TUM Data Innovation Lab

Munich Data Science Institute (MDSI)

Technical University of Munich

&

TUM Chair of Mathematical Modeling of Biological Systems (MDSI Prof. Theis) and Institute of Computational Biology (Helmholtz Zentrum München)

Final report of project:

Creating a single-cell atlas of human blood

Authors	Florian Brandl, Yifan Chen, Tanja Hausler, Katharina Schmid, Min Tang
Mentor(s)	PhD candidate Karin Hrovatin, PhD candidate Christopher Lance, Dr. Malte Luecken, PhD candidate Lisa Sikkema
Co-Mentor	Prof. Dr. Massimo Fornasier (MDSI)
Project Lead	Dr. Ricardo Acevedo Cabra (MDSI)
Supervisor	Prof. Dr. Massimo Fornasier (MDSI)
Submission date	February 10, 2023

Abstract

Large-scale single-cell atlases have the potential to transform our understanding of human biology. Through integration of multiple datasets, it is possible to capture the variability present in the population. Due to the large availability of single-cell RNA sequencing data of human blood, it is especially interesting for atlas creation. However, the size of the datasets lead to computational and procedural problems. Here, we present different analyses, solutions and possibilities to improve the integration of large-scale single-cell RNA sequencing data.

Firstly, we evaluate GPU implementations of regular data analysis methods. In order to improve computational performance and scalability, we introduce a parallel version of SCRAN, a state of the art normalization method, making normalization of even huge atlases feasible.

Furthermore, we compare the current best practice method of performing quality control to an alternative approach, where low quality cells are identified in the joint space of integrated datasets. We find that the alternative approach has no disadvantages with regards to the integration process and can potentially lead to improved quality control.

To reduce the size of the data, we benchmark random subsampling against a cluster-dependent downsampling approach and two further optimization algorithms. To compare the four diverse methods, we evaluate them in practicability and preservation of the biological variance in the subsample, which is essential for a deeper understanding of how cells function and interact. We find that the Leiden-cluster-dependent subsampling and the algorithm Sphetcher are the most promising methods in this regard.

Lastly, we examine the possibility of preserving the biological distinctions between healthy and diseased cells after single-cell data integration. Our results show that the "mapping after integration" method effectively retains this difference, and we confirm its biological significance through multiple assessments.

