after correcting with Canek (see Figure 4.2.5 and Appendix-A Figure A10). In Figure 4.2.5d, except for clusters 12 and 16 from IPF donors where the cells formed a bridge between macrophages, cells are well-mixed between conditions. This bridge is characterized by the expression of matrix metallopeptidase 9 (MMP9) (Figures 4.2.5e-g) with clusters 12 and 16 showing an over-expression of this gene in cells from IPF donors. This correlates with results found in the original publication, where the over-expression of MPP9 suggested an autocrine feedback loop for recruitment and activation in macrophages [1]. This demonstrates that Canek can integrate large datasets while preserving biological information in large datasets.

4.3 Time benchmark

The goal of this test is to compare Canek and eight integration methods in terms of computational performance. Then, using Splatter [64], simulated datasets were obtained with a number of cells varying from 10,000 to 100,000, and with a fixed number of 2,000 genes. The batches were integrated five times with each method and the elapsed times were recorded. Figure 4.3.6 shows the results of this test, with dots representing the correction time for each dataset colored by method. The lines in the plot represent square estimations of the time vs. the number-of-cells relation. The fastest methods were Combat, Scanorama, Harmony, and Canek, which displayed a linear-like behavior. It is worth noticing that these methods were able to integrate 100k cells in less than 20 min. On the other hand, methods like MNN, Seurat, and ComBat-seq showed a strong non-linear dependence taking more than 20 min. to integrate 40,000 cells, which makes them unfeasible methods for the integration of large datasets. These results demonstrated that Canek is an efficient methods that can applied in the investigation of large datasets.

5 Application example

This chapter presents an example of a scRNA-seq data multi-batch analysis following the workflow from Figure 1.2 and using Canek to correct batch effects. Using two spleen Droplet and FACS batches from Tabula Muris [53], the analysis focuses on the definition of *cell annotations* to characterize groups of cells. The analysis is divided in three main sections, *pre-processing*, *batch effects correction*, and *downstream analysis*.

5.1 Pre-processing

In this section, the pre-processing of the batches is performed in two levels: *quality control* to filter low-quality cells, and *feature selection* to identify and select high variable genes to use in further analysis.

5.1.1 Quality control (QC) and normalization

The quality control (QC) of cells is performed using three features: *number of counts*, *number of genes*, and *percentage of mitochondrial counts* (see section 2.1.1). Here, it is worth noticing that, even though *these publicly available datasets were already analyzed and filtered by the staff of Tabula Muris*, it is important to verify the quality of the cells to guarantee appropriate downstream analysis. Figure 5.1.1 displays QC scatter plots of the two datasets, where the x and y axes show the cells' number of counts and number of genes, respectively, with the color of dots representing the percentage of mitochondrial counts. In this case, all cells contained the same zero percentage of counts from mitochondrial genes. This is prob-

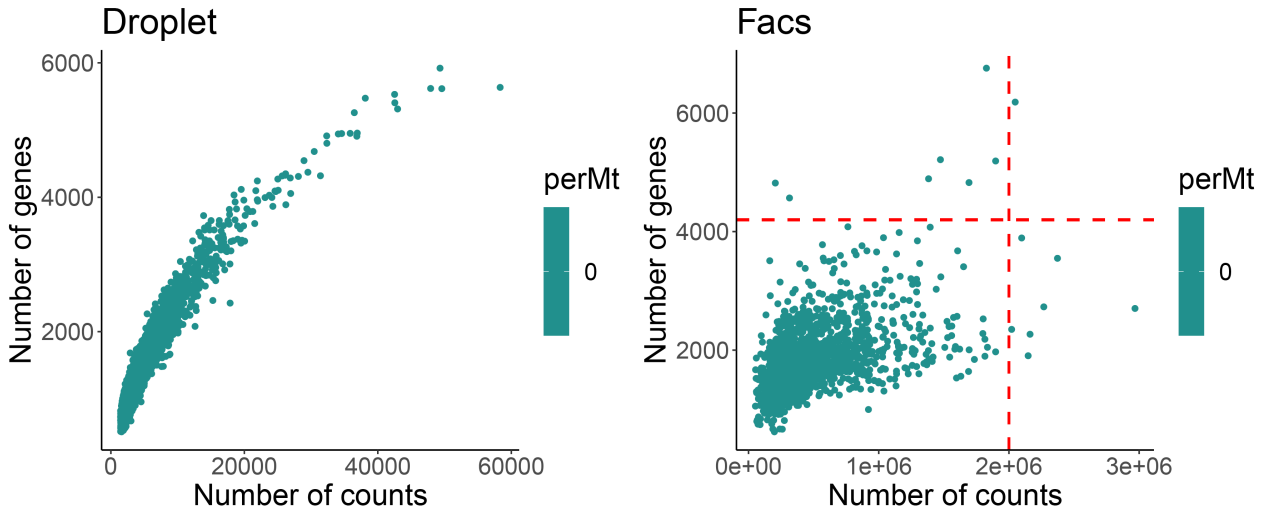

Figure 5.1.1: Quality control batches. QC scatter plots of the spleen Droplet and Facs batches from Tabula Muris. The cells from both batches contained the same zero percentage of counts from mitochondrial genes. Suspicious cells with more than 4,200 number of genes and more than 2,000,000 number of counts were filtered from the Facs dataset.

ably because of a previous QC performed in the original publication of these data. The cells from the Droplet dataset presented a consistent number of genes and counts with no apparent low-quality cells. However, some cells from the Facs dataset presented a relatively high number of counts (e.g. more than 3,000,000 of counts) and/or an inconsistent number of genes (e.g. cells with a low number of counts but a high number of genes). Even though the number of these suspicious cells is low, they were filtered to be confident in the QC of the cells used in this analysis. The red dashed lines in the right plot from Figure 5.1.1 show the QC thresholds used to filter cells, retaining cells with number of genes less than 4,200 and number of counts less than 2,000,000.

After quality control of cells, the datasets were log normalized using the function *NormalizeData* from the Seurat R package [19].

5.1.2 Variable features selection

After quality control and normalization of cells, *variable features* (VF) were identified to characterize the datasets. Using VF lowers the dimensionality of the datasets speeding up further analyses and reducing the effect of noisy genes (see section 2.1.3). However, VF selection directly affects further steps, like principal components analysis (PCA) [42], where *a small number of VF could fail to characterize cells heterogeneity* of batches, but *a large number of VF would over-fit the data* and possibly introduce noise causing the spurious distribution of cells. In this context, over-fitting refers to selecting a large number of genes that do not produce a great impact on downstream analysis. Consequently, an *incremental-intersection* approach was used to select an *enough* subset of VF without over-fitting. Therefore, the top n features were independently obtained for each batch using the *variance-stabilizing transformation* (see. section 2.1.3), resulting in two subsets of genes $n_{Droplet}$ and n_{Facs} from the

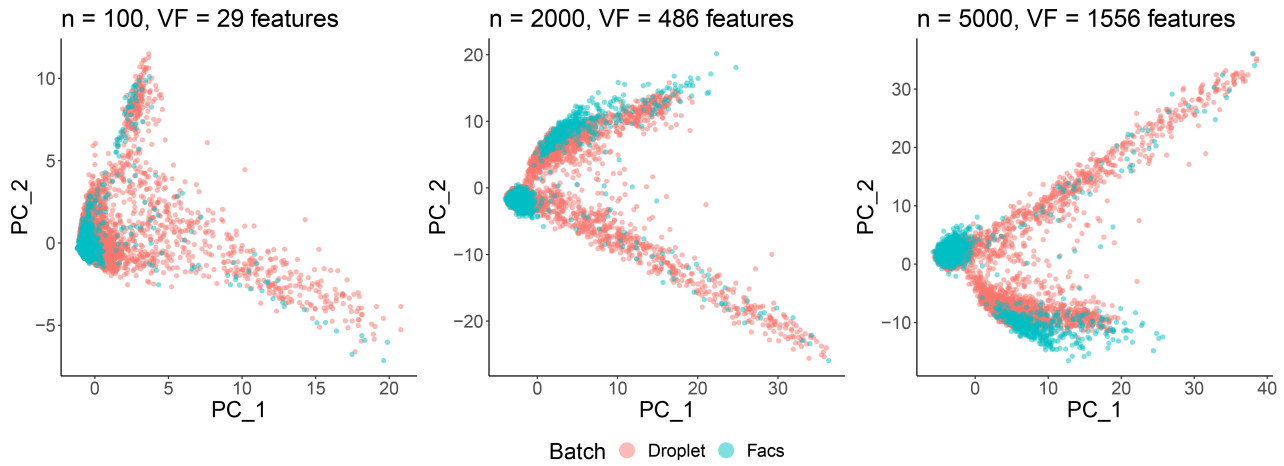

Figure 5.1.2: Effects of variable features (VF) selection. Scatter plots of the two first PCs using an incremental-intersection approach for VF selection. PCA plots from a low number of n do not present well defined groups. However, as the number of n increments, differences between batches also increase.

Droplet and Facs datasets, respectively. Then, the VF would be those genes that belong to the intersection of these two subsets (e.g. VF $= n_{Droplet} \cap n_{Facs}$). This approach was implemented with an increasing number of n to analyze how the number of features n affects further analysis as PCA. Figure 5.1.2 and Appendix-B Figure B1 show scatter plots of the two first PC using this approach with numbers of n ranging from 100 to 20,000 genes. It can be observed that using a low number of n, e.g 100 to 1,000 features before the intersection, does not produce well-defined groups in the PCA as using a higher n, e.g. 2,000 to 10,000 features before the intersection. However, as the number of n increments, differences between batches also increase. For example, if using $n = 20{,}000$ features, the first principal component *PC_1* correlates with batch differences rather than cells diversity. Therefore, a good choice of n would be between 2,000 to 10,000 features, where cells heterogeneity is well captured but batch differences are not bigger than the biological insights. In fact, within this range of n, the PCA plots are similar, suggesting that either of these n would be a good choice of VF. Then, $n = 5000$ was selected because it leads to VF = 1556, a good-balanced number of genes to use on scRNA-seq analysis. It is worth noticing that VF are chosen to characterize cell diversity between batches, not to reduce batch differences. For this task, batch effects correction is performed in the next section.

5.2 Batch effects correction

In the previous section, the independent quality control of batches was performed, and a set of intersecting variable features (VF) was selected to characterize the datasets. In the process of VF selection, principal components analysis (PCA) showed that when a sufficiently large number of VF were used, cells formed batch-specific groups (see Appendix-B Figure B1), a problem known as batch effects (see section 2.5). To analyze the correlation of batch

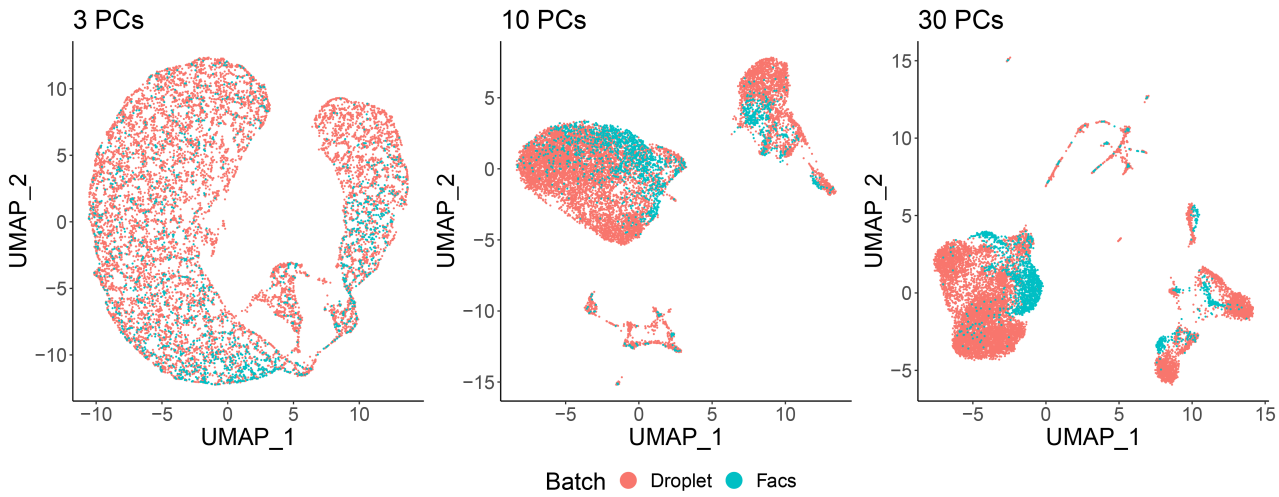

Figure 5.2.3: Batch effects on different number of principal components (PC). UMAP representations of the batches were obtained using an increasing number of PC to analyze the correlation of batch effects with the biological components of the data. Using a low number of PCs reduced batch effects improving cell mixing between batches, but did not capture cells diversity. Using 10 PCs (or more) captured more complex structures of cells but increased differences between cells from different batches.

effects with the biological components of the samples, using the selected VF, the Uniform Manifold Approximation and Projection (UMAP) [40] representation of the data was calculated for increasing numbers of PCs. Figure 5.2.3 and Appendix-B Figure B2 show UMAP plots of this analysis. While using a low number of PCs (e.g. 3 PCs) helped to reduce batch effects, using a higher number of PCs (e.g. 10 PCs or more) increased batch differences forming batch-specific clusters of cells. However, the UMAP obtained with 3 PCs didn't capture the diversity of the cells (see section 5.3.1), thus higher dimensions would be preferred. Therefore, the datasets were integrated using Canek to eliminate the correlation of batch effects with the biological insights of the data.

In Canek, two important parameters that may affect the correction results are the number of PC and the number of nearest neighbors (NN) used to obtain the mutual nearest neighbors cell pairs (see section 3.2.1). To analyze the effect of these parameters in the correction results, two independent analyses were performed as follows: first, the batches were integrated using increasing numbers of PC while fixing the number of NN to the default value (30 NN), then, the number of NN was changed while using the default number of PC (50 PC). Figure 5.2.4 and Appendix-B Figure B3 show UMAP plots of the first analysis using distinct numbers of PC ranging from 5 to 200. Canek integrated the datasets improving the mix of cells from different batches. Except for the UMAP plot of "10 PC", where the mixing of cells still presents some deviations by batch, the plots are similar. Figure 5.2.5 and Appendix-B Figure B4 show UMAP plots of the analysis of the number of NN. Except for the plot of "10 NN", all the UMAP visualizations are similar. Here, it is worth noticing that a strong change in the UMAP's embedding doesn't imply a strong change in the integration results, as the change in the structure of cells could be caused by the non-linear behavior of UMAP (see

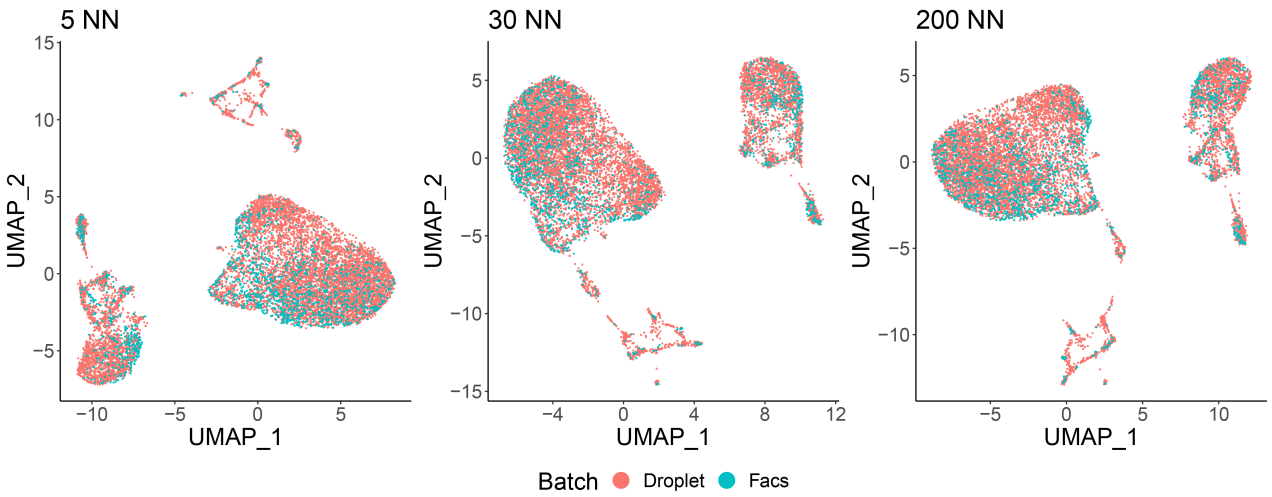

Figure 5.2.4: Effects of changing the number of PC in the integration of the datasets. The batches were integrated using increasing numbers of PC while fixing the number of NN to the default value (30 NN). Canek correctly integrated the datasets improving the mix of cells from different batches. Except for the UMAP plot of ”10 PC”, the plots are similar, proving Canek’s robustness to changes in the number of PC on this test.

section 2.2). Therefore, except for the plot of ”5 NN”, where the mix of cells still presents some deviations by batch, all the plots present well mixing of batches.

In the previous analyses of parameter dependence, Canek proved to be a *robust technique*, where strong changes in the number of PC or the number of NN didn’t produce significant changes in the results. Therefore, to ease the further usage of this analysis example, the integrated datasets using 50 PC and 30 NN, the default parameters in Canek, will be used in further steps.

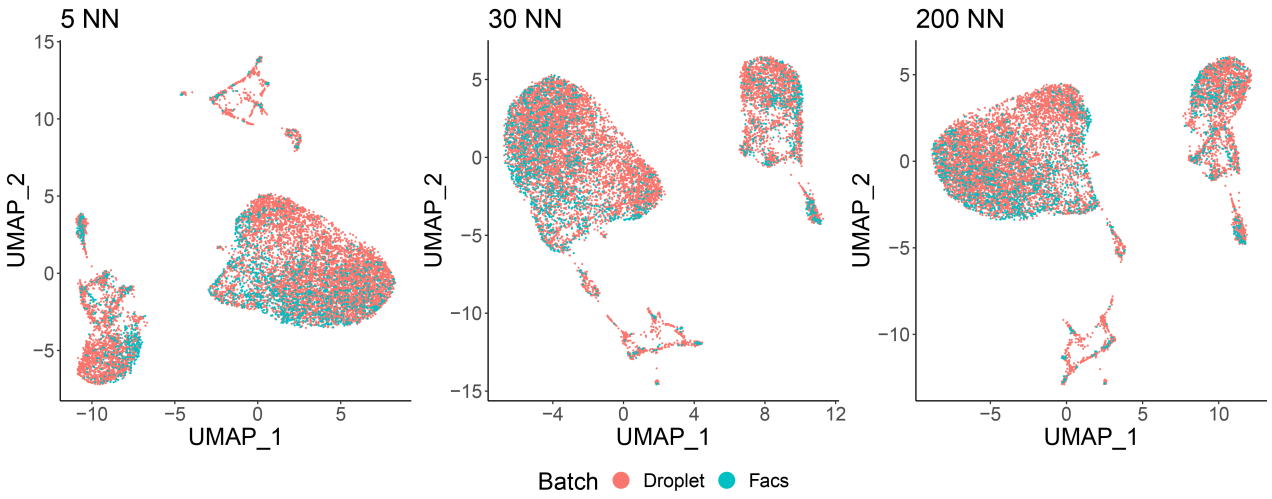

Figure 5.2.5: Effects of changing the number of NN in the integration of the datasets. The batches were integrated using increasing numbers of nearest neighbors while fixing the number of PC to the default value (50 PC). The results are similar, proving Canek’s robustness to changes in the number of NN on this test.

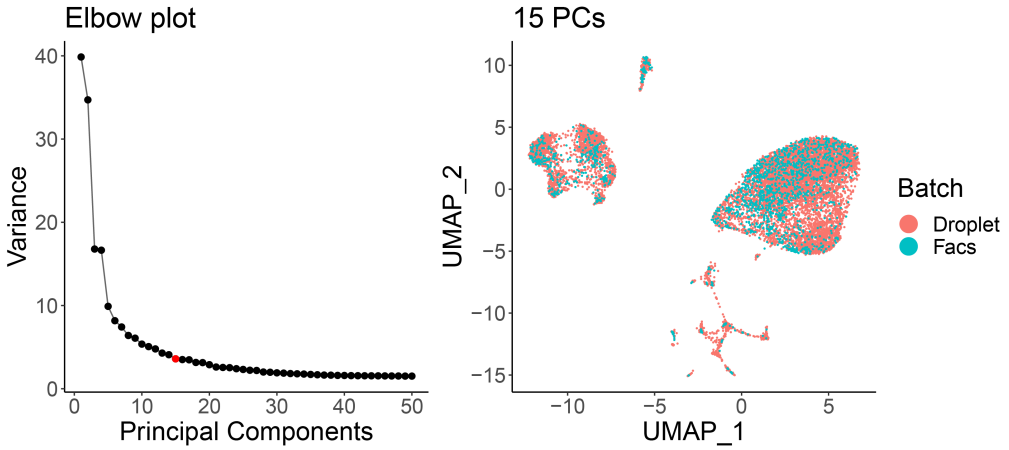

Figure 5.3.6: Summarizing the data with PCA. The left figure shows the elbow plot of the spleen datasets after integration with Canek. The variance rapidly decreases on the first 5 PCs, next it slowly decrease until 20 PCs, and then it remains almost constant. The red dot represents the elbow point at 15 PCs. The right presents the UMAP visualization using the selected 15 PCs.

5.3 Downstream analysis

This section presents the joint downstream analysis for the annotation of cells in the spleen datasets from Tabula Muris. The analysis includes *dimensionality reduction*, *clustering*, and *annotation of cells*.

5.3.1 Dimensionality reduction

Typically, dimensionality reduction is implemented on two levels: dimensionality reduction for *summarizing the data*, and dimensionality reduction for *visualization* of the data (see 2.2).

Summarizing the data. The most common dimensionality reduction method for summarizing is principal components analysis (PCA) [42], which transforms the data such that most of the variance is captured by a small set of principal components or dimensions. Depending on the integration method, batch effects correction and dimensionality reduction for summarizing the data are performed in a distinct order. Methods that correct batch effects from embedded representations of the data, like Harmony and Liger [29, 63], require to reduce the dimensionality of the data before integration. On the other hand, for methods that correct batch effect on the expression levels of variable features, like Canek, dimensionality reduction is performed after the integration of batches. An important step when using PCA is to decide the number of dimensions to use. For this purpose, the easiest way is to look at the elbow plot and select the number of PCs at the *elbow point* of the graph (see section 2.2). Figure 5.3.6 left shows the elbow plot of the spleen datasets after correcting batch effects with Canek. In this plot, the variance rapidly changes on the first 5 PCs, next slowly decrease until around 20 PCs, and then it remains almost constant. Therefore, a proper number of

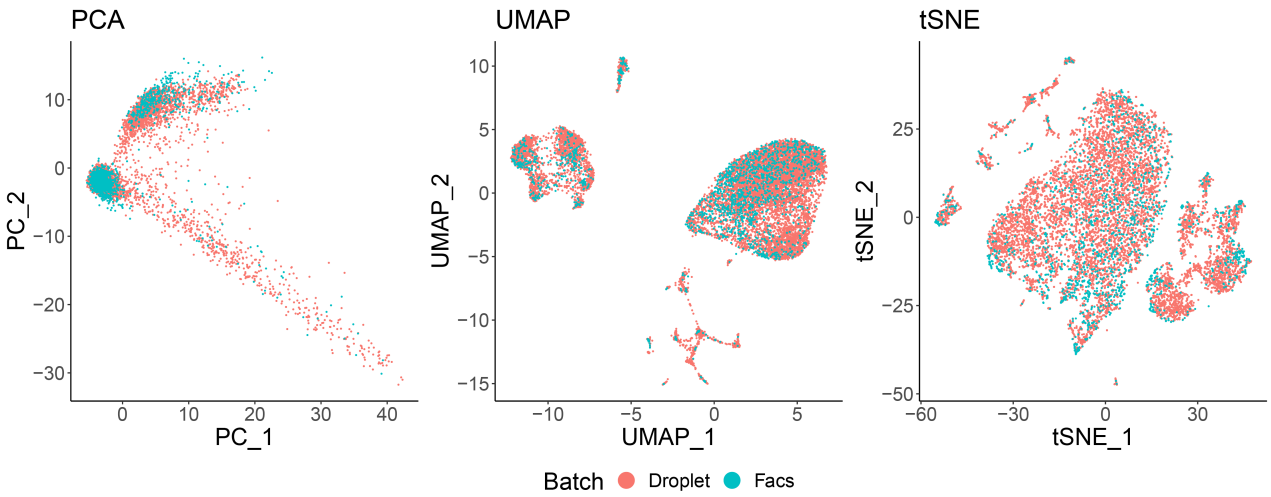

Figure 5.3.7: Visualizing the data. PCA, UMAP, and tSNE visualizations of the spleen datasets. UMAP and tSNE were obtained using 15 PCs. UMAP and tSNE present well-defined clusters as compared with PCA. The clusters from UMAP are more compact as compared with those of tSNE.

dimensions (elbow point) would be between 5 and 20 PCs. A complementary analysis to aid the selection of the number of PCs is to visualize the results using distinct numbers of PCs. Appendix-B Figure B5 shows UMAP plots for the visualization analysis using increasing number of PCs. The plots obtained using 3 and 5 PCs presented a simple structure of cells, failing to properly capture cell diversity in the datasets. On the other hand, UMAP plots using more than 20 PCs produced similar results, an indicator of over-fitting the data; in this context, over-fitting refers to using a large number of PC without relevant changes in further analysis. Therefore, consistently with the elbow plot analysis, a good choice of the number of PC would be between 5 or 20 PCs. Even though either of these choices produced similar results in the elbow plot analysis, the UMAP visualizations presented local changes in the distribution of groups of cells, where the mid-size cluster of cells using 10 PCs divided in two when using 15, or 20 PCs, a feature that could be useful to visualize clustering and annotation results. Furthermore, using 15 or 20 PCs did not produce strong changes in the visualization analysis, thus the *simplest* representation is preferred. Then, 15 PCs were selected to summarize the data in further steps of this analysis. The red dot in the elbow plot of Figure 5.3.6 left depicts the selected number of dimensions, and Figure 5.3.6 right shows the corresponding UMAP plot.

Visualizing the data.

In the previous analysis to select the number of dimensions to summarize the data, using 5 or 20 PCs didn't represent a great change in the captured variance, but it greatly changed the visualization results. Even though the results of dimensionality reduction for visualization are subjective (see section 2.2), they are of great importance as they become the *face* of the analysis. Three popular techniques for visualization are Principal Components Analysis

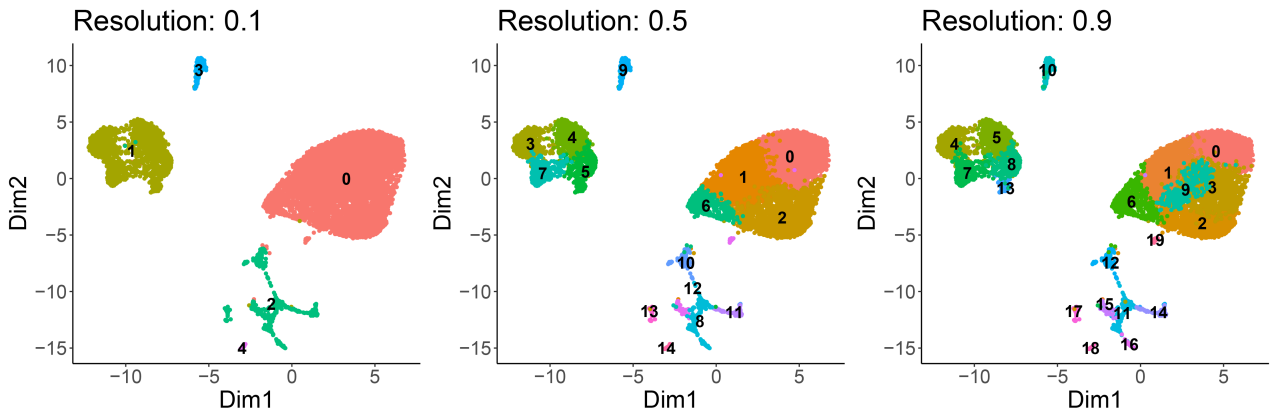

Figure 5.3.8: Clustering results using the Louvain algorithm. UMAP plots with the color of dots representing the results of Louvain using 0.1, 0.5, and 0.9 resolutions. With a resolution of 0.1, and 0.5, the algorithm identified well-defined communities. On the contrary, with a resolution of 0.9, the algorithm found closely related communities, e.g. clusters 1, 9, and 3, making it difficult to visualize differences between them.

(PCA), Uniform Manifold Approximation and Projection (UMAP), and t-distributed Stochastic Neighbour Embedding (tSNE). While PCA is a good option to visualize simple datasets (see section 2.2), UMAP and tSNE usually produce better visualizations of data with complex structures. Figure 5.3.7 show the PCA, UMAP, and tSNE visualizations of the spleen datasets. The UMAP and tSNE plots were obtained using the selected 15 PCs, and the PCA plot shows the distribution of the cells on the first two PCs. In this case, PCA's visualization didn't capture the complexity of the datasets, this is an expected result, due to the 15 PC needed to capture the variance of the data (see 5.3.1). UMAP and tSNE produced better representations of groups of cells, however, UMAP presented a wider distribution of clusters, a feature that could be useful for further results. Thus the plot produced by UMAP was chosen to represent the data in the next steps.

5.3.2 Clustering

Clustering allows the identification of groups of cells with similar expression profiles. Two popular clustering techniques are k-Means and Louvain (see section 2.3). One important advantage of the Louvain algorithm over k-Means is that the number of clusters does not need to be decided in advance, however, its results highly depend on a *resolution* parameter. Figure 5.3.8 and Appendix-B Figure B6 show the UMAP representation of the data with the color of cells representing the results of Louvain with resolution ranging from 0.1 to 1.7. When a resolution of 0.1 was used, the algorithm found 5 well-defined communities. On the contrary, if using a resolution of 1.7, 25 communities were identified, but their distribution on the UMAP plot overlaps making it difficult to visually differentiate between clusters. For example, it's difficult to recognize differences between clusters 4, 5, and 11, which are almost merged in the UMAP representation. Using differential gene expression analysis (see next section) on clustering results with increasing resolutions is a way to compare and assess

Cluster	Label	Markers
0	B cells	*Ms4a1*
1	T cells	*Cd3d*
2	Monocytes/Macrophages	*Cst3*
3	Natural Killers (NK) cells	*Nkg7*
4	Dentritic cells	*Siglech*

Table 5.1: Low-level cell markers. Cells from different clusters were labeled using five gene markers. Genes *Ms4a1*, *Cd3d*, and *Nkg7* are well-known markers of *B*, *T*, and *Natural Killers (NK)* cells. *Monocytes/Macrophages* and *Dentritic* cells, where characterized by the expression of *Cst3* and *Siglech*, respectively

clustering results. In this way, the resolution that best represents cells heterogeneity in the context of the objectives of the experiments is selected.

5.3.3 Group identities.

In the previous section, clusters were identified using the Louvain method, but depending on the resolution, different numbers of clusters were obtained, making it difficult to decide the best clustering results to represent the datasets. A common way to validate clustering results is to look at *cluster-specific gene markers* (see section 2.4), where the biology captured by the clusters should agree with the objectives of the experiment. Accordingly, clustering results should help to *characterize cell populations in the mouse Spleen at different levels*; in this context, a low-level analysis would match with already studied cell populations, and a high-level one would allow the identification of new cell types by the expression of specific gene markers. Therefore, differential gene expression (DGE) analysis was performed using the Seurat R package [51] (see section 2.4) for the identification of *gene markers*, and the results were compared with public publications and databases for the annotation of clusters at *low* and *medium* levels using clustering results with resolutions of 0.1, and 0.3, respectively. It is worth noticing that a high level would correspond to higher resolutions, e.g. 1.0 or 1.3. Those analyses were not included in this study as using higher resolutions could conclude in the *over-characterization* of cells that could complicate the understanding of the results with meaningful biological labels.

Low-level analysis.

Figure 5.3.8 left shows the clustering results using the Louvain method with 0.1 resolution, which resulted in the identification of 5 clusters. While a 0.1 resolution might not be sufficient for the proper identification of clusters in the analysis of *different* datasets, e.g. in some datasets using a resolution of 0.1 might lead to identifying only one cluster, in this case, using a 0.1 resolution captured the main groups of cells represented in the UMAP plot. Figure 5.3.9 shows violin plots of gene-markers identified with DGE analysis. Some of these genes corresponded with state-of-the-art markers of cell types, e.g. T cells characterized by the

Cluster	Low-level labels	Low-level markers	Medium-level labels	Medium-level markers
0	B cells	*Ms4a1*	T2 B cells	*Cd24a*$^{+}$, *Myc*$^{low/-}$
1			Mature B cells	*Cr2*
2			T1 B cells	*Cd24a*$^{+}$, *Myc*$^{+}$
3	T cells	*Cd3d*	Cd8 T cells	*Cd8a*
4			Cd4 T cells	*Cd4*
5	Monocytes/Macrophages	*Cst3*	Monocytes	*Cd24a*
6			Macrophages	*Ifitm3*
8			Monocytes C1qb	*C1qb*
7	Natural Killers (NK) cells	*Nkg7*	Natural Killers (NK) cells	*Cd8a*$^{-}$, *Nkg7*
9	Dentritic cells	*Siglech*	Dentritic cells	*Siglech*

Table 5.2: Medium-level cell markers. Clusters 0 (*Cd24a*$^{+}$, *Myc*$^{low/-}$), cluster 1 (*Cr2*), and cluster 2 (*Cd24a*$^{+}$, *Myc*$^{+}$) correspond to *T2*, *Mature*, and *T1 B cells*, respectively. In the case of *T cells*, cells from cluster 3 are *Cd8 T cells* and cells from cluster 4 are *Cd4 T cells*. The *Immuno-Navigator* database [61] correlated expression levels of cluster 5 (*Cd24a*) and 6 (*Ifitm3*) with *Monocytes* and *Macrophages*, respectively. Cells from cluster 8 showed a similar expression profile as *Monocytes*, with the difference of expressing *C1qb*, thus were annotated as *Monocytes C1qb*. Clusters 7 (*Cd8a*$^{-}$, *Nkg7*) corresponds to *Natural Killer cells*, and cluster 9 (*Siglech*) to *Dentritic cells*.

expression of *Cd3d* gene, but others were more difficult to define without a deeper knowledge of the biology of spleen cells. In that case, the *Immuno-Navigator* database [61] was used to correlate found markers with known cell types. Table 5.1 and Figure 5.3.9 summarize the assigned labels and the main markers used in this analysis. Cells from clusters 0, 1, and 3 exhibited high expressions of *Ms4a1*, *Cd3d*, and *Nkg7* genes, respectively, which are known markers for *B* and *T* and *Natural Killers (NK)* cells. Cluster 2 presented high levels of *Cst3*, a gene correlated with *Monocytes* and *Macrophages*. Finally, cells from cluster 4 highly expressed *Siglech*, a marker correlated with *Dentritic cells*. The sparse distribution of the gene *Cst3* in the violin plot in Figure 5.3.9, suggests the existence of more than one group of cells that were clustered together because of the low-level of clustering. This is supported by the UMAP plot, where cells from cluster 2, annotated as *Monocytes/Macrophages*, are composed by closely related groups.

Medium-level analysis.

Next, cell annotations at medium-level were performed using clustering results with a resolution of 0.3. The clusters are shown in Appendix-B Figure B6 top-center. The advantage of already having low-level annotations is that clusters from higher resolutions can be compared with the already defined labels. For example, it can be observed that the low-level cluster of *T cells* was divided into two clusters, clusters 3 and cluster 4. Even though this is a *subjective analysis*, it's useful to get a quick insight into the biology captured by clustering, easing the annotation process. Therefore, differential gene expression (DGE) analysis was performed between *similar clusters* defined by *low-level annotations*. For example, DGE was performed between clusters 3 and 4, because they are associated by the same low-level *T cell* label. In the case of clusters 7 and 9, which present similar distributions in the UMAP plot as compared

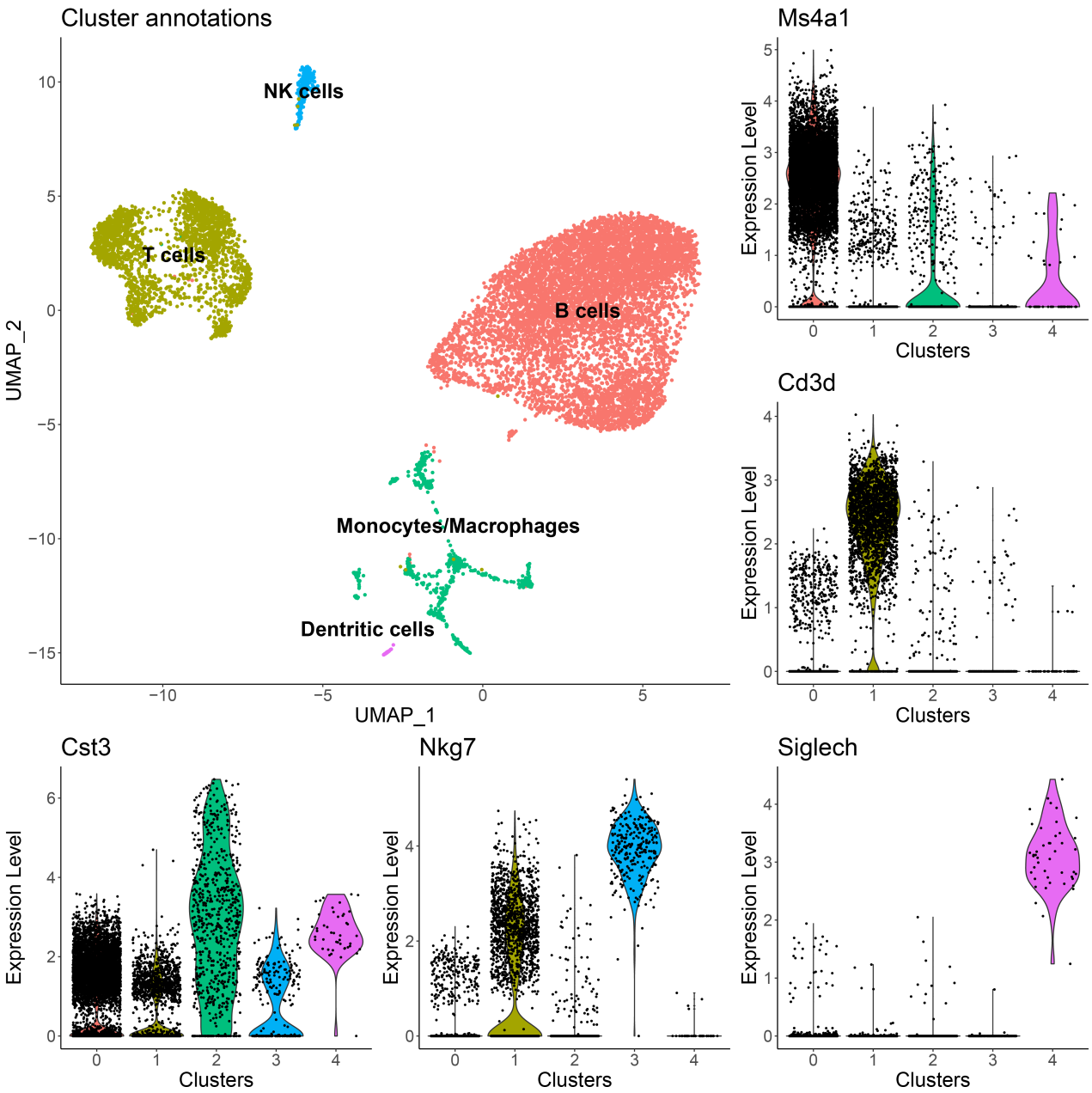

Figure 5.3.9: Low-level cell annotations. Violin plots of five cluster markers identified with DGE analysis. Cells from clusters 0, 1, and 3, respectively exhibited high expressions of *Ms4a1*, *Cd3d*, and *Nkg7* genes. Cluster 2 presented high levels of *Cst3*, and cells from cluster 4 highly expressed *Siglech*. The color of cells in the UMAP depicts the cell annotations performed using these five gene markers.