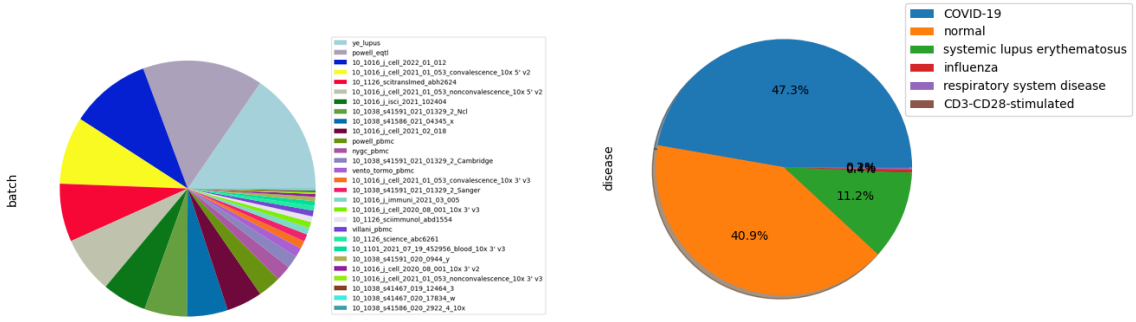
Contents

1 Introduction

RNA-sequencing is a genomic approach for detection and quantitative analysis of messenger RNA molecules in a biological sample [1]. Traditional profiling methods assess samples comprising thousands to millions of cells. While these bulk genomics have fueled much discovery and innovation in medicine, they fail to enable the direct analysis of the fundamental unit of biology - the cell [1]. Single-cell RNA-sequencing in contrast, assesses the genomics on a cellular level. It can therefore reveal complex and rare cell populations, uncover regulatory relationships between genes, and track the trajectories of distinct cell lineages in development [2].
The key elements of an experimental single-cell RNA-sequencing workflow are described by [3]. As a first step, the single-cell data from a biological sample is generated by single-cell dissociation, single-cell isolation, library construction and sequencing. The resulting raw data is processed to obtain matrices of molecular counts or, alternatively, read counts. The next step is to perform quality control, to ensure that all cellular barcode data correspond to viable cell, and normalization, which corrects relative gene expression abundances between cells. This is followed by data correction, which targets further covariates to remove unwanted technical variability. The last step is to reduce the size of the dataset by keeping only genes that are informative of the variability in the data, which is followed by dimensionality reduction and visualization. To extract biological insights and describe the underlying biological system, downstream analysis is performed. Thereby, we distinguish between cell-level approaches, describing clusters and trajectories, and gene-level approaches [3].
Single-cell omics datasets often contain many samples, that are generated across and under various conditions [4], leading to complex batch effects. Batch effects are technical variations in data, generated from observations in distinct batches. Not only technical settings such as sequencing depth, experimental laboratories, and sample composition can cause these effects, but also biological factors such as tissues and time points can be considered as batch effects. A single-cell data integration method aims to combine various datasets or samples into a joint space to create a consistent version of the data for downstream analysis. Ideally, after integration the joint version should be able to cluster cells with similar biological information together and at the same time, remove various batch effects. In the scope of this project, we aim to, on the one hand side, improve the computational performance of a state of the art single-cell RNA normalization method and on the other hand, study the impact of quality control on integration. Furthermore, we compare different subsampling approaches to decrease the sample size but preserve the biological variance from the original data, as well as examine what cells to include in the reference atlas, and what cells to map onto the reference atlas after integration to preserve biological diversity.
In this report, we present different analyses, solutions and possibilities to improve the integration of large-scale single-cell dataset integration. While there already have been integrated atlases for some organs[5, 6], the blood remains one of many organs that do not have an adequate atlas released yet. The creation of such an atlas would be of great importance to the research community as a whole, not only because of the importance of blood to the immune system, but also because of the great accessibility of single-cell RNA data on blood compared to other organs.

1.1 Data

The data we used throughout this project is Peripheral Blood Mononuclear Cells (PBMC) data, which comes from 28 different batches. In total, it contains 8195299 cells and 34001 genes and involves various health conditions. We have detailed information on cell type, disease, and age for each cell in our PBMC dataset. Additionally, information on the batch in which each cell was collected is provided. Figure 1 displays the distribution of cells per batch and per disease in the PBMC data.



(a) Proportion of cells per batch

(b) Proportion of cells per disease

Figure 1: Proportion of cells in the PBMC data

The dataset is utilized differently across various tasks:

- In order to investigate the impact of quality control on the integration performance, the raw data from original batch datasets was used, where quality control was not already applied. Only healthy data was considered, from which we took a subset into account. The data was later normalized (with a target sum of one million cells, known as CPM normalization) and $\log(x+1)$-transformed for integration.
- Apart from quality control, the other tasks were performed on the aggregated dataset, which was already normalized (with a target sum of one million cells, known as CPM normalization) and $\log(x+1)$-transformed.
- For the improvement of computational performance, the focus lays on the methods not the data, hence a subset of approximately 350000 cells was used.

If not mentioned otherwise all methods were run on Python using the packages `scanpy` for functions and `anndata` for the annotated single-cell data.

2 Improving computational performance

One of the biggest problems when dealing with currently generated datasets is how to scale existing tools to the ever-increasing size of the data. With single-cell RNA dataset sizes growing immensely in recent years[3] as well as atlases trying to combine these to even larger data, scalability is becoming more and more critical for single-cell RNA analysis methods.
This section covers the introduction of two performance-increasing methods to the general project: First, the setup and testing of the NVIDIA RAPIDS suite[7] using GPU implementations of common single-cell RNA analysis workflows. Second and more importantly, the parallelization of one of the most widely used single-cell RNA normalization methods, SCRAN.

2.1 Introduction of RAPIDS suite

2.1.1 Approach

As single-cell RNA analysis uses a lot of common data science methods such as PCA, t-SNE, and UMAP, any improvement in either memory usage or computing time would be hugely beneficial. To achieve this, NVIDIA introduced their RAPIDS suite in 2019[7], providing GPU implementations of multiple methods used for data analysis. In addition, the implementations were tested specifically on single-cell RNA data, with the authors citing the rapid increase of dataset sizes and the new development to integrate multiple datasets as key reasons.
To test whether these methods could also be used on atlas-scale single-cell data of PBMCs, the task was to run these methods on data provided by the institute and measure the time improvement compared to regular CPU implementations. Additionally, a python environment including all required dependencies was created together with a Jupyter notebook detailing the workflow and improvements of each method. The main goal here was to make these new methods easily accessible for other project members and their downstream analyses.