with the low-level clusters, a ”one-versus-all” DGE strategy was implemented (see section 2.4). Table 5.2 summarizes the DGE results, and Appendix-B Figure B7 shows violin plots of the found markers. Clusters 0 and 2 exhibited similar expression of *Cd24a*, but cluster 2 expressed *Myc*, an indication of being in a lower stage in the B cells differentiation process [28]. Therefore, clusters 0 and 2 were annotated as *T2* and *T1 B cells*, respectively. Cluster 1 presented high levels of *Cr2* thus was labeled as *Mature B cells* [28]. In the case of *T cells*, clusters 3 and 4 showed high levels of *Cd8a* and *Cd4*, respectively, thus they correspond to *Cd8* and *Cd4 T cells*. The *Immuno-Navigator* database [61], correlated expression levels

of *Cd24a*, and *Ifitm3* from clusters 5 and 6, with *Monocytes* and *Macrophages* cell types, respectively. Cells from cluster 8 showed a similar expression profile as *Monocytes*, with the difference of expressing *C1qb*, thus were annotated as *Monocytes C1qb*. Finally, clusters 7 (*Cd8a*$^{-}$*, Nkg7*) corresponds to *Natural Killer (NK) cells*, and cluster 9 (*Siglech*) to *Dentritic cells*, as expected. Figure 5.3.10 shows the UMAP plots with cells colored by the medium-level annotations. It can be noted that some clusters present distributions in the UMAP that suggest the existence of *sub-clusters*, e.g. *Monocytes* and *Cd4 T cells*. These sub-clusters are captured by clustering with higher resolutions. For example, the cluster 4 using 0.3 resolution divides into cluster 3 and 7 using 0.5 resolution (Appendix-B Figure B6). However, as mentioned before, clustering with higher resolutions could potentially *over-characterize* the data, resulting in highly overlapping clusters. This has the potential problem of complicating the analysis and lacking interpretability. For example, the highest clustering resolution would be the *one-cell-one-cluster* scenario, where each cell could be differentiated from the rest of the cells. Even though this analysis can be performed, it will not produce objective results as more probable DGE results would be mainly driven by random noise. Therefore, as the medium-level labels *illustrate the characterization of cells in the mouse spleen*, the main objective of this example, they were selected as the *final cell annotations* of this multi-batch scRNA-seq analysis.

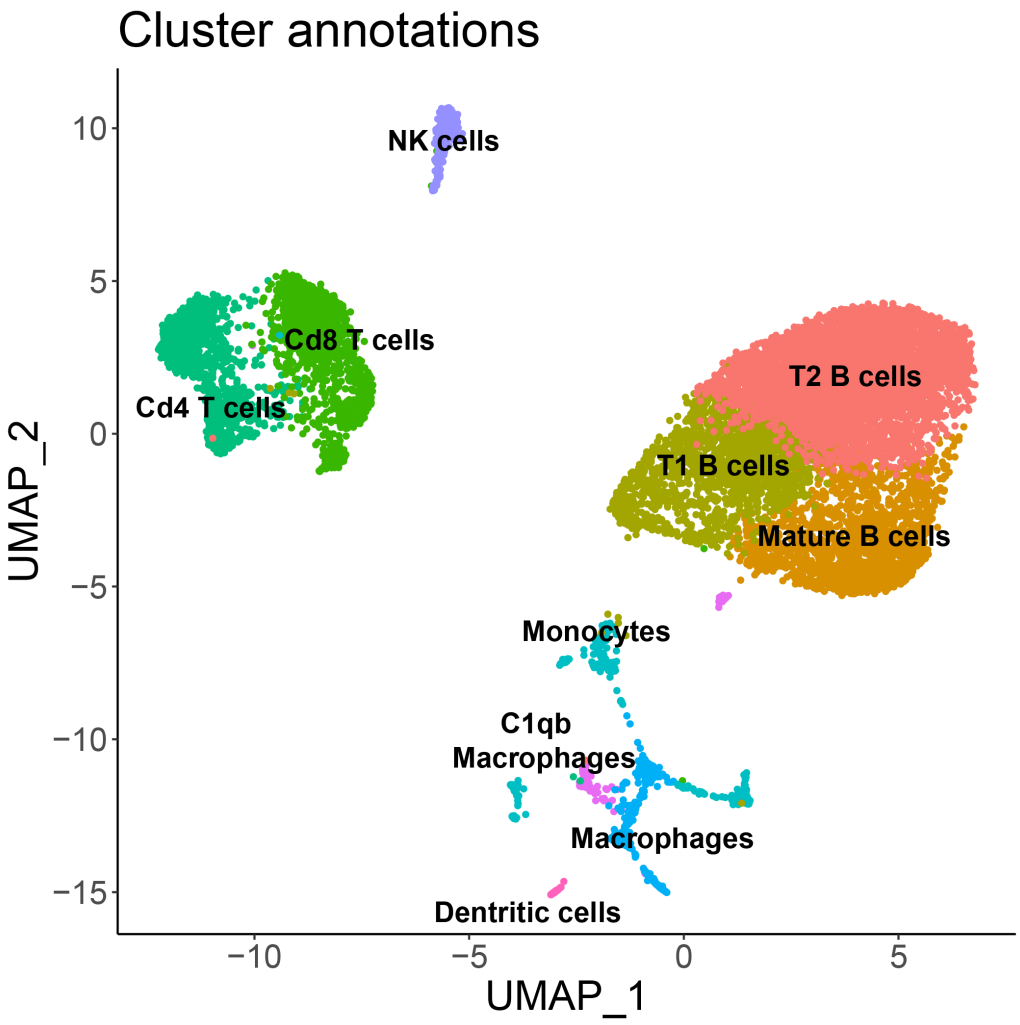

Figure 5.3.10: Medium-level annotations. Differential gene expression analysis guided the annotation of 10 medium level clusters. Three clusters of B cells in different differentiation stages were found: *T1, T2,* and *Mature B cells*. Two populations of *T cells* were identified as *Cd8* and *Cd4 T cells*. *Monocytes* and *Macrophages* were characterized by the expression of *Cd24a* and *Ifitm3* genes. A special group of monocytes expressing *C1qb* were annotated as *Monocytes C1qb*. Finally, *Natural Killer* and *Dentritic cells* were defined by the expression of *Nkg7* and *Siglech* gene markers.

6
Discussion

Existing batch effect correction methods focus on the integration of single-cell RNA-sequencing (scRNA-seq) datasets regardless of their nature, an asset for the creation of cell atlases or the comparison of samples with strong differences. However, in the presence of small technical differences, these methods could over-correct. This is of special concern in datasets where subtle changes in the gene composition of samples are expected, as in the case of experimental replicates with different conditions. Over-correction can potentially hinder downstream analysis as clustering and differential gene expression, negatively affecting the conclusions of experiments. Canek provides an unbiased framework for the integration of experimental replicates with small batch effects, e.g. data sequenced using the same technology. Canek's workflow focused on preserving the inherent structure of cells while being flexible to deal with small non-linearities that may appear on heterogeneous datasets. I compared Canek with state-of-the-art methods in diverse scenarios using qualitative and quantitative metrics. Here is worth noticing that, because of the difficulties to characterize bath-effects, quantitative metrics can be objectively used only in the presence of a *known-truth* (e.g. a gold standard solution). In other cases, quantitative metrics should be used to *support* qualitative characteristics of the integrated batches (e.g. UMAP representations). Current metrics focus on the assessment of *only one feature* of the data (e.g. mixing of batches or cell type preservation) which complicates the objective assessment of methods in different directions. Moreover, the metrics can occasionally be inconsistent, as in the integration of simulated datasets, where the uncorrected samples obtained a higher cell type preservation score as compared with the gold standard. The definition of robust metrics that weigh batch effects in more than one direction will improve the comparison of the integrated dataset under unknown conditions. To reduce this problem, I compared Canek with other

correction methods on diverse scenarios with and without a known truth. Canek was able to integrate datasets with the lowest distortion of biological signals (e.g. no over-correction) in the presence of known-zero (pseudo-batches test) and known-non-zero (simulation test) batch effects. In a different scenario, Canek was being able to correct technical differences while preserving known cell types (Jurkat and HEK293T cells test). In the tests performed using datasets that may be out-of-scope of Canek's assumption of a mostly linear batch effect, Canek was able to integrate datasets from different technologies improving the identification of previously annotated cell types (e.g. Tabula Muris spleen test). However, depending on the nature of the dataset, Canek didn't lead to the best mixing of batches (e.g. the pancreatic test), but further investigation showed that Canek preserved "latent variables" that might be influencing batch differences in the results. This was particularly notable in the integration of two datasets with and without Interferon-β stimulation, where Canek was able to reduce batch effects while preserving cell differences due to stimulation in accordance with published results. I further implemented Canek in a large dataset with more than 300,000 cells, where Canek was able to preserve condition-specific differences of transitional cells which correlated with results reported in the original publication. In the running time benchmark, Canek was within the fastest methods being able to integrate 100,000 cells in less than 20 minutes. To demonstrate the usage of Canek within a workflow, I performed a complete example using two spleen datasets where Canek led to the proper annotation of different cell types. In the same example, I showed the parameter dependence of tools used for scRNA-seq analysis. For instance, the selection of the number of variable features (VF) and the number of principal components (PC) greatly changed further analysis as UMAP, where low numbers of VFs and PCs didn't capture complex structures of cells but higher numbers intensified batch differences. In the case of PCA, using a high number of dimensions also led to *over-fitting*, where including higher dimensions did not cause visible changes in the UMAP plots. In this case, using an elbow plot aid to identify an elbow point where most of the variance of the data was captured. Using qualitative analysis of UMAP plots obtained with distinct numbers of PCs around the elbow point helped to fix the number of dimensions to use in further steps. In the case of visualization methods, UMAP and tSNE produced better-defined clusters of cells as compared with the plot generated using the first two PCs. This was an expected result because of the more than two dimensions needed to summarize the data with PCA, but simpler datasets could be correctly visualized using just two PCs, as the integration of Jurkat and HEK293T cells. Clustering was another delicate step where distinct resolutions produced significant changes in results. Assessing clustering results following the annotation of cells at different levels proved to be useful in the characterization of cell types. Because most of the parameters were chosen using qualitative features, the presented example could transmit the image of scRNA-seq data analysis being "hand-made-like" investigation of cells. However, the importance of the results is priceless, as distinct cell types from the spleen were properly characterized: T and B cells, natural killers, monocytes and macrophages, etc. These annotated cells could be used in further processes with di-

verse objectives, for example, the creation of cell atlases, the definition of gene expression dictionaries, the analysis of monocytes' differentiation process in the spleen, etc. There is a wide range of possibilities, which is the main beauty of single-cell RNA-sequencing data analysis.

7
Conclusions

Every scRNA-seq analysis requires the proper choice of parameters and tools to rightly capture the biological insights of the data following the objectives of the project. Experimental replicates performed with the same technology are commonly used for the investigation of samples with different conditions. In this kind of experiments, subtle batch effect correction methods that preserved small changes in cells populations are needed. Canek provides a simple but effective solution to this problem. Canek was the highest scored method in tests specifically designed to assess over-correction. While other methods disturbed the distribution of batches, Canek corrected batch effects without distortion to the structures of cells as compared with gold standards. Canek successfully integrated batches from different technologies in different scenarios. Depending on the nature of the datasets, Canek didn't present the best mixing of batches as compared with other methods, however, the deviations could be driven by "latent variables" of batches. Then, further investigation of samples is necessary to correctly assess methods in scenarios with no gold standard. Canek is a robust methods that can be easily used within scRNA-seq joint analysis. Canek corrected batch effects in a complete example of scRNA-seq joint analysis leading to the proper characterization of cell types.

Appendix-A

Dataset	Number of cells	Technology	Reference
Jurkat cells	3,258		
HEK293T cells	2,885	10x	[68]
50:50 (Jurkat:HEK293T) mixture	3,388		
Mouse spleen	1,697	SMART-seq2	[53]
	9,552	10x	
Human pancreas	8,569	inDrop	[5]
	2,285	CEL-seq	[41]
	1,004	CEL-seq2	[17]
	638	Fluidgm C1	[30]
	2,394	Smart-seq2	[46]
Human PBMCs (Interferon beta)	13,999	10x	[26]
Human Lung	312,928	10x	[1]

Table 7.1: Public datasets used on this dissertation.

Method	Batch effect corrected output	Package version
Canek	Normalized gene expression matrix	Canek version 0.1.7
ComBat	Normalized gene expression matrix	sva version 3.38.0 [25]
ComBat-seq	Normalized gene expression matrix	sva version 3.38.0 [66]
Harmony	Normalized feature reduction vectors	Harmony version 1.0 [29]
Liger	Normalized feature reduction vectors	Liger version 0.5.0 [63]
MNN	Normalized gene expression matrix	Batchelor version 1.6.2 [18]
Scanorama	Normalized gene expression matrix	Scanorama version 1.6 [22]
scMerge	Normalized gene expression matrix	scMerge version 1.6.0 [33]
Seurat	Normalized gene expression matrix	Seurat version 3.2.2 [51]

Table 7.2: Batch effect correction methods.

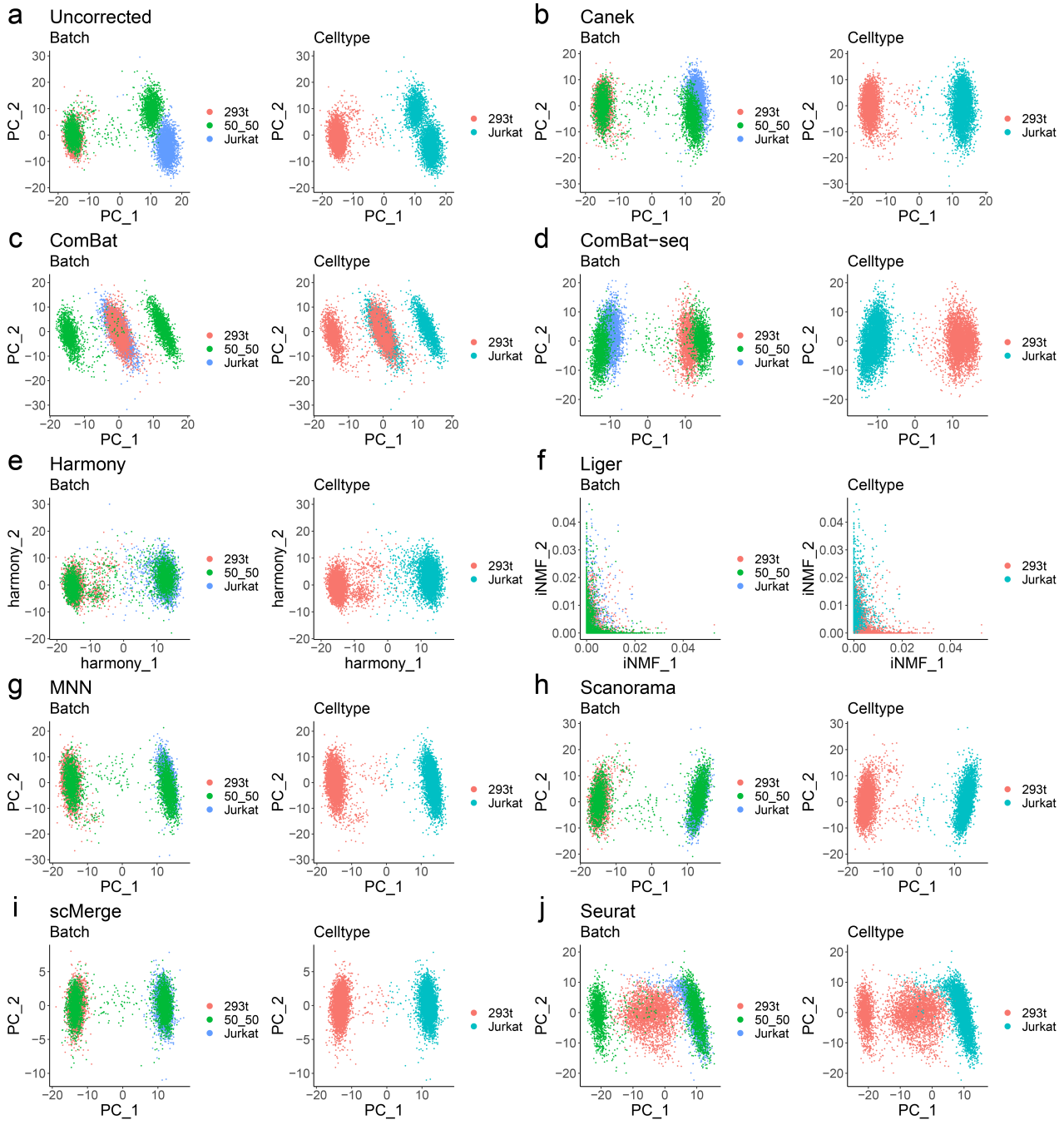

Figure A1: Integration of batches with known cell types (PCA). Three batches containing Jurkat, 293T, and 50:50 Jurkat:293T cells [68] are integrated to compare batch effect methods in a simple scenario where the biology of the data is known. Correction methods like Canek, Scanorama, and scMerge resulted in a good-mixing of Jurkat cells while preserving 293T cells. On the other hand, correction methods like ComBat, Liger, and Seurat incorrectly mixed different cell type populations.

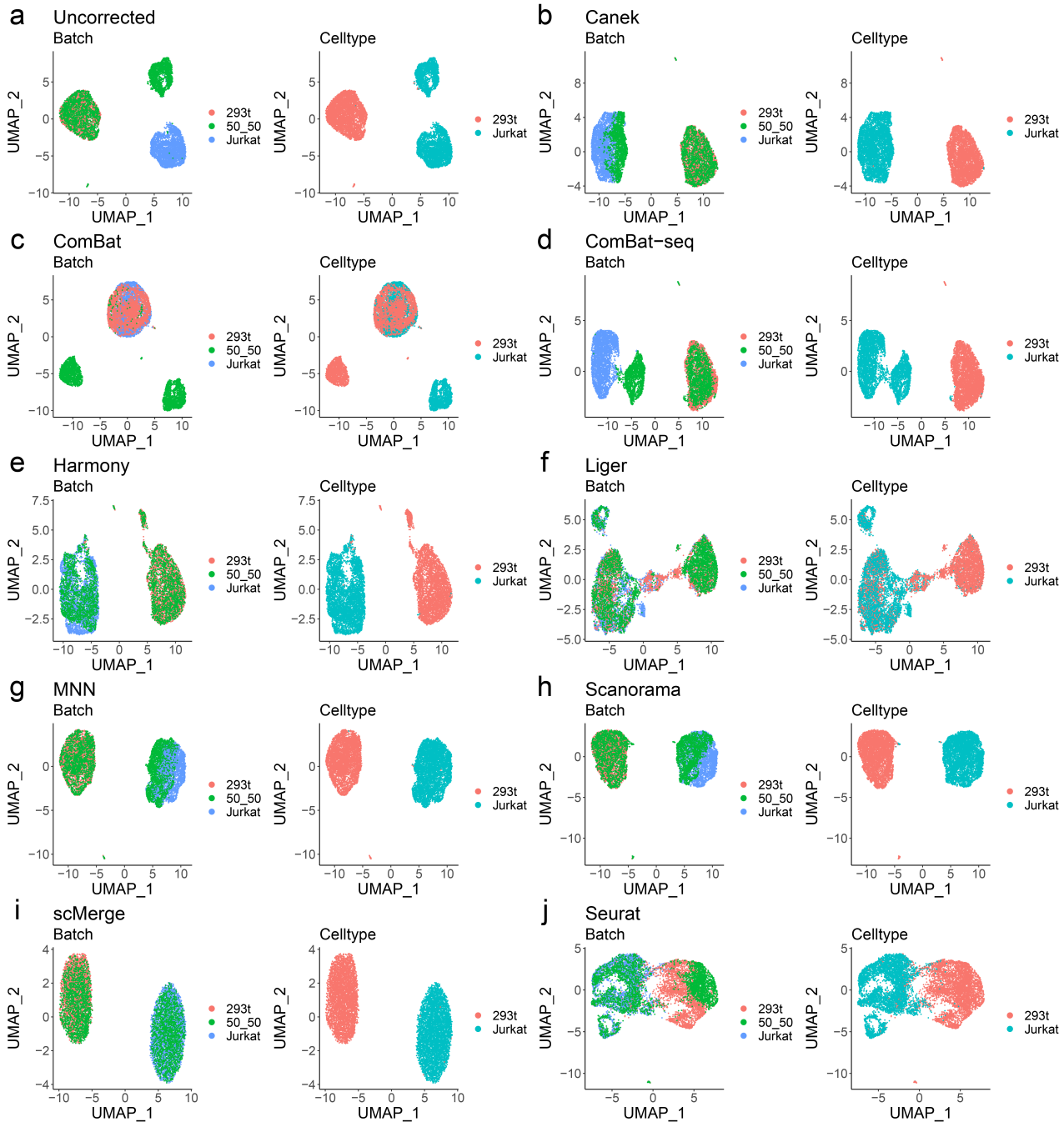

Figure A2: Integration of batches with known cell types (UMAP). UMAP plots for batch and celltype of the Jurkat/293t cells mixture dataset **a)** before and **b-j)** after batch correction.

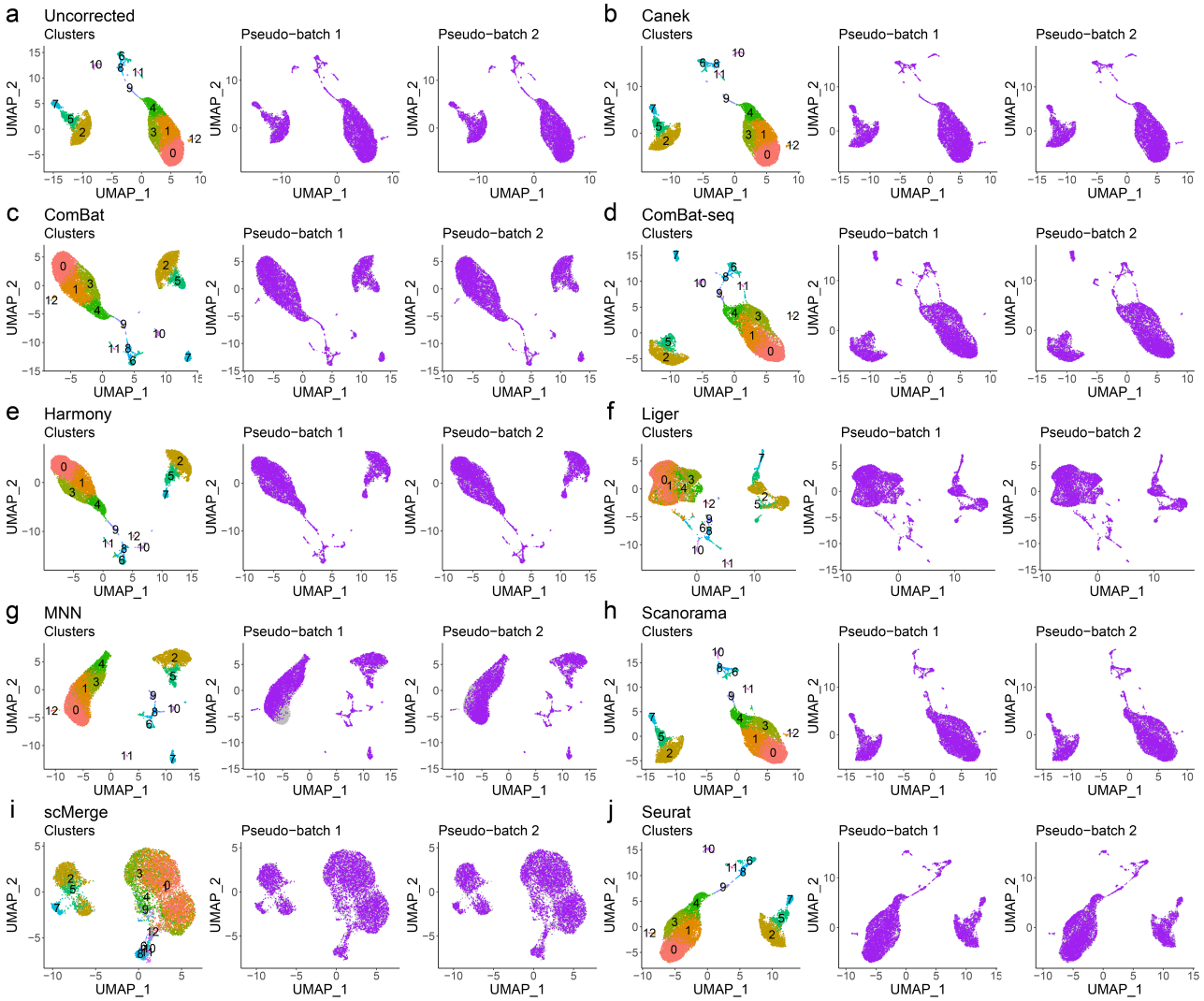

Figure A3: Batch correction result of the pseudo-batches experiment. UMAP plots for all methods in the pseudo-batch experiment.

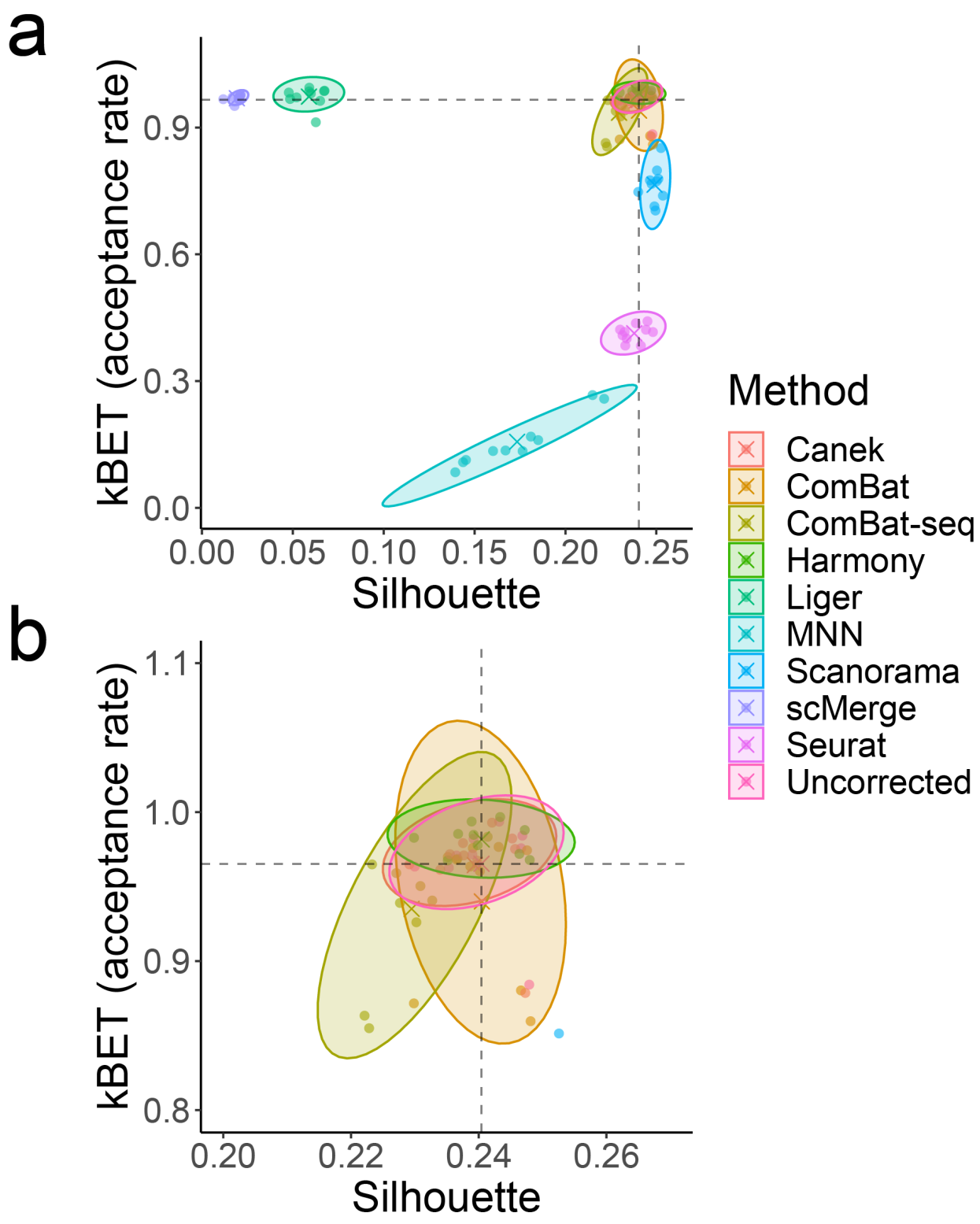

Figure A4: Results from ten pseudo-batches experiments. **a)** Silhouette vs. kBET scores for the 10 pseudo-batch experiments together with the average of each kBET/silhouette replicate. **b)** Zoom in to the methods closer to the metrics for the Uncorrected dataset (intersection of dashed lines).

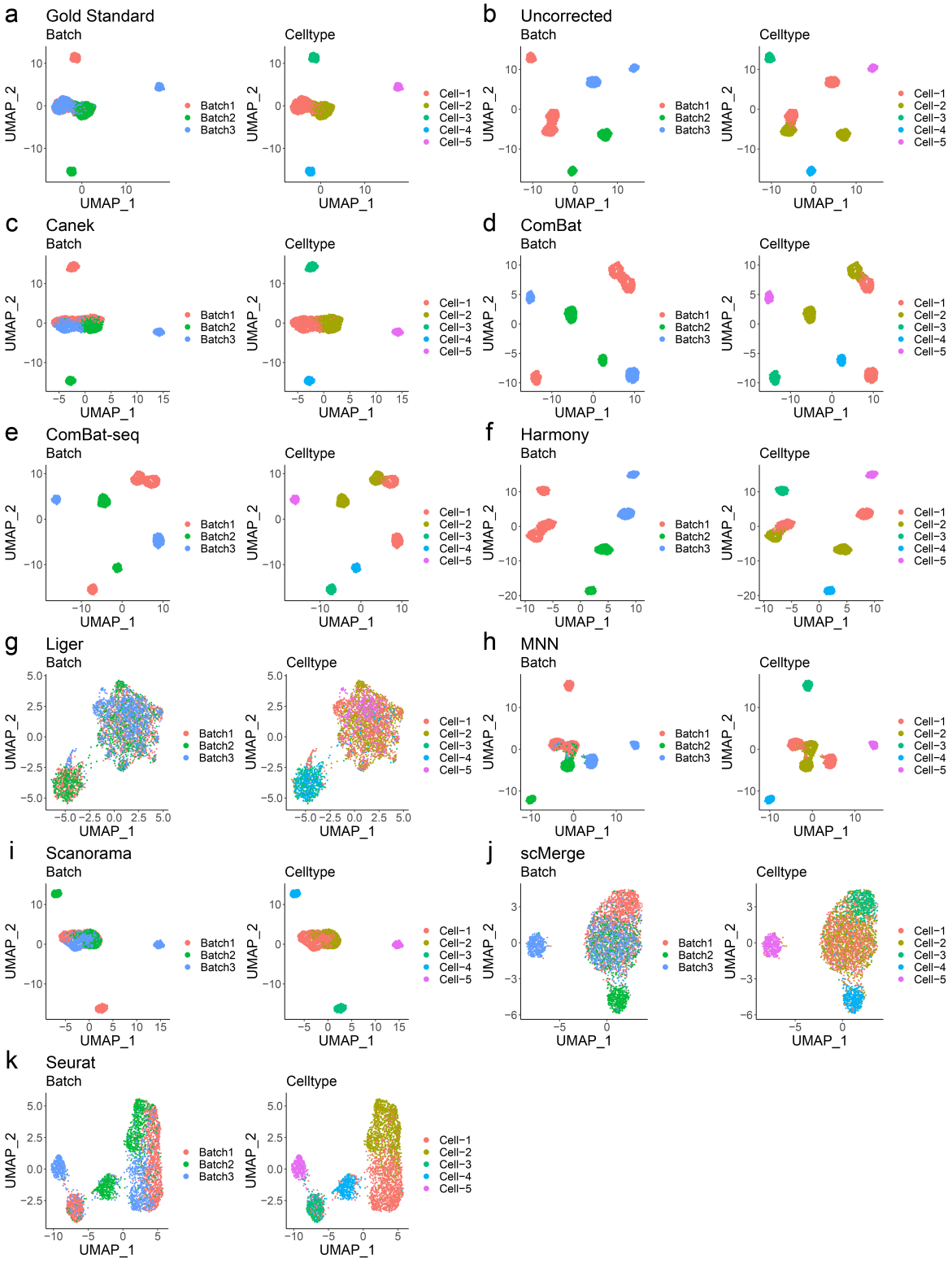

Figure A5: Gold standard (GS), Uncorrected, and corrected datasets in the scenario of known non-zero batch effects. a,b) Shared Cell-1 and Cell-2 cell types are well-mixed in the UMAP plot of the simulated GS, but batch effects in the Uncorrected data have caused cells from these cell types to group by batch. Cells in the left and right UMAP plots are colored by batch and cell type, respectively. **c-k)** Integrated data with Canek and other eight methods. Methods like Seurat, Liger, and scMerge incorrectly mixed cells from different cell types. Harmony, ComBat, and ComBat-seq failed to integrate the batches, where cells from the same cell type still cluster by batch. Canek, correctly preserved batch-specific cell types while correcting shared ones.

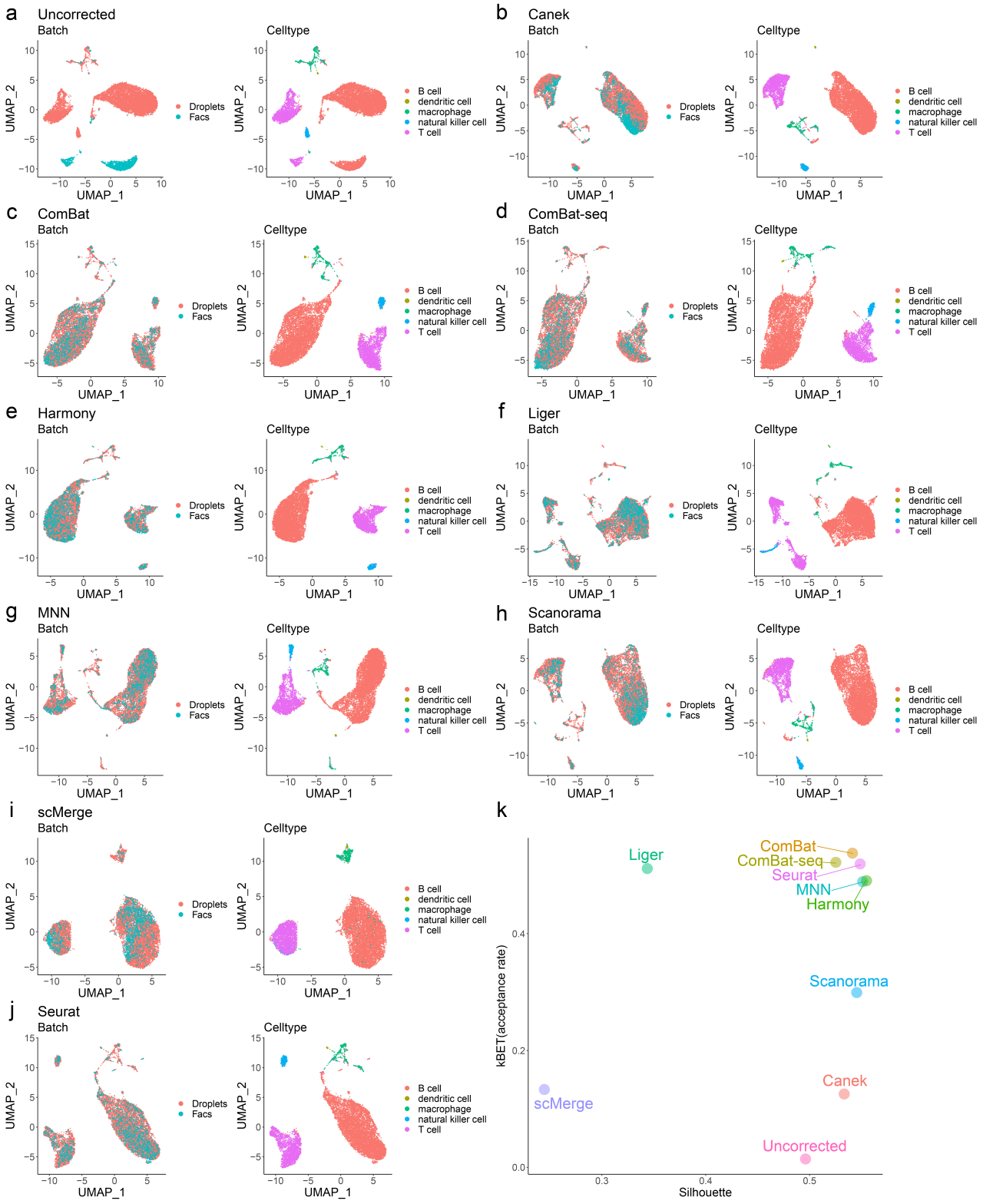

Figure A6: Integration of Tabula Muris spleen datasets The Tabula Muris spleen Facs and Droplets datasets were integrated with Canek and other eight methods. This dataset presents the scenario where the same sample were sequenced with different technologies. *a)* UMAP plots of the batches before integration. Batch effect causes cells to group by batch rather than cell type. **b-j** UMAP plots after batch effect correction. Except for scMerge and Liger, all the methods successfully integrated the datasets while preserving cell type annotations. **k)** kBET and Silhouette metrics supports the observed results, where Silhouette score penalized scMerge and Liger insufficient preservation of cell types.