Task	**CPU Time Blood Data**	**GPU Time Blood Data**	**Improvement Blood Data**	**Improvement Krasnow Data**
PCA	2min 24s	N/A	N/A	-85,6%
t-SNE	48min 33s	26.2 s	-99,1%	-99,6%
k-Means	58.6 s	724 ms	-98,8%d	-99%
Nearest Neighbors	40.5 s	21 s	-48%	-79%
UMAP	10min 47s	10.4 s	-98,4%	-99,6%
Louvain Clustering	3min 14s	2.45 s	-98,7%	-99,3%
Leiden Clustering	3min 12s	1.1 s	-99,4%	-99,5%

Table 1: Improvements of the RAPIDS suite in different data analysis tasks, compared on both an internal blood dataset as well as the public Krasnov HLCA 10x lung dataset

2.1.2 Results

Table 1 indicates the performance for seven different data analysis tasks on two datasets: A blood dataset of the Helmholtz institute containing ∼350000 cells, as well as the public

Krasnov HLCA 10x lung dataset[8] containing around 70000 cells. While the first two columns indicate the absolute time each method took on the internal dataset for CPU and GPU respectively, the latter two columns list the relative improvement between CPU and GPU for both datasets.
Except for PCA and Nearest Neighbors, all methods achieved an improvement of ∼98-99%, with t-SNE and UMAP experiencing the largest absolute time improvement. The GPU implementation of PCA did not execute on our 350000 cell large dataset due to memory issues. However, due to other team members already experiencing memory issues using the regular CPU implementation of PCA and its low absolute CPU time it was decided to not pursue this issue further.

2.2 SCRAN parallelization

While methods of capturing scRNA are improving rapidly, there is still a lot of variability occurring in the overall data collection process.[3] This can lead to differences in gene counts solely as a result of sampling effects. To estimate the number of molecules originally in the cells and therefore make any downstream-analysis task more significant, normalization has to be done.

2.2.1 SCRAN as state of the art normalization method

Normalization, just like a lot of other scRNA analysis steps, offers a wide variety of methods that can be used. One of these methods is SCRAN[9], whose approach not only computes more robust size factors but also helps to alleviate batch effects to a certain extent.[3] To estimate true size factors more effectively, SCRAN assumes at least 50% non-differentially expressed genes, requiring a first clustering of cells. It first pools the cells in each cluster over a sliding window, constructing pool-based size-factors[9] consisting of the still unknown cell-specific size-factors. This linear system is then solved to receive cluster-specific size factors per cell, which are finally aligned with the help of a chosen reference cluster. For every cluster, all genes are then compared against the genes of the reference cluster, resulting in ratios for every single gene occurring in both clusters. The median of these ratios is then taken to get one value with which the cluster is scaled one final time.

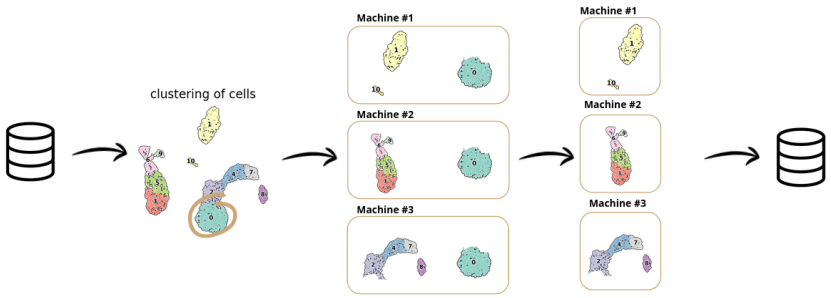
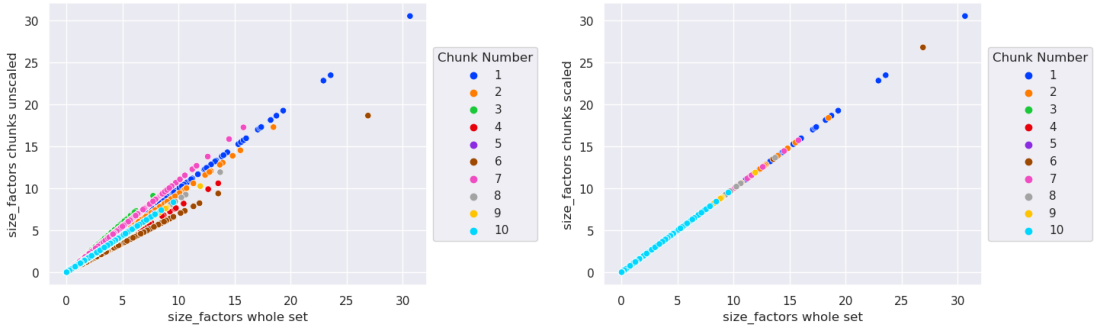