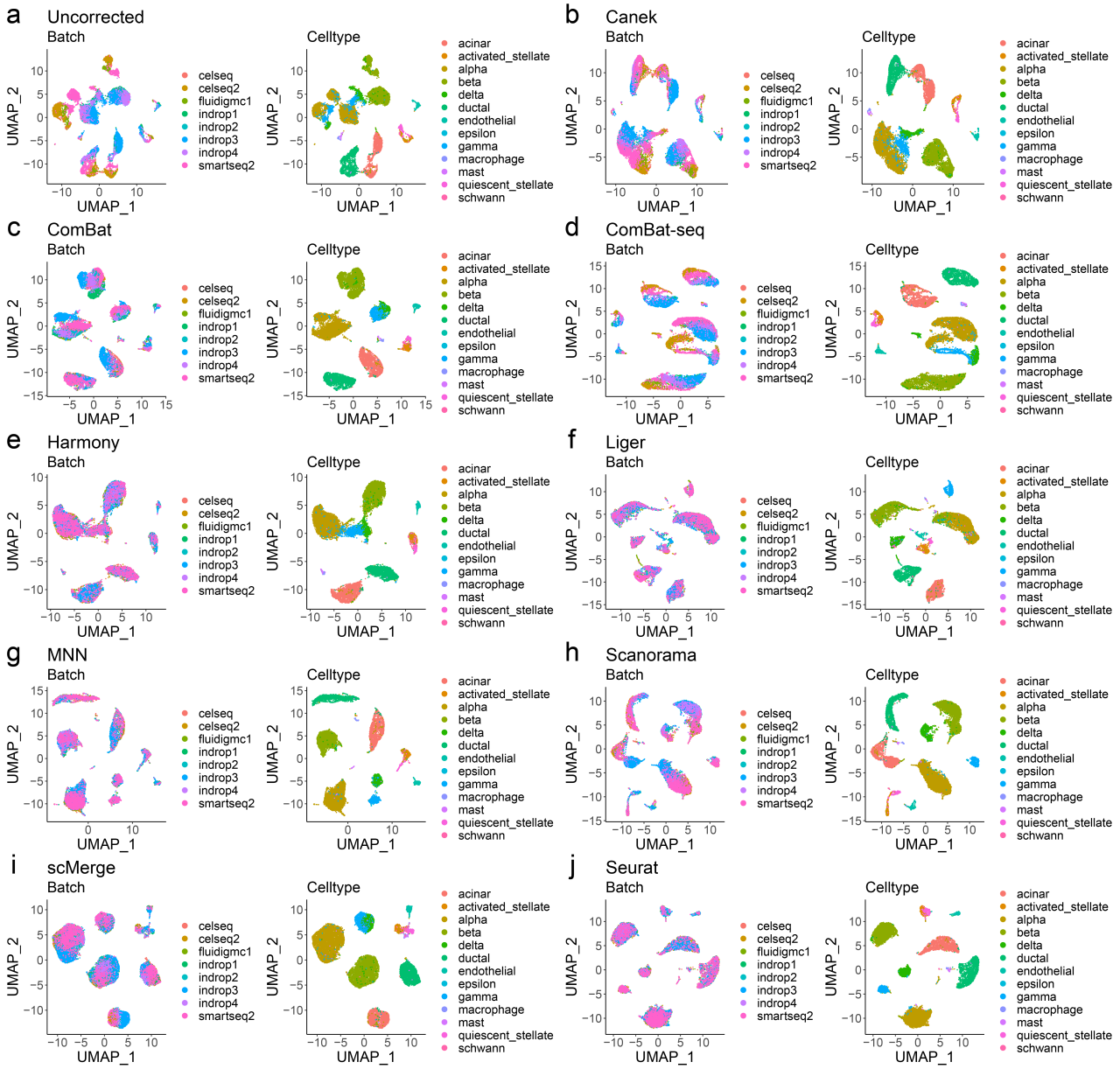

Figure A7: Integration of batches from the same tissue sequenced with different technologies at different laboratories. Integration of eight human pancreatic islet datasets sequenced with different technologies at different laboratories. **a-j)** UMAP plots of Uncorrected and integrated data with Canek and the other eight methods, where cells were colored by batch and by the cell type annotations provided in the *SeuratData* R package [44]. Seurat, Liger, and Harmony minimized batch differences. The remaining methods, including Canek, integrated the batches with different mixing levels. Because of the lack of a gold standard, it is difficult to assess methods.

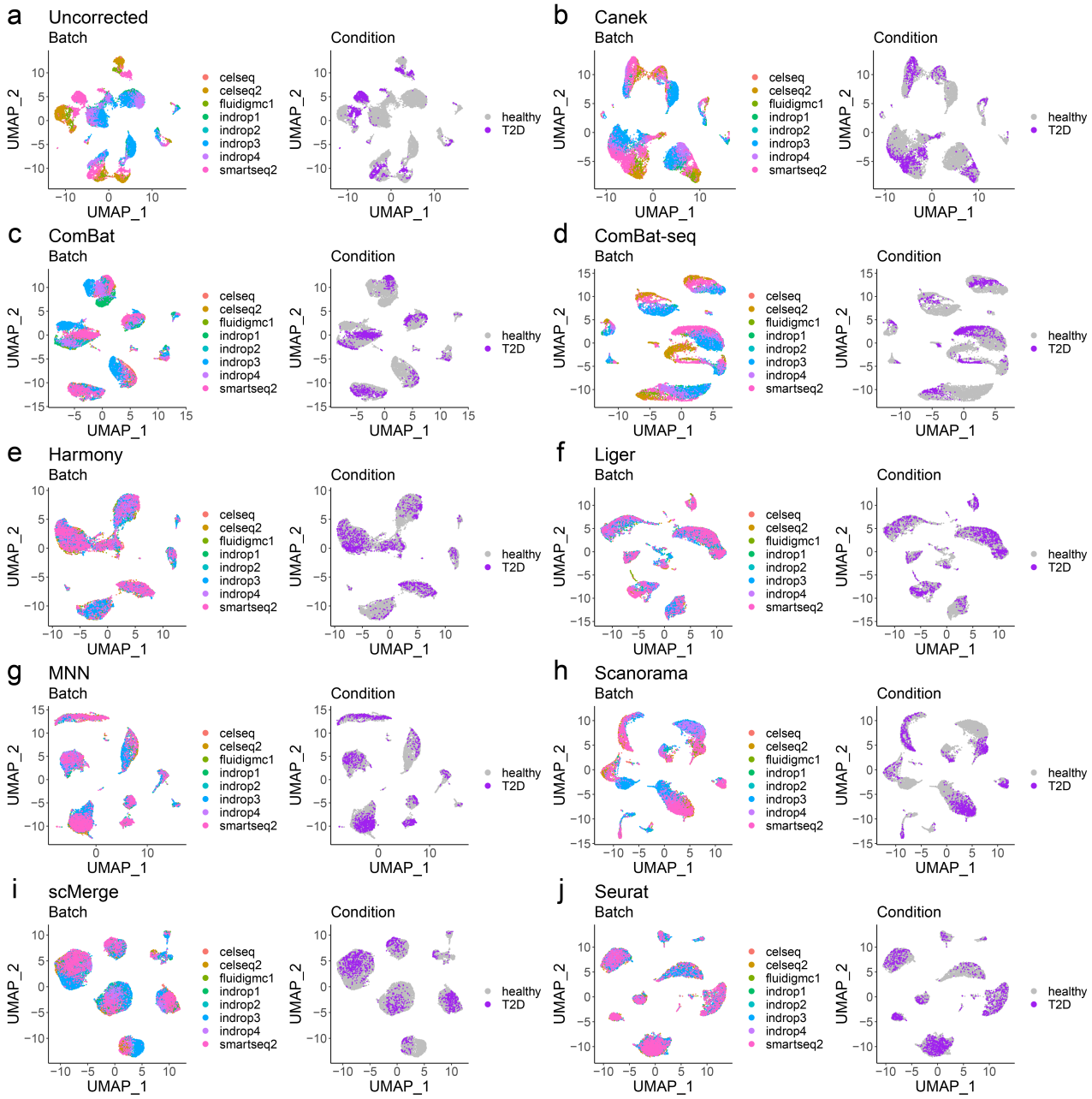

Figure A8: Integration of batches from the same tissue sequenced with different technologies at different laboratories (different conditions). Integration of eight human pancreatic islet datasets but different conditions sequenced with different technologies at different laboratories. **a-j)** UMAP plots of Uncorrected and integrated data with Canek and the other eight methods, where cells were colored by batch and by two disease conditions: healthy and type 2 diabetes (T2D). Methods like Canek and Scanorama facilitate comparisons between cells from different conditions. Contrastingly, samples differences are difficult to distinguish from methods like Harmony and Seurat.

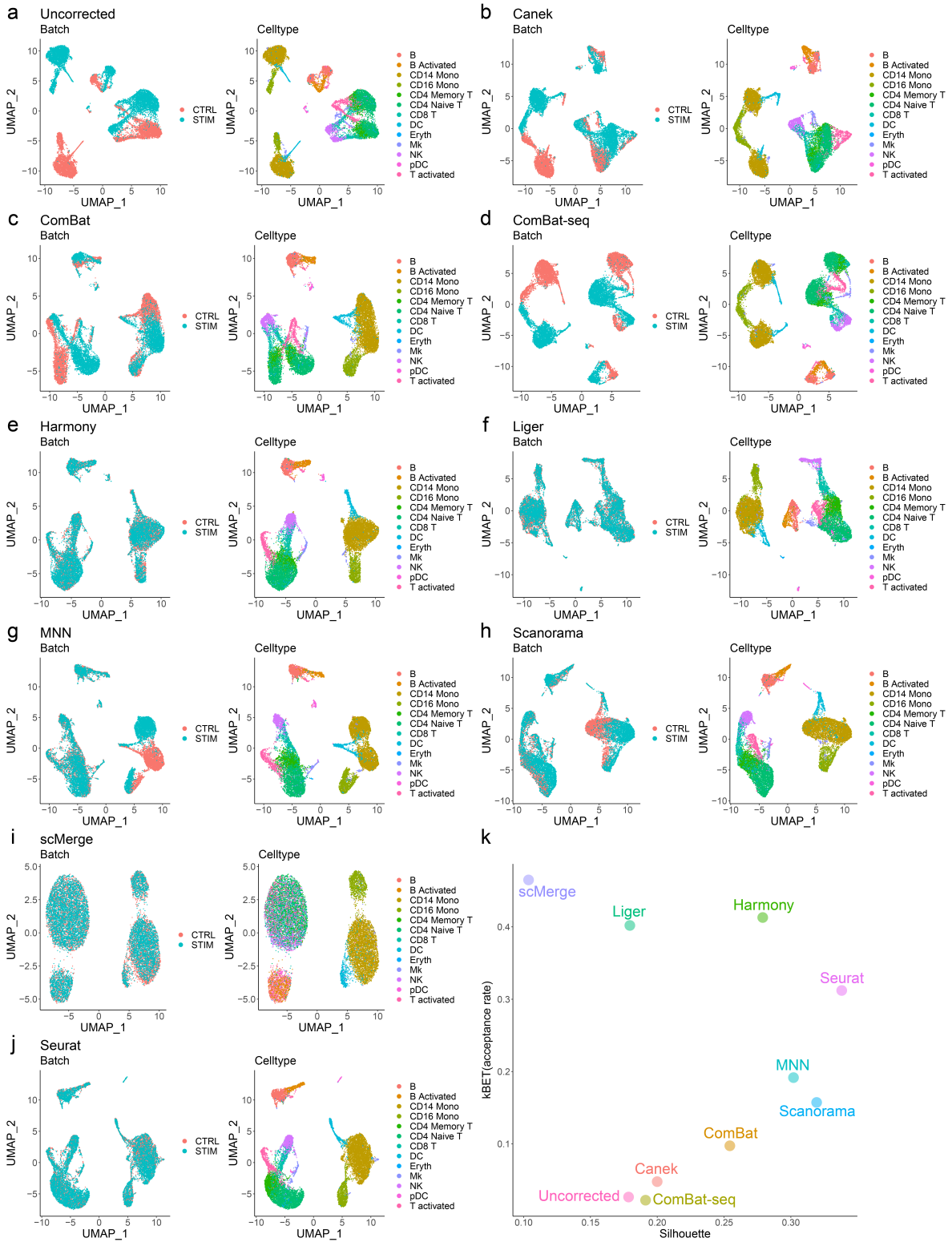

Figure A9: Integration of batches from different conditions sequenced with the same technology. Two PBMCs datasets with and without interferon-beta stimulation [26] were integrated. **a-j)** UMAP plots of Uncorrected and integrated data with Canek and the other eight mehods. Un-stimulated and stimulated cells were labeled as CTRL and STIM, respectively. Seurat, Liger, scMerge, and Harmony perfectly integrated the batches, which is supported by their high kBET acceptance rates **k)**. However, this contradicts previous results[21] suggesting over-correction of these methods. Conversely, methods like Canek, Combat, and MNN preserved strong differences in monocytes as expected.

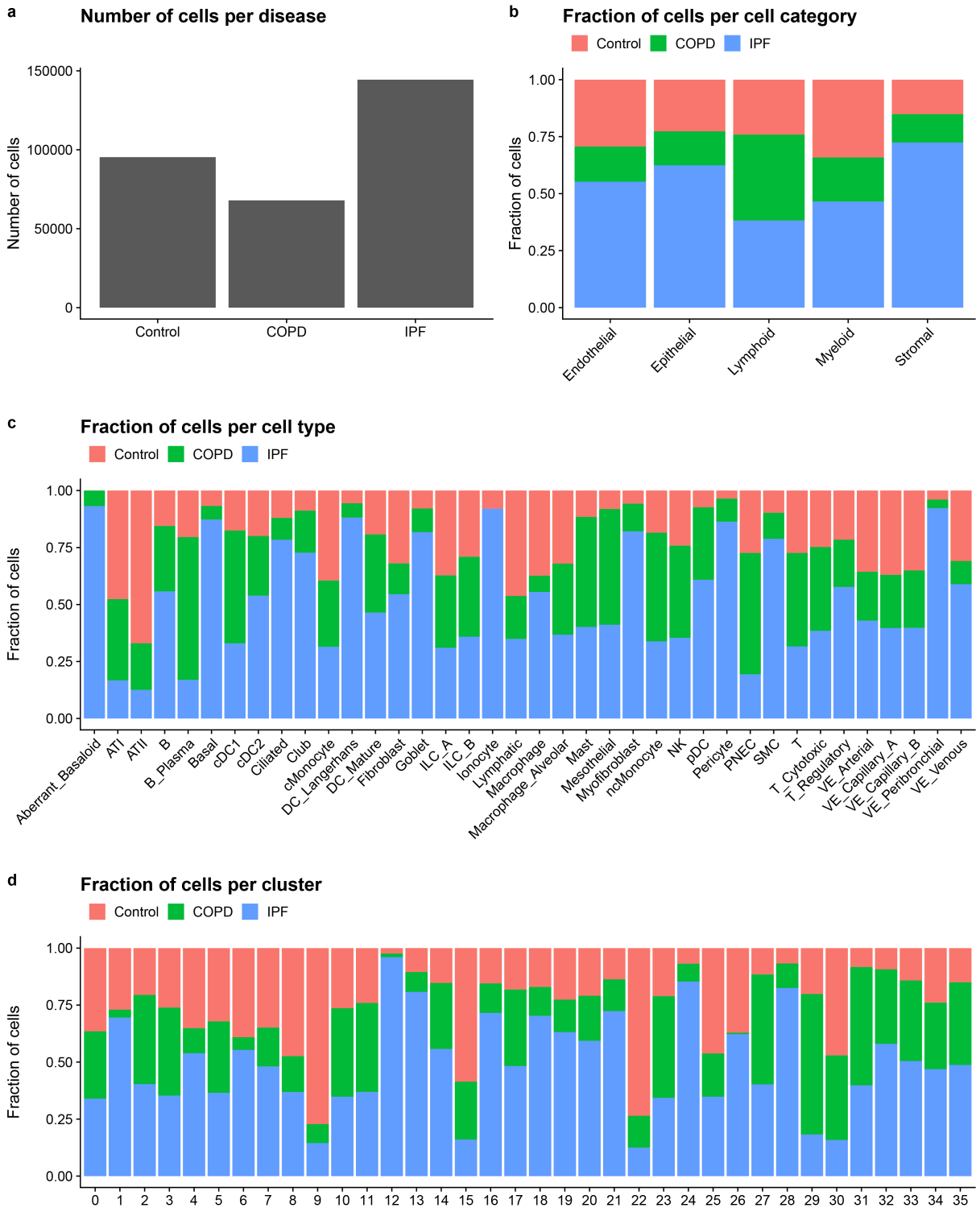

Figure A10: Appendix-A Figure A10. Enrichment of cell populations. **a)** Number of cells per disease. **b)** Fraction of disease cells per cell type category. **c)** Fraction of disease cells per cell populations, captures changes in cell populations in agreement with those reported in the original publication (Adams et al., Science, 2020). For example, there is enrichment of airway epithelial cells (i.e, Basa, Ciliated, and Goblet cells) and depletion of alveolar epithelial cells (i.e., ATI, AT2) in IPF donors. Likewise, the reported Aberrant-Basaloid, cells are enriched in IPF. **d)** Fraction of disease cells per cluster. This shows that enrichment of IPF cells in specific clusters of interstitial and alveolar macrophages (e.g., clusters 12 and 16).