Figure 2: Proposed parallel version of SCRAN

This approach, however, does not scale well for larger datasets and results in a lot of current studies rather using a more naive total counts normalization, in which all counts are just divided by the total number of counts per cell.

2.2.2 Assumed method for paralellization

To combat this issue, we proposed and tested a new parallelized version of SCRAN, visualized in figure 2. Because SCRAN normalizes cells first in each cluster separately before scaling it once against the reference cluster, each cluster only needs the additional reference cluster to be correctly normalized. Leveraging this information, it should be possible to split all clusters into smaller chunks, before adding the same reference cluster to each chunk. These chunks can then be normalized on smaller partitions in parallel, before finally being merged back into one normalized file.



(a) (b)

Figure 3: Size factors computed by chunks both before and after scaling the chunks by the scaling factor between each reference cluster

First using the functionality provided by SCRAN of this approach however experienced some issues. While there was an exact linear correlation between the size factors of each chunk with the size factors of the whole file, the correlation between the merged size factors was only 0.93. On closer inspection, it was found that each chunk was scaled differently3(a). One potential reason for this could be a faulty indexing of the SCRAN package, resulting in the SCRAN not always choosing our proposed reference cluster but a random cluster as the reference cluster instead.
With this information in mind, a second approach was proposed. As the normalization inside each cluster is deterministic, all size factors of each chunk only differed in size by a single factor. By choosing one reference cluster and dividing the size factors of another, we get a scaling factor by which the chunks differ from each other. To finally normalize all chunks, we, therefore, scale the whole chunk by this scaling factor obtained from each reference cluster to achieve chunks with the same scaling.

2.2.3 Results

To evaluate the effect this approach would have on normalization, we evaluated the size factors of three different approaches, all done on a smaller subset mentioned in section 2.1 containing 350k cells. In the first approach, which we used as the baseline, the whole dataset was to be normalized with a maximum cluster size of 5000 and a minimum cluster size of 100. In the second approach, the data was to be first split into 10 chunks, adding a reference cluster to each one. These chunks were then normalized in parallel, before being merged back once more and scaled based on the scaling factor between each of the

	SCRAN_chunks - SCRAN_whole	SCRAN_whole - total_counts	SCRAN_chunks - total_counts
HVG Overlap	0.966	0.952	0.936
ARI - Leiden	0.625	0.068	0.066
ARI - Kmeans	0.507	0.023	0.016

Table 2: Comparison of HVG overlap as well as ARI-score for leiden and kmeans clustering between our proposed SCRAN by chunks, SCRAN on the whole dataset(baseline) and normalization by total counts

10 reference clusters. Lastly, the dataset was normalized using the total counts method described above. The results can be seen in figure 3 (b). The size factors of our proposed parallel approach and the baseline normalization of the whole dataset correlated with a factor of 1, while both had a correlation of 0.89 against the total counts.
To further evaluate the quality of the size factors obtained by our parallelized SCRAN approach, we looked at two different downstream analysis steps typically done with single cell RNA data. The first is the computation of highly variable genes(HVG), indicating the genes that explain the most variability of the data[3]. With the second, we once more clustered the data by both K-means and Leiden clustering and compared their similarity using the Adjusted Rand Index[10]. Both results can be seen in table 2, with a closer inspection of the highly variable genes in figure 4. Our parallel approach had 0.966 overlap of highly variable genes with the baseline, while the total counts normalized method only achieved an overlap of 0.952. Similar trends were observed with the ARI score, with our approach achieving an ARI score of 0.625 and 0.507 for Leiden and K-means clustering respectively, while the total counts normalization only achieved 0.068 and 0.023.

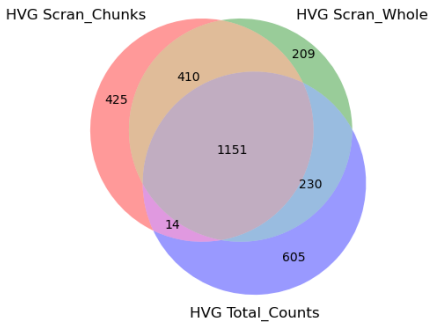


Figure 4: HVG comparison

2.3 Conclusion and Outlook

In this chapter, we showed the implementation of GPU implementations and a parallelized scran approach to improve the performance and scalability of single cell RNA analysis workflows. While the shown solutions already yield great improvements, there are still several aspects to further be addressed. One first potential improvement would be the ability of SCRAN to reliably specify a single Reference Cluster. As we showed in this chapter that the normalization per cluster is deterministic as shown by correlation between the different reference clusters, the usage of the same reference cluster for normalization between clusters as originally proposed should work in theory, making the current scaling between the reference clusters obsolete. In addition, a thorough analysis and benchmarking between current normalization methods could be done in the future, enabling a proper evaluation of the impact of our findings.

3 Quality control

Quality control (QC) is the process of ensuring that all profiled droplets used in downstream analysis correspond to viable cells. When performing droplet QC, the droplets are examined to detect low quality cells by studying the three QC metrics (total counts, number of expressed genes and fraction of mitochondrial genes) [4]. The standard practice in QC is to filter out low quality cells by setting appropriate thresholds on the QC metrics [11]. In addition, QC also includes filtering out genes which are not informative of the cellular heterogeneity as they are not expressed in more than a few cells. Lastly, QC comprises special computational strategies for ambient RNA removal [12, 13, 14], empty droplet removal [15] and doublet detection [16, 17, 18].
To ensure a sufficient data quality within single-cell atlases QC plays an important role in the workflow of creating the atlases. In the state of the art approach, QC is performed as a preprocessing step at the beginning of the workflow [5]. By applying different thresholding methods to all samples of each dataset individually, outliers are filtered out. In a later step, the datasets are integrated into an atlas.
An alternative approach to identify low quality cells is to study the data on the joint space after data integration with the same metrics as described above. This method could lead to time savings because the QC step has to be performed only once instead of repeating it for every sample in each dataset respectively. Furthermore, we hypothesize that the clustering approach could lead to an improved detection of low quality cells.

3.1 Goals of the study

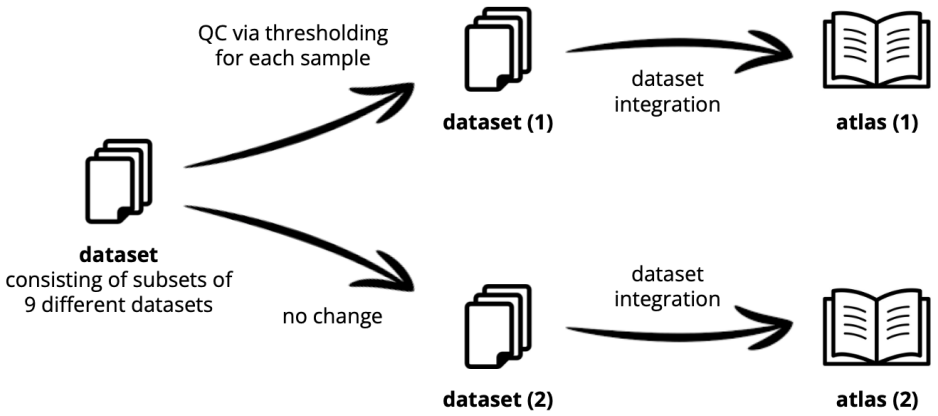
In this study, we evaluate how low quality cells affect the data integration and compare the two QC approaches described above. The main goal is to find out whether the alternative QC approach leads to better results than the current best practice of performing QC on the individual datasets before integration via thresholding. To break down this problem, we defined two guiding questions that examine different aspects of a successful atlas creation:

1. Which approach leads to better integration results?
2. Which approach is able to better identify low-quality cells?

In this study we will focus on the effect of damaged or dying cells.