Appendix-B

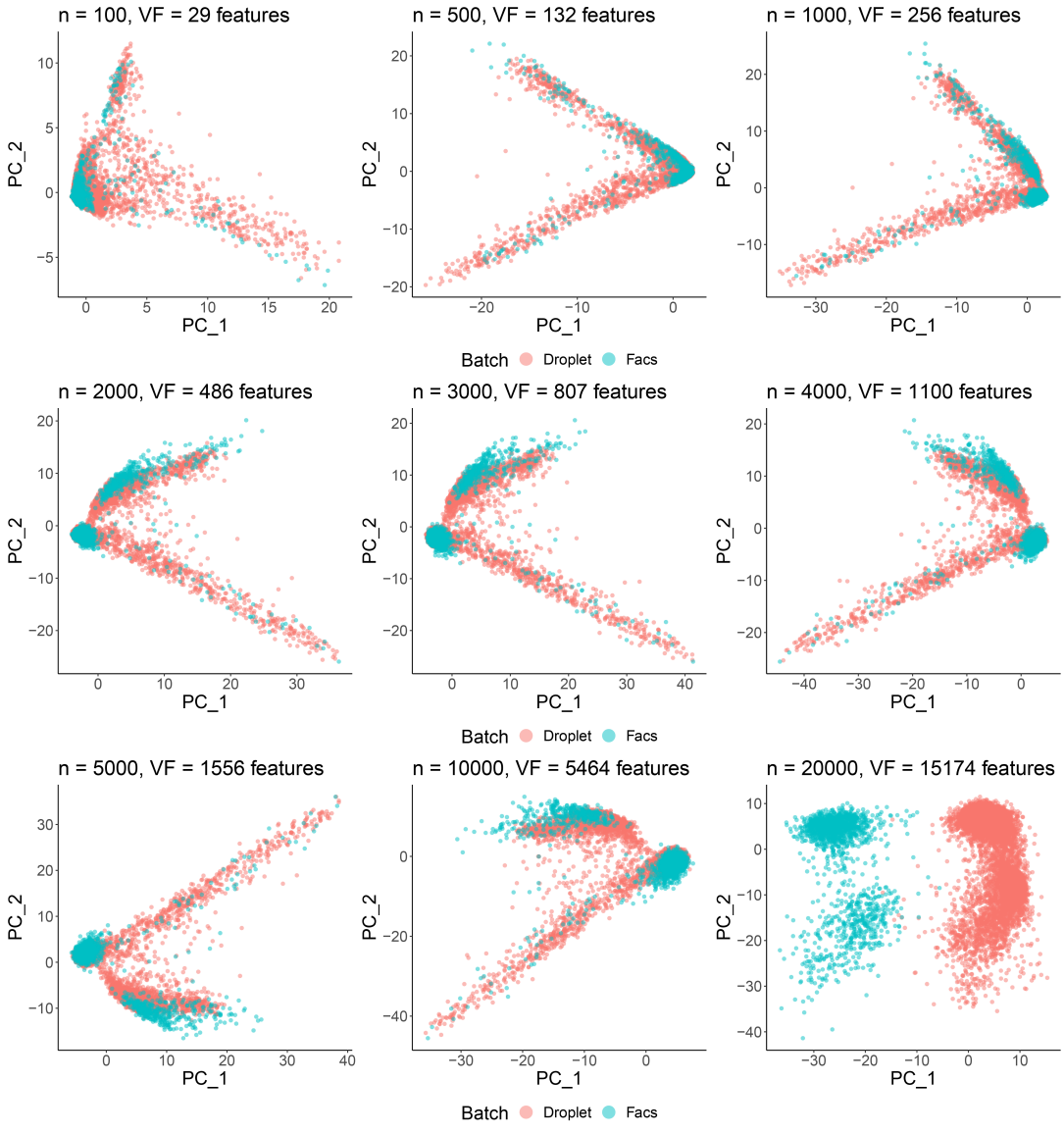

Figure B1: Effect of variable features (VF) selection. *Incremental-intersection* approach to select VF. The the top n features for each batch are independently found. VF would be those genes that belong to the intersection of these two subsets. Using a low number of n, does not produce well-defined groups in the PCA. Using a large n increases differences between batches.

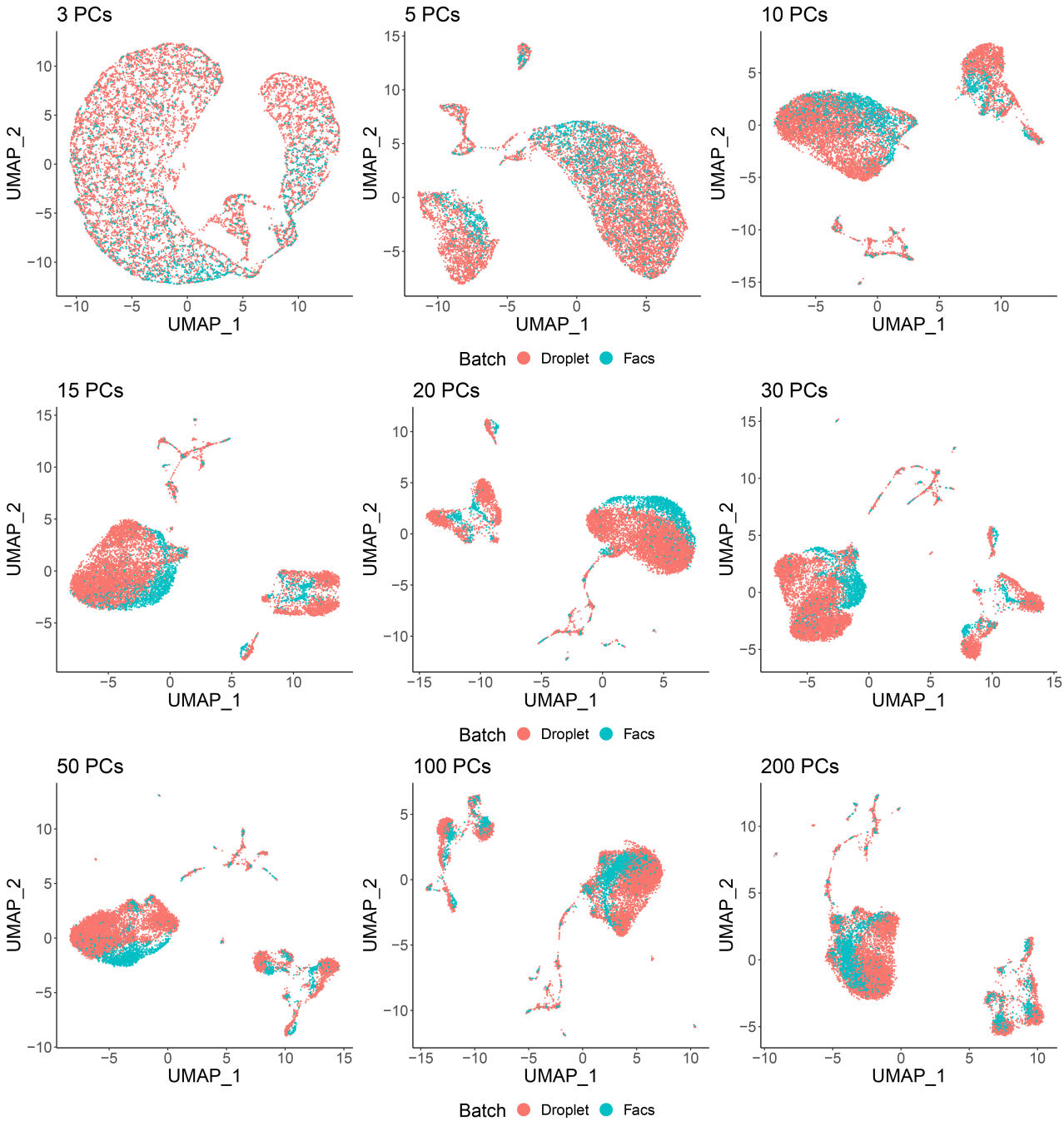

Figure B2: Effect of using different numbers of principal components in UMAP. Uniform Manifold Approximation and Projection (UMAP) representations using increasing numbers of principal components (PC). Using a low number of PCs improved cell mixing between batches but cells diversity was not properly captured. Using a large number of PC captured more complex structures of cells but batch differences increased.

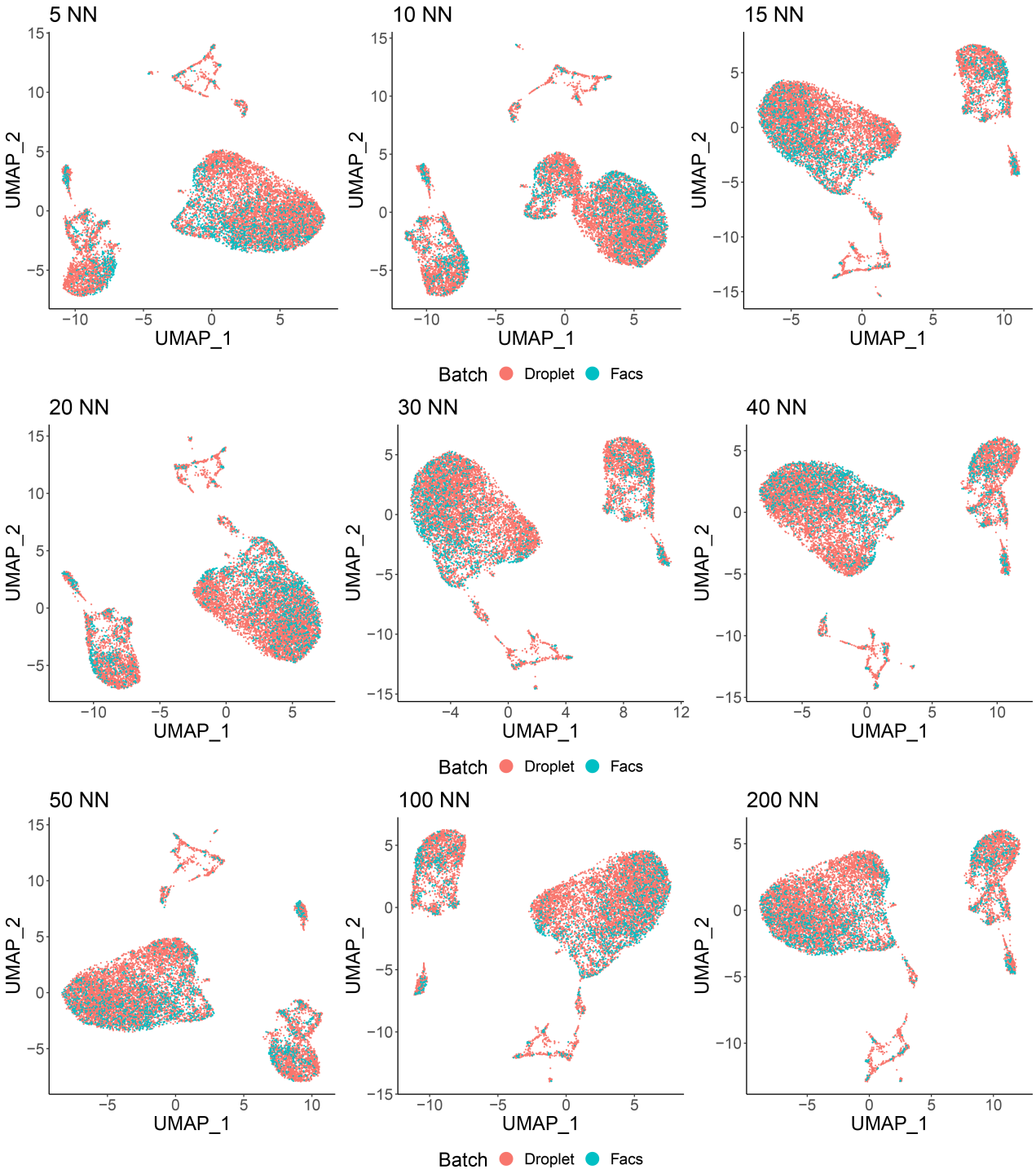

Figure B3: Effects of changing the number of PC in the integration of the datasets. The batches were integrated using increasing numbers of PC while fixing the number of NN to the default value (30 NN). Except for the plot using ”10 PC”, the results presents similar UMAP representations, proving Canek’s robustness to changes in the number of PC on this test.

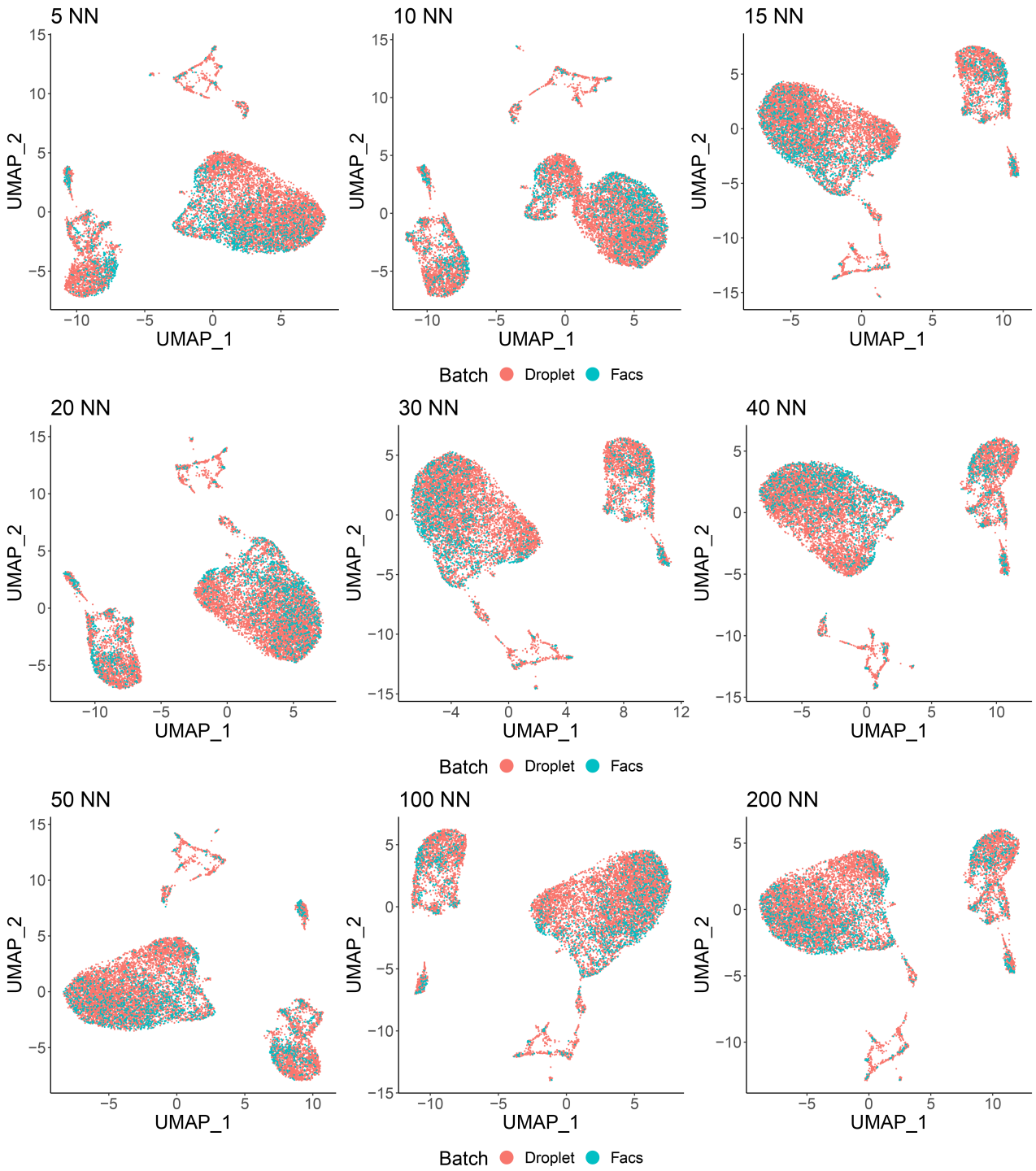

Figure B4: Effects of changing the number of NN in the integration of the datasets. The batches were integrated using increasing numbers of nearest neighbors while fixing the number of PC to the default value (50 PC). The change in the structure of cells in the UMAP plots using ”10 NN” could be caused the non-linear behavior of UMAP. The results are similar, proving Canek’s robustness to changes in the number of NN on this test.

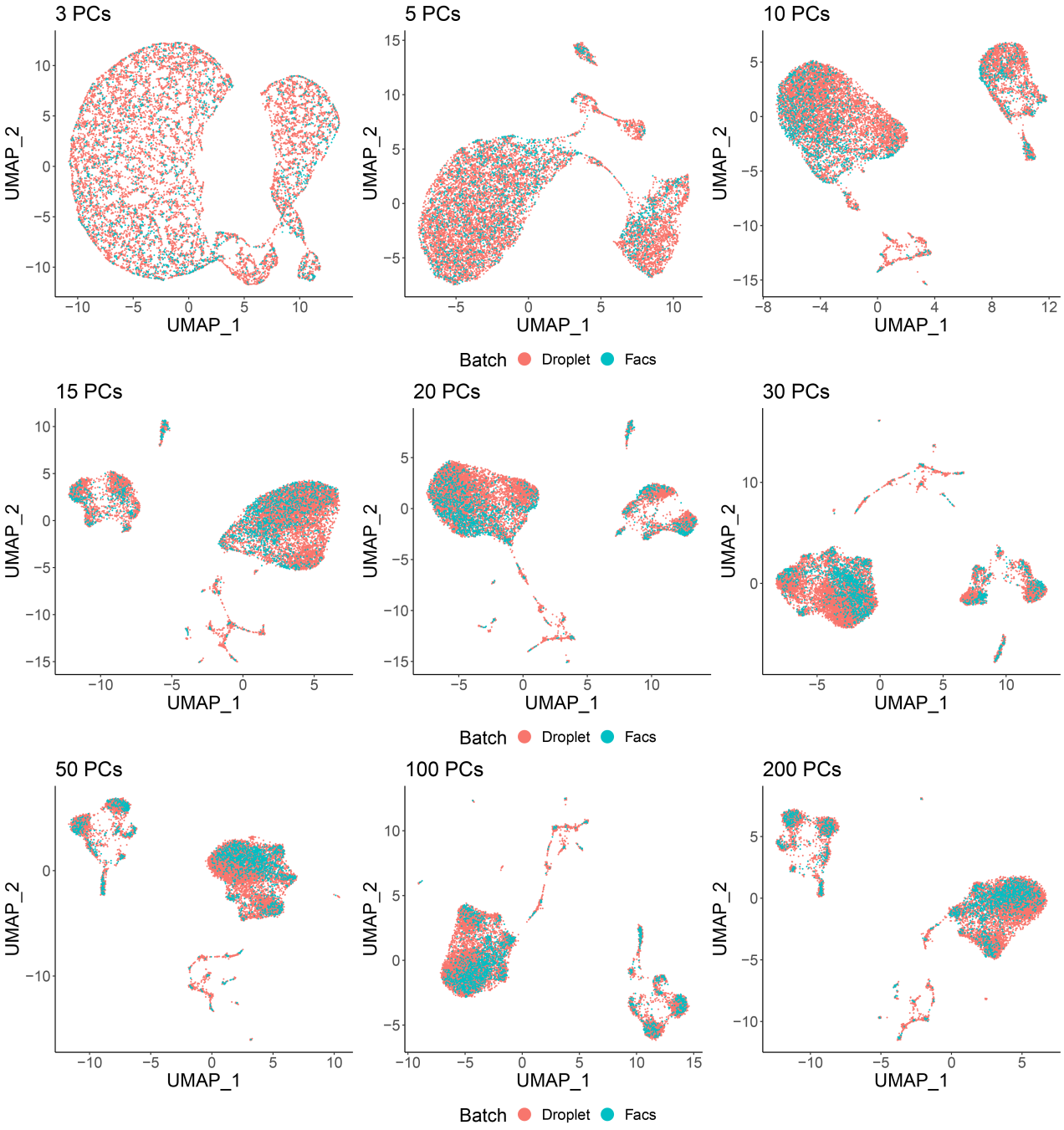

Figure B5: Effect of using distinct numbers of principal components in UMAP results. UMAP plots for different number of PCs after integrating the Tabula Muris spleen datasets with Canek. UMAP plots using 3 and 5 PCs present simple structure of groups as compare with plots obtained using 10 PCs or more. UMAP plots using more than 20 PCs are similar.

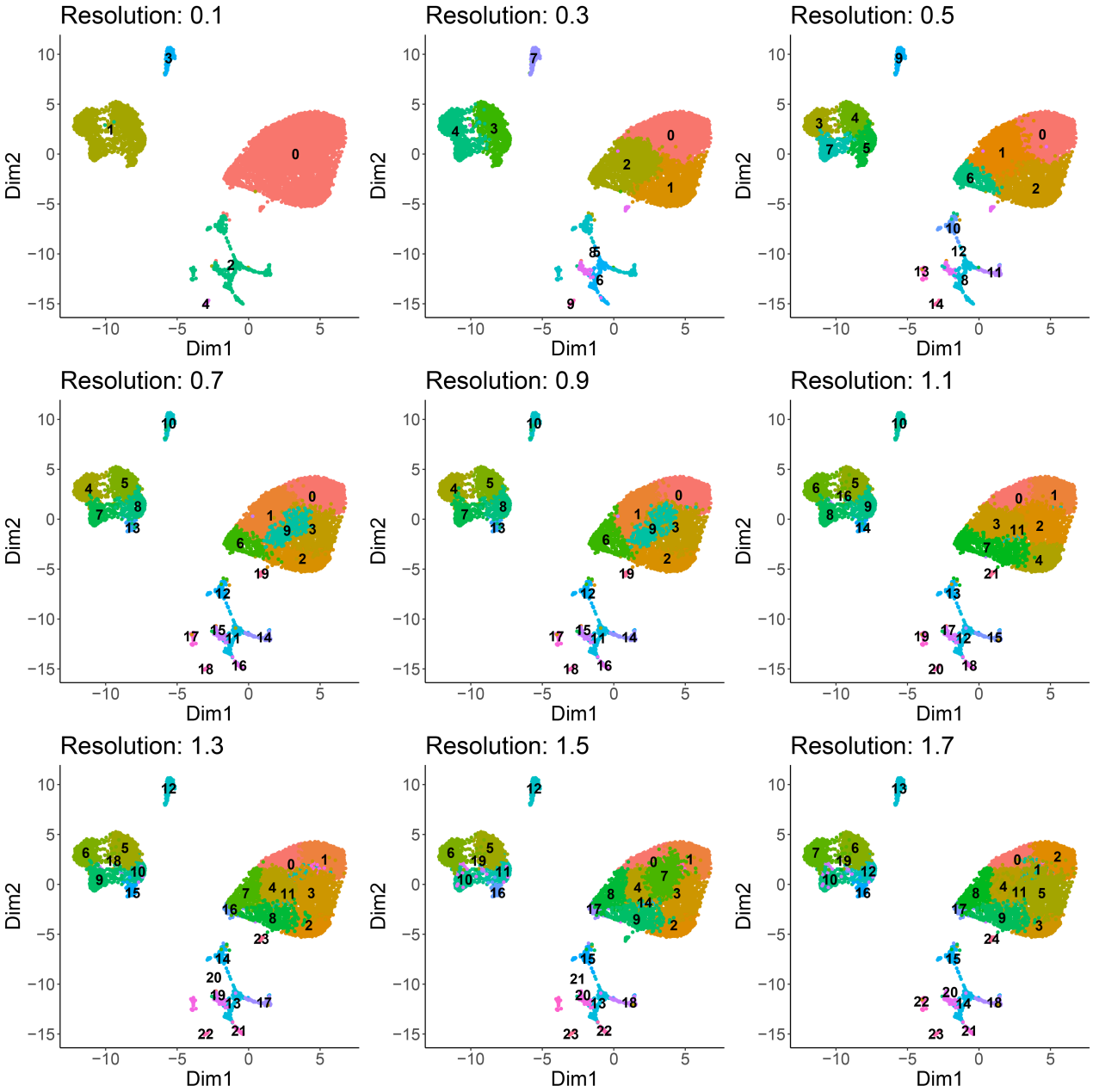

Figure B6: Clustering results using the Louvain algorithm with increasing resolutions. UMAP plots with the color of dots representing the results of Louvain with resolutions ranging from 0.1 to 1.7. Using a relatively low resolution as 0.1, and 0.5, allows the identification of well-defined communities. On the contrary, using high resolutions increases the number of clusters with close distributions on the UMAP, making it difficult to differentiate between clusters, e.g. if using a resolution of 1.7, clusters 4, 5, and 11 are almost merged.

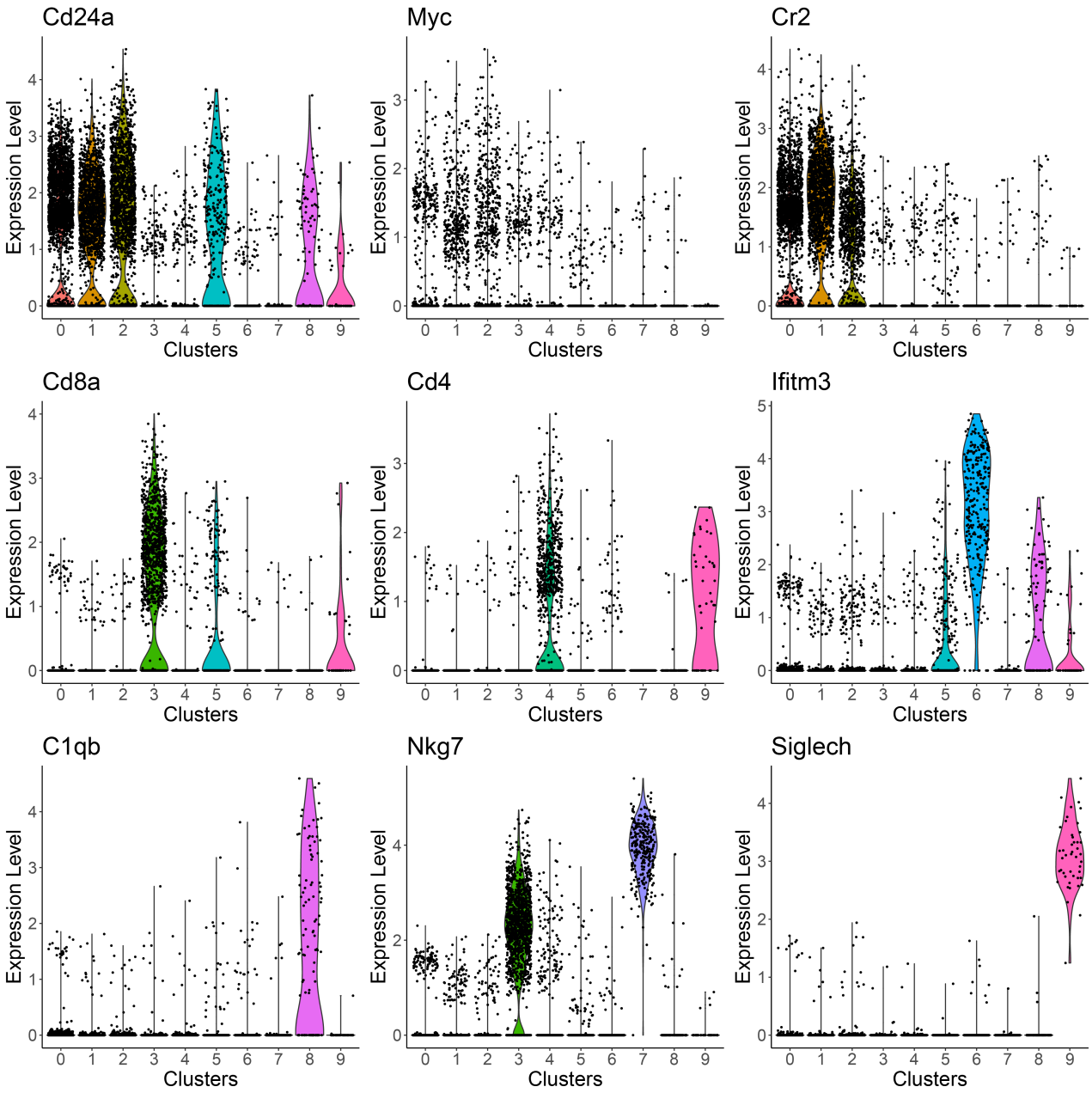

Figure B7: Violin plots for medium-level annotation of cells. Ten clusters were recovered the using medium-level clustering. Clusters 0, 1, and 2 (*B cells*) expressed different levels of *Cd24a*, *Cr2*, and *Myc* genes. Two groups of *T cells* were characterized by the expression of the markers *Cd8a* and *Cd4*. Different expression levels of *Cd24a*, *Ifitm3*, and *C1qb* allowed the identification of *Monocytes, Macrophages*, and *Monocytes C1qb*. Cells expressing *Nkg7* and *Siglech* were labelled as *Natural Killer cells* and *Dentritic cells*, respectively.

List of academic accomplishments

Publications

- *Unbiased integration of single cell transcriptome replicates,* M. J. Loza-Lopez, S. Teraguchi, D. M. Standley, D. Diez, bioRxiv, 2021.

- *Methods for sequence and structural analysis of B and T cell receptor repertoires,* S. Teraguchi, D. S. Saputri, M. A. Llamas-Covarrubias, A. Davila, D. Diez, S. A. Nazlica, J. Rozewicki, H. S. Ismanto, J. Wilamowski, J. Xie, Z. Xu, M. J. Loza-Lopez, F. J. Eerden, S. Li, and D. M. Standley, Computational and Structural Biotechnology Journal, Elsevier, Vol 18, 2020, pp 2000-2011. *In this journal publication, I collaborated by describing the main approaches to describe B and T cell repertoires using single cell technologies*.

Poster presentations at international conferences

- *Integration of single-cell transcriptome replicates using linear estimation and fuzzy logic*, M. J. Loza-Lopez, S. Teraguchi, D. M. Standley, and D. Diez, Human Cell Atlas Asia, November 2021.

- *Batch effect correction of single-cell experimental data using MNN and Extended Kalman Filter*, M. J. Loza-Lopez, D. Diez, EMBO workshop on single cell biology, Tokyo, Japan, May 2019.

NOTE: Currently, the *Unbiased integration of single cell transcriptome replicates* manuscript is under the second peer-review for its publication in the *NAR Genomics and Bioinformatics* journal. The comments and answers from the review are:

Reviewer 1

Comment 1: (From a previous review) We are happy with the authors' reply to our comments and with the additional work provided.

Reviewer 2

Comment 1: The authors present a new computational method for the integration of single-cell datasets, implemented as an open-source R package called ”Canek”. Canek first matches samples between datasets using mutual nearest neighbors (MNN), then attributes to each sample a ”correction vector” based on this matching plus a clustering information. There is no mention of using Canek to integrate single-cell datasets across modalities.

Mutual nearest neighbors is a state-of-the-art method to perform cell-cell matching, used in many existing data integration pipelines (Seurat, Harmony...). The originality proposed here is to cluster samples so that we obtain a correction vector per cluster. Then, each sample is attributed a weighted sum of these correction vectors depending on its relative position to the clusters.

Canek seems to perform similarly to other methods in the field, which is already quite crowded. According to the authors, it is the method that introduces the least amount of bias among their benchmarking datasets, using as a proxy for the bias a combination of kBET and silhouette metrics preservation. This is indeed encouraging, even if defining comprehensive metrics to assess integration quality is still an open and challenging problem in the field.

Response to comment 1: We appreciate the careful reading of our manuscript and were happy that the reviewer found merit in the assessment of bias, which we agree is challenging to quantify.

Comment 2: A typical issue we encounter when dealing with dataset integration is the rapid diminution of common genes between datasets, especially when integrating several tens of datasets together. Could you elaborate on this point, and possible solutions Canek could bring?

Response to comment 2: We agree that this is a very important issue. Currently, Canek does not provide a specific way to approach this problem. Indeed, how to choose the appropriate set of genes to determine cell similarity is not specific to the integration method. When the number of samples is small and the expected cell populations in each sample match, using common genes seems reasonable. As the reviewer indicates, when the number of samples is very large or we expect strong differences in cell populations between samples, the number of intersecting genes drops drastically, and a different approach must be used. By default, Canek uses the method implemented in the Seurat package. In this method, a list of highly variable features (HVF) is calculated first for each batch independently. Then, the features are ranked based on the number of times they are among the top HVF (nfeatures=2,000 features by default) in each batch. Finally, the top nfeatures in this ranking are selected. This method will first favor features found in all batches, then those in all but one, etc. Although not perfect, we think this approach is reasonable and flexible enough for most applications. We modified the manuscript to address this issue (the modified paragraph is

included in our response to the reviewer's comment about Euclidean distance, since both are related).

Comment 3: Canek, as many other methods in the field necessitates the choice of a reference. This can be an issue when datasets are of comparable quality, or when the goal is exactly to create a reference by aggregating a large amount of datasets. Choosing a reference is a high source of bias, could you discuss how to deal with it so that Canek stays "unbiased"?

Response to comment 3: The choice of reference can be important for some methods. For example, we saw this with the Jurkat dataset. Some methods like Seurat were quite sensitive to the order of the samples, indicating that the choice of reference might be very important. By default, Canek performs hierarchical optimization that aims to choose the order of batches, making the reference the one with the largest number of cells. The query batch is selected as the batch which shares the highest number of MNN pairs with the reference, an indicator of having the largests number of similar cells with the reference. Unfortunately, the current approach is not feasible when the number of batches is very large, like in the human lung disease dataset used in our manuscript, for which we turned it off. It should be straightforward to implement an alternative clustering method that scales with larger datasets in the future. However, in our tests, Canek results were not strongly affected by the order/choice of reference. One reason for this might be that Canek does not perform strong corrections. Another reason might be that when a cell population does not have a match in the reference dataset, there is no correction applied. We have added the following text to the hierarchical section in the methods:

The choice of reference batch may have a strong effect on the integration. To address this potential problem, we carry out hierarchical integration when there are more than two input batches. The purpose is to determine the optimal order of batches. First, we define the reference as the batch with the largest number of cells. Then, the query batch is chosen to be the batch sharing the highest number of MNN pairs with the reference, an indication of higher similarity between batches. To this end, we obtain the first three PCs using the prcomp_irlba function in the irlba R package (12), find the MNN pairs and select the query batch as the one with the highest number of pairs.

Comment 4: Averaging correction vectors corresponds to minimizing a sum of Euclidean distances, which is known to behave poorly in high dimensional spaces. Could you discuss why it still seems to be a decent strategy given results on your benchmarks?

Response to comment 4: Thank you for raising this point. In fact, the effective dimensionality used during integration is much smaller than the original dimensionality. First, the top 2,000 highly variable features (HVF) are selected from the dataset. Second, these 2,000

HVF are subjected to principal component analysis. The top 50 principal components are then used to perform cell matching and clustering; subsequently, the correction vectors are formed by independent estimations of the batch effect on each gene. For the fuzzy integration step, the top 10 principal components are used. By performing dimensionality reduction and focusing on the combination of features that contribute the most to the variability in the data, we effectively alleviate most of the problems associated Euclidean distances in higher dimensions. We modified the following paragraph in the methods to clarify this point (and the related issue of the common features):

To speed up computation and reduce the potentially negative impact of Euclidean distance metrics in high dimensions, we take several steps. First, we focus on the top (2,000 by default) most highly variable features (HVF). Ideally, we want to choose features common to all batches, but the number of common HVF diminishes drastically as the number of datasets increases. Therefore, we used the method implemented in the Seurat package in FindVariableFeatures. In this method, a list of highly variable features is calculated first for each batch independently. Then, the features are ranked based on the number of times they are among the top HVF in each batch. Finally, the top features in this ranking are selected. This method will first favor features found in all batches, then those in all but one, etc. Although not perfect, we think this approach is reasonable and flexible enough for most applications. Next, we use the HVF to calculate the first 50 principal components (PCs) (11) using the prcom_irlba function from the irlba R package (12). This lower dimensional space is used to identify MNNs, and in the clustering and fuzzy correction steps. However, during the calculation of the correction vector step, we use the original input datasets. By focusing on the combination of features that contribute the most to the variability in the data and performing dimensionality reduction, we effectively alleviate most of the problems associated with Euclidean distances in high dimensions.

Comment 5: A very hot topic in dataset integration today is integration across modalities, for instance RNA-seq vs ATAC-seq, if possible without prior information about sample-sample matching. Do you think Canek could be extended to such applications?

Response to comment 5: The reviewer points to a very interesting topic. We haven't thoroughly tested Canek with other modalities. We can expect Canek to perform similarly, although the much lower number of features in the case of CITE-seq could hinder the cell matching procedure. For the integration of different modalities when there is no sample-sample matching, the main problem is to find which cells in one modality are close to other cells in the other modality. We do not think Canek can currently be applied to such scenario, but we will consider that possibility in the future. A different problem would be the integration/batch correction of multi-modal, perhaps replicated, datasets. A simple solution to this problem could be to perform batch correction with Canek in each modality separately and then integrate the modalities, e.g., with the weighted-nearest neighbor method implemented

in the Seurat package. We will investigate this idea in future versions of Canek and have added the following paragraph in the Discussion:

Single cell genomics datasets performed on other modalities like protein (CITE-seq) and chromatin accessibility (scATAC-seq) are becoming more popular. Furthermore, integration of datasets with multi-modal (e.g., RNA + ATAC) measurements are beginning to appear. We are working to extend Canek's approach to the integration of multi-modal datasets.

Bibliography

[1] T. S. Adams, J. C. Schupp, S. Poli, E. A. Ayaub, N. Neumark, F. Ahangari, S. G. Chu, B. A. Raby, G. DeIuliis, M. Januszyk, Q. Duan, H. A. Arnett, A. Siddiqui, G. R. Washko, R. Homer, X. Yan, I. O. Rosas, and N. Kaminski. Single-cell rna-seq reveals ectopic and aberrant lung-resident cell populations in idiopathic pulmonary fibrosis. *Sci Adv*, 6(28):eaba1983, 2020.

[2] Robert A Amezquita, Aaron TL Lun, Etienne Becht, Vince J Carey, Lindsay N Carpp, Ludwig Geistlinger, Federico Marini, Kevin Rue-Albrecht, Davide Risso, and Charlotte Soneson. Orchestrating single-cell analysis with bioconductor. *Nature methods*, 17(2):137–145, 2020.

[3] S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome Biol*, 11(10):R106, 2010. Anders, Simon Huber, Wolfgang eng Research Support, Non-U.S. Gov't England Genome Biol. 2010;11(10):R106. doi: 10.1186/gb-2010-11-10-r106. Epub 2010 Oct 27.

[4] R. Argelaguet, A. S. E. Cuomo, O. Stegle, and J. C. Marioni. Computational principles and challenges in single-cell data integration. *Nat Biotechnol*, 2021.

[5] M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein, D. A. Melton, and I. Yanai. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst*, 3(4):346–360.e4, 2016.

[6] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[7] P. Brennecke, S. Anders, J. K. Kim, A. A. Kolodziejczyk, X. Zhang, V. Proserpio, B. Baying, V. Benes, S. A. Teichmann, J. C. Marioni, and M. G. Heisler. Accounting for technical noise in single-cell rna-seq experiments. *Nat Methods*, 10(11):1093–5, 2013. Brennecke, Philip Anders, Simon Kim, Jong Kyoung Kolodziejczyk, Aleksandra A Zhang, Xiuwei Proserpio, Valentina Baying, Bianka Benes, Vladimir Teichmann, Sarah A Marioni, John C Heisler, Marcus G eng Research Support, Non-U.S. Gov't Nat Methods. 2013 Nov;10(11):1093-5. doi: 10.1038/nmeth.2645. Epub 2013 Sep 22.

[8] A. Butler, P. Hoffman, P. Smibert, E. Papalexi, and R. Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol*, 36(5):411–420, 2018.

[9] M. Büttner, Z. Miao, F. A. Wolf, S. A. Teichmann, and F. J. Theis. A test metric for assessing single-cell rna-seq batch correction. *Nat Methods*, 16(1):43–49, 2019.