3.2 Methods

To approach the problem two atlas versions are created from the same subset of datasets as shown in Figure 5. Atlas (1) is created by first performing QC via thresholding on the individual datasets and then executing the integration of the filtered datasets. For atlas (2) no QC is performed before integration. Instead, the unfiltered datasets are integrated into an atlas. At a later stage, a QC analysis via clustering is performed on atlas (2).
To answer the first guiding question, we compare atlas (1) and atlas (2) with regards to their integration performance. To assess the second guiding question, the low quality cells identified in the workflow of creating atlas (1) and analysing atlas (2) are evaluated.



(a)

Figure 5: Process of atlas creation for the analysis.

3.2.1 Data selection and sub-sampling

The first step was to create a subset consisting of several datasets which serves as a basis for the creation of the different atlases mentioned above. All datasets considered in the following analysis need to be unfiltered in order not to distort the results. As QC was already performed on some of the datasets available for analysis, we had to select the datasets which were not filtered yet. Therefore, for each dataset we created a violin plot for every sample showing the total counts per cell, the number of genes per cell and the fraction of mitochondrial genes per sample. By manually analysing these plots we excluded datasets with clearly visible cut-offs, which indicates that QC has already been performed.
From every unfiltered dataset a few samples were chosen. To ensure that these samples are representative of their respective dataset, we choose samples whose number of cells is close to the median number of cells in the dataset. Furthermore, we manually analysed the violin plots of all samples to exclude samples with strongly deviating distributions with regards to total counts, number of genes per count and fraction of mitochondrial cells to remove failed samples. The final subset consists of approximately 240000 cells and 93 samples, with 3-27 samples per dataset.

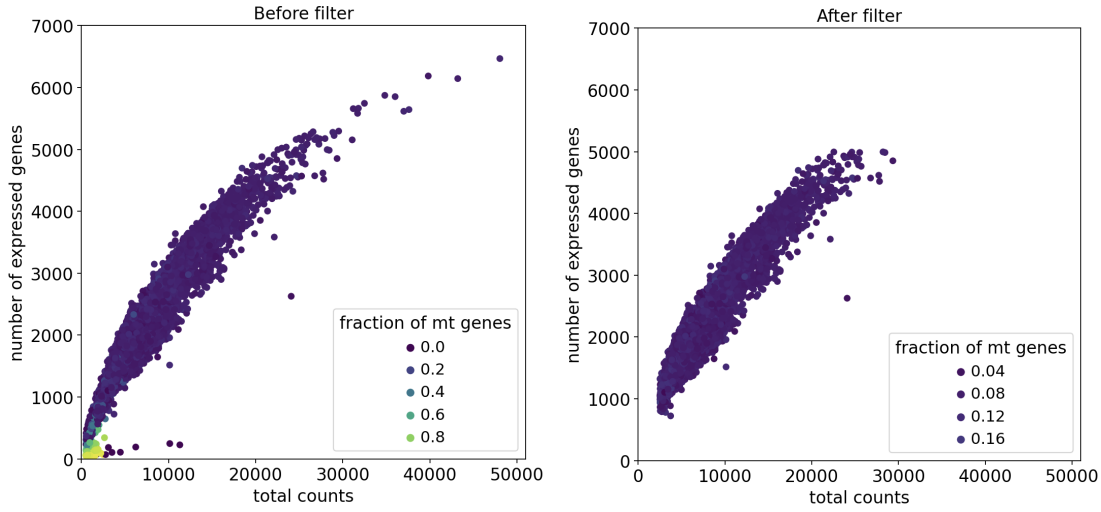
3.2.2 Quality control before integration

On each of the samples selected in the previous section, we performed QC by analysing the total number of counts per cell, the number of expressed genes per cell, and the fraction of mitochondrial genes per cell. In particular, cells with low total counts and number of expresses genes and at the same time a high fraction of mitochondrial genes are most likely low quality cells. For one of the samples, we decided to consider cells as outliers, for which one of the following conditions is satisfied.