[10] M. B. Cole, D. Risso, A. Wagner, D. DeTomaso, J. Ngai, E. Purdom, S. Dudoit, and N. Yosef. Performance assessment and selection of normalization procedures for single-cell rna-seq. *Cell Syst*, 8(4):315–328 e8, 2019. Cole, Michael B Risso, Davide Wagner, Allon DeTomaso, David Ngai, John Purdom, Elizabeth Dudoit, Sandrine Yosef, Nir eng F31 DE025176/DE/NIDCR NIH HHS/ U01 MH105979/MH/NIMH NIH HHS/ U19 MH114830/MH/NIMH NIH HHS/ Evaluation Study Research Support, N.I.H., Extramural Research Support, Non-U.S. Gov't Cell Syst. 2019 Apr 24;8(4):315-328.e8. doi: 10.1016/j.cels.2019.03.010.

[11] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal, complex systems*, 1695(5):1–9, 2006.

[12] Vladimir Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD explorations newsletter*, 4(1):65–75, 2002.

[13] Joseph L. Fix, Evelyn; Hodge. Discriminatory analysis. nonparametric discrimination: Consistency properties. Report, USAF School of Aviation Medicine, Randolph Field, Texas., 1951 1951.

[14] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. Yang, and J. Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol*, 5(10):R80, 2004. Gentleman, Robert C Carey, Vincent J Bates, Douglas M Bolstad, Ben Dettling, Marcel Dudoit, Sandrine Ellis, Byron Gautier, Laurent Ge, Yongchao Gentry, Jeff Hornik, Kurt Hothorn, Torsten Huber, Wolfgang Iacus, Stefano Irizarry, Rafael Leisch, Friedrich Li, Cheng Maechler, Martin Rossini, Anthony J Sawitzki, Gunther Smith, Colin Smyth, Gordon Tierney, Luke Yang, Jean Y H Zhang, Jianhua eng England Genome Biol. 2004;5(10):R80. doi: 10.1186/gb-2004-5-10-r80. Epub 2004 Sep 15.

[15] Inc GitHub. Github. *URl: https://github. com/(visited on 04/23/2014)*, 2016.

[16] W. W. B. Goh, W. Wang, and L. Wong. Why batch effects matter in omics data, and how to avoid them. *Trends Biotechnol*, 35(6):498–507, 2017.

[17] D. Grün, M. J. Muraro, J. C. Boisset, K. Wiebrands, A. Lyubimova, G. Dharmadhikari, M. van den Born, J. van Es, E. Jansen, H. Clevers, E. J. P. de Koning, and A. van

Oudenaarden. De novo prediction of stem cell identity using single-cell transcriptome data. *Cell Stem Cell*, 19(2):266–277, 2016.

[18] L. Haghverdi, A. T. L. Lun, M. D. Morgan, and J. C. Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol*, 36(5):421–427, 2018.

[19] Y. Hao, S. Hao, E. Andersen-Nissen, 3rd Mauck, W. M., S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zager, P. Hoffman, M. Stoeckius, E. Papalexi, E. P. Mimitou, J. Jain, A. Srivastava, T. Stuart, L. M. Fleming, B. Yeung, A. J. Rogers, J. M. McElrath, C. A. Blish, R. Gottardo, P. Smibert, and R. Satija. Integrated analysis of multimodal single-cell data. *Cell*, 2021. Hao, Yuhan Hao, Stephanie Andersen-Nissen, Erica Mauck, William M 3rd Zheng, Shiwei Butler, Andrew Lee, Maddie J Wilk, Aaron J Darby, Charlotte Zager, Michael Hoffman, Paul Stoeckius, Marlon Papalexi, Efthymia Mimitou, Eleni P Jain, Jaison Srivastava, Avi Stuart, Tim Fleming, Lamar M Yeung, Bertrand Rogers, Angela J McElrath, Juliana M Blish, Catherine A Gottardo, Raphael Smibert, Peter Satija, Rahul eng Cell. 2021 May 27. pii: S0092-8674(21)00583-3. doi: 10.1016/j.cell.2021.04.048.

[20] G. Heimberg, R. Bhatnagar, H. El-Samad, and M. Thomson. Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. *Cell Syst*, 2(4):239–250, 2016. Heimberg, Graham Bhatnagar, Rajat El-Samad, Hana Thomson, Matt eng DP5 OD012194/OD/NIH HHS/ P50 GM081879/GM/NIGMS NIH HHS/ T32 EB009383/EB/NIBIB NIH HHS/ T32 GM067547/GM/NIGMS NIH HHS/ Research Support, N.I.H., Extramural Cell Syst. 2016 Apr 27;2(4):239-250. doi: 10.1016/j.cels.2016.04.001. Epub 2016 Apr 27.

[21] N. Henig, N. Avidan, I. Mandel, E. Staun-Ram, E. Ginzburg, T. Paperna, R. Y. Pinter, and A. Miller. Interferon-beta induces distinct gene expression response patterns in human monocytes versus t cells. *PLoS One*, 8(4):e62366, 2013.

[22] B. Hie, B. Bryson, and B. Berger. Efficient integration of heterogeneous single-cell transcriptomes using scanorama. *Nat Biotechnol*, 37(6):685–691, 2019.

[23] T. Ilicic, J. K. Kim, A. A. Kolodziejczyk, F. O. Bagger, D. J. McCarthy, J. C. Marioni, and S. A. Teichmann. Classification of low quality cells from single-cell rna-seq data. *Genome Biol*, 17:29, 2016. Ilicic, Tomislav Kim, Jong Kyoung Kolodziejczyk, Aleksandra A Bagger, Frederik Otzen McCarthy, Davis James Marioni, John C Teichmann, Sarah A eng Biotechnology and Biological Sciences Research Council/United Kingdom Research Support, Non-U.S. Gov't England Genome Biol. 2016 Feb 17;17:29. doi: 10.1186/s13059-016-0888-1.

[24] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[25] W. E. Johnson, C. Li, and A. Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–27, 2007.

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

[27] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley and Sons, 2009.

[28] E. Kleiman, D. Salyakina, M. De Heusch, K. L. Hoek, J. M. Llanes, I. Castro, J. A. Wright, E. S. Clark, D. M. Dykxhoorn, E. Capobianco, A. Takeda, R. M. McCormack, E. R. Podack, J. C. Renauld, and W. N. Khan. Corrigendum: Distinct transcriptomic features are associated with transitional and mature b-cell populations in the mouse spleen. *Front Immunol*, 7:267, 2016. Kleiman, Eden Salyakina, Daria De Heusch, Magali Hoek, Kristen L Llanes, Joan M Castro, Iris Wright, Jacqueline A Clark, Emily S Dykxhoorn, Derek M Capobianco, Enrico Takeda, Akiko McCormack, Ryan M Podack, Eckhard R Renauld, Jean-Christophe Khan, Wasif N eng F31 AI106290/AI/NIAID NIH HHS/ Published Erratum Switzerland Front Immunol. 2016 Jul 4;7:267. doi: 10.3389/fimmu.2016.00267. eCollection 2016.

[29] I. Korsunsky, N. Millard, J. Fan, K. Slowikowski, F. Zhang, K. Wei, Y. Baglaenko, M. Brenner, P. R. Loh, and S. Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nat Methods*, 16(12):1289–1296, 2019.

[30] N. Lawlor, J. George, M. Bolisetty, R. Kursawe, L. Sun, V. Sivakamasundari, I. Kycia, P. Robson, and M. L. Stitzel. Single-cell transcriptomes identify human islet cell signatures and reveal cell-type-specific expression changes in type 2 diabetes. *Genome Res*, 27(2):208–222, 2017.

[31] J. T. Leek, R. B. Scharpf, H. C. Bravo, D. Simcha, B. Langmead, W. E. Johnson, D. Geman, K. Baggerly, and R. A. Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat Rev Genet*, 11(10):733–9, 2010.

[32] Bryan W Lewis, Jim Baglama, and Lothar Reichel. The irlba package, 2019.

[33] Y. Lin, S. Ghazanfar, K. Y. X. Wang, J. A. Gagnon-Bartsch, K. K. Lo, X. Su, Z. G. Han, J. T. Ormerod, T. P. Speed, P. Yang, and J. Y. H. Yang. scmerge leverages factor analysis, stable expression, and pseudoreplication to merge multiple single-cell rna-seq datasets. *Proc Natl Acad Sci U S A*, 116(20):9775–9784, 2019.

[34] Yang Liu, Tao Wang, Bin Zhou, and Deyou Zheng. Robust integration of multiple single-cell rna sequencing datasets using a single reference space. *Nature biotechnology*, pages 1–8, 2021.

[35] M. D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, and F. J. Theis. Benchmarking atlas-level data integration in single-cell genomics. *bioRxiv*, page 2020.05.22.111161, 2020.

[36] M. D. Luecken and F. J. Theis. Current best practices in single-cell rna-seq analysis: a tutorial. *Mol Syst Biol*, 15(6):e8746, 2019. Luecken, Malte D Theis, Fabian J eng Research Support, Non-U.S. Gov't Review England Mol Syst Biol. 2019 Jun 19;15(6):e8746. doi: 10.15252/msb.20188746.

[37] A. T. Lun, K. Bach, and J. C. Marioni. Pooling across cells to normalize single-cell rna sequencing data with many zero counts. *Genome Biol*, 17:75, 2016. Lun, Aaron T L Bach, Karsten Marioni, John C eng A17197/Cancer Research UK/United Kingdom Research Support, Non-U.S. Gov't England Genome Biol. 2016 Apr 27;17:75. doi: 10.1186/s13059-016-0947-7.

[38] James MacQueen. Some methods for classification and analysis of multivariate observations, 1967.

[39] C. Mayer, C. Hafemeister, R. C. Bandler, R. Machold, R. Batista Brito, X. Jaglin, K. Allaway, A. Butler, G. Fishell, and R. Satija. Developmental diversification of cortical inhibitory interneurons. *Nature*, 555(7697):457–462, 2018. Mayer, Christian Hafemeister, Christoph Bandler, Rachel C Machold, Robert Batista Brito, Renata Jaglin, Xavier Allaway, Kathryn Butler, Andrew Fishell, Gord Satija, Rahul eng R01 NS039007/NS/NINDS NIH HHS/ P01 NS074972/NS/NINDS NIH HHS/ R01 MH071679/MH/NIMH NIH HHS/ T32 GM007308/GM/NIGMS NIH HHS/ DP2 HG009623/HG/NHGRI NIH HHS/ F30 MH114462/MH/NIMH NIH HHS/ R01 NS081297/NS/NINDS NIH HHS/ R01 MH111529/MH/NIMH NIH HHS/ Research Support, N.I.H., Extramural Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, Non-P.H.S. England Nature. 2018 Mar 22;555(7697):457-462. doi: 10.1038/nature25999. Epub 2018 Mar 5.

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

[41] M. J. Muraro, G. Dharmadhikari, D. Grün, N. Groen, T. Dielen, E. Jansen, L. van Gurp, M. A. Engelse, F. Carlotti, E. J. de Koning, and A. van Oudenaarden. A single-cell transcriptome atlas of the human pancreas. *Cell Syst*, 3(4):385–394.e3, 2016.

[42] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[43] K. Polański, M. D. Young, Z. Miao, K. B. Meyer, S. A. Teichmann, and J. E. Park. Bbknn: fast batch alignment of single cell transcriptomes. *Bioinformatics*, 36(3):964–965, 2020.

[44] R Satija, P Hoffman, and A Butler. Seuratdata: Install and manage seurat datasets, 2019.

[45] L. Scrucca, M. Fop, T. B. Murphy, and A. E. Raftery. mclust 5: Clustering, classification and density estimation using gaussian finite mixture models. *R J*, 8(1):289–317, 2016. Scrucca, Luca Fop, Michael Murphy, T Brendan Raftery, Adrian E eng R01 HD054511/HD/NICHD NIH HHS/ R01 HD070936/HD/NICHD NIH HHS/ U54 HL127624/HL/NHLBI NIH HHS/ R J. 2016 Aug;8(1):289-317.

[46] Å Segerstolpe, A. Palasantza, P. Eliasson, E. M. Andersson, A. C. Andréasson, X. Sun, S. Picelli, A. Sabirsh, M. Clausen, M. K. Bjursell, D. M. Smith, M. Kasper, C. Ämmälä, and R. Sandberg. Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell Metab*, 24(4):593–607, 2016.

[47] C. Soneson and M. D. Robinson. Bias, robustness and scalability in single-cell differential expression analysis. *Nat Methods*, 15(4):255–261, 2018. Soneson, Charlotte Robinson, Mark D eng Research Support, Non-U.S. Gov't Nat Methods. 2018 Apr;15(4):255-261. doi: 10.1038/nmeth.4612. Epub 2018 Feb 26.

[48] O. Stegle, S. A. Teichmann, and J. C. Marioni. Computational and analytical challenges in single-cell transcriptomics. *Nat Rev Genet*, 16(3):133–45, 2015. Stegle, Oliver Teichmann, Sarah A Marioni, John C eng Review England Nat Rev Genet. 2015 Mar;16(3):133-45. doi: 10.1038/nrg3833. Epub 2015 Jan 28.

[49] Alexander Strehl. *Relationship-based clustering and cluster ensembles for high-dimensional data mining*. The University of Texas at Austin, 2002.

[50] Stephen Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering (studies in nonlinearity)*. CRC Press, 2001.

[51] T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902.e21, 2019.

[52] V. Svensson, R. Vento-Tormo, and S. A. Teichmann. Exponential scaling of single-cell rna-seq in the past decade. *Nat Protoc*, 13(4):599–604, 2018.

[53] Consortium Tabula Muris, coordination Overall, coordination Logistical, collection Organ, processing, preparation Library, sequencing, analysis Computational data, annotation Cell type, group Writing, group Supplemental text writing, and investigators Principal. Single-cell transcriptomics of 20 mouse organs creates a tabula muris. *Nature*, 562(7727):367–372, 2018. eng DP1 LM012179/LM/NLM NIH HHS/ DP1 AG053015/AG/NIA NIH HHS/ K08 DK101603/DK/NIDDK NIH HHS/ I01 RX001222/RX/RRD VA/ P30 DK116074/DK/NIDDK NIH HHS/ I01 BX002324/BX/BLRD VA/ P30 DK026743/DK/NIDDK NIH HHS/ R01 CA157877/CA/NCI NIH HHS/ Research

Support, N.I.H., Extramural Research Support, Non-U.S. Gov't Research Support, U.S. Gov't, Non-P.H.S. England Nature. 2018 Oct;562(7727):367-372. doi: 10.1038/s41586-018-0590-4. Epub 2018 Oct 3.

[54] Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control, 1985.

[55] R Core Team. Vienna: R foundation for statistical computing, 2020, 2020.

[56] H. T. N. Tran, K. S. Ang, M. Chevrier, X. Zhang, N. Y. S. Lee, M. Goh, and J. Chen. A benchmark of batch-effect correction methods for single-cell rna sequencing data. *Genome Biol*, 21(1):12, 2020.

[57] C. Trapnell. Defining cell types and states with single-cell genomics. *Genome Res*, 25(10):1491–8, 2015. Trapnell, Cole eng DP2 HD088158/HD/NICHD NIH HHS/ Research Support, Non-U.S. Gov't Review Genome Res. 2015 Oct;25(10):1491-8. doi: 10.1101/gr.190595.115.

[58] K. Van den Berge, F. Perraudeau, C. Soneson, M. I. Love, D. Risso, J. P. Vert, M. D. Robinson, S. Dudoit, and L. Clement. Observation weights unlock bulk rna-seq tools for zero inflation and single-cell applications. *Genome Biol*, 19(1):24, 2018. Van den Berge, Koen Perraudeau, Fanny Soneson, Charlotte Love, Michael I Risso, Davide Vert, Jean-Philippe Robinson, Mark D Dudoit, Sandrine Clement, Lieven eng ABS4NGS ANR-11-BINF-0001/Agence Nationale de la Recherche/International 1S 418 16N/Fonds Wetenschappelijk Onderzoek/International U01 MH105979/MH/NIMH NIH HHS/ 280032/European Research Council/International P7/06/IAP StUDyS grant/International U01 MH105979/NH/NIH HHS/ P01 CA142538/CA/NCI NIH HHS/ FK-16-107/Forschungskredit/International ERC-SMAC-290032/European Research Council/International CA142538-08/NH/NIH HHS/ Research Support, Non-U.S. Gov't England Genome Biol. 2018 Feb 26;19(1):24. doi: 10.1186/s13059-018-1406-4.

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

[60] Guido Van Rossum. Python programming language, 2007.

[61] A. Vandenbon, V. H. Dinh, N. Mikami, Y. Kitagawa, S. Teraguchi, N. Ohkura, and S. Sakaguchi. Immuno-navigator, a batch-corrected coexpression database, reveals cell type-specific gene networks in the immune system. *Proc Natl Acad Sci U S A*, 113(17):E2393–402, 2016. Vandenbon, Alexis Dinh, Viet H Mikami, Norihisa Kitagawa, Yohko Teraguchi, Shunsuke Ohkura, Naganari Sakaguchi, Shimon eng Research Support, Non-U.S. Gov't Proc Natl Acad Sci U S A. 2016 Apr 26;113(17):E2393-402. doi: 10.1073/pnas.1604351113. Epub 2016 Apr 13.

[62] Kaijun Wang, Jie Zheng, Junying Zhang, and Jiyang Dong. Estimating the number of clusters via system evolution for cluster analysis of gene expression data. *IEEE Transactions on information technology in biomedicine*, 13(5):848–853, 2009.

[63] J. D. Welch, V. Kozareva, A. Ferreira, C. Vanderburg, C. Martin, and E. Z. Macosko. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887.e17, 2019.

[64] L. Zappia, B. Phipson, and A. Oshlack. Splatter: simulation of single-cell rna sequencing data. *Genome Biol*, 18(1):174, 2017.

[65] L. Zappia, B. Phipson, and A. Oshlack. Exploring the single-cell rna-seq analysis landscape with the scrna-tools database. *PLoS Comput Biol*, 14(6):e1006245, 2018. Zappia, Luke Phipson, Belinda Oshlack, Alicia eng Research Support, Non-U.S. Gov't PLoS Comput Biol. 2018 Jun 25;14(6):e1006245. doi: 10.1371/journal.pcbi.1006245. eCollection 2018 Jun.

[66] Y. Zhang, G. Parmigiani, and W. E. Johnson. : batch effect adjustment for rna-seq count data. *NAR Genom Bioinform*, 2(3):lqaa078, 2020.

[67] Y. Zhang and F. Wang. Ssber: removing batch effect for single-cell rna sequencing data. *BMC Bioinformatics*, 22(1):249, 2021. Zhang, Yin Wang, Fei eng 61472086/National Natural Science Foundation of China England BMC Bioinformatics. 2021 May 14;22(1):249. doi: 10.1186/s12859-021-04165-w.

[68] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, M. T. Gregory, J. Shuga, L. Montesclaros, J. G. Underwood, D. A. Masquelier, S. Y. Nishimura, M. Schnall-Levin, P. W. Wyatt, C. M. Hindson, R. Bharadwaj, A. Wong, K. D. Ness, L. W. Beppu, H. J. Deeg, C. McFarland, K. R. Loeb, W. J. Valente, N. G. Ericson, E. A. Stevens, J. P. Radich, T. S. Mikkelsen, B. J. Hindson, and J. H. Bielas. Massively parallel digital transcriptional profiling of single cells. *Nat Commun*, 8:14049, 2017.