- The fraction of mitochondrial genes is above 0.2.
- The total number of counts is below 2500 or above 30000.

- The number of expressed genes is below 400.

Figure 6 shows the scatter plots of the number of expressed genes against the total number of counts, coloured by the fraction of mitochondrial genes, before and after filtering the outliers on the exemplary sample.



(a) (b)

Figure 6: Scatter plot of number of expressed genes against the total number of counts, coloured by the fraction of mitochondrial genes (a) before and (b) after filtering out the outliers.

After performing quality control, we created two datasets, where one of them contains the unfiltered samples with a total number of 235201 cells, whereas the other one only contains viable cells with a total number of 216408 observations. In particular, 18793 cells were identified as outlier from which approximately 13300 cells are low quality cells.

3.2.3 Integration

For each of the two datasets from the previous section, we used the single-cell variational inference (scVI) framework [19] to integrate cells coming from different samples into a joint space. As the method uses stochastic optimization and deep neural networks, we ran the procedure six times for each of the datasets to account for random initialization.

3.2.4 Metrics evaluating integration performance

To evaluate which of the two atlases performs better, we focused on two categories of metrics, which were performed on each of the twelve atlases. With accordance to [4], the first category consists of the metrics PC Regression, Average Silhouette Width (ASW), Graph Connectivity, and Graph Local Inverse Simpson's Index Graph on batch mixing (iLISI). These evaluate the batch effect removal. The second group consists of the metrics Normalised Mutual Information (NMI), ASW, Isolated Label F1, Isolated Label Silhouette, and Graph Local Inverse Simpson's Index Graph on cell type separation (cLISI). These metrics measure how well the biological variation is conserved. All metrics scale

between 0 and 1, where 1 is the optimal score. Based on the mean metric value in both metric categories for each dataset, we can conduct a statistical test to investigate if there are any significant differences between the two datasets.

Two sided t-test. For both metric categories, we considered the mean value of the metrics for each of the six integration runs of both datasets. As integration was performed on both datasets separately,we can assume that the integration results for the two datasets to be independent. In order to perform a two sided t-test for the null hypothesis that the two independent atlases have identical metric means for both metric categories, we further need the assumptions that the data (mean value across metrics in a metric category) for both atlases have equal variances and that the data follows a Normal distribution. We assumed the later to hold and performed a Bartlett's test for both metric categories to check for equal variances (null hypothesis).

3.2.5 Quality control on atlas based on clustering

From six of the atlases, where all cells were included, we randomly selected one to identify regions of low quality cells. For this, we computed a Leiden clustering with resolution 0.5, which can be seen in Figure 7a and studied the total counts, the number of expressed genes and the fraction of mitochondrial genes on the atlas level, which are shown in Figure 7b - 7d.
Based on Figure 7, we can observe that cluster 10, 13, and 16 are very likely clusters containing low quality cells as cells in these clusters tend to have low total number of counts, low number of expressed genes and at the same time, high fraction of mitochondrial genes. By observing Figure 7, we can see that there are potential low quality cells at the borders of cluster 0, 1, 2, 3, 4, 5, 14, and 15. For each of these clusters, we recomputed the neighbourhood graph individually and performed the Leiden clustering algorithm with resolution 0.5 and 1 to identify subclusters, which contain low quality cells in a similar way as we did to identify cluster 10, 13, and 16.

3.3 Results

Data integration is not worsened by low quality cells. In Figure 8 and 9, we can observe that for both atlases similar cell types are rather clustered together while different batches are quite well distributed across the entire atlas. The values of the metrics specified in Section 3.2.4 for each atlas can be found in Table 4 and 5 in Appendix. For each of the twelve atlases, Figure 10 shows the mean value of the metric scores in both metric categories, as well as the overall mean across the six integration runs for each dataset in both metric categories.
The Bartlett's test for the batch effect removal metric category yields a p-value of 0.3272 and the Bartlett's test for the conservation of biological variation metric group yields a p-value of 0.6440. On a significance level of 5%, we cannot reject the null hypothesis. Thus, we assume equal variance. Based on this assumption, we performed a two sided t-test for the null hypothesis that the two independent atlases have identical averaged means in both metric categories. The t-test for the batch effect removal metric category yields a p-value of 0.1054 and the t-test for the conservation of biological variation metric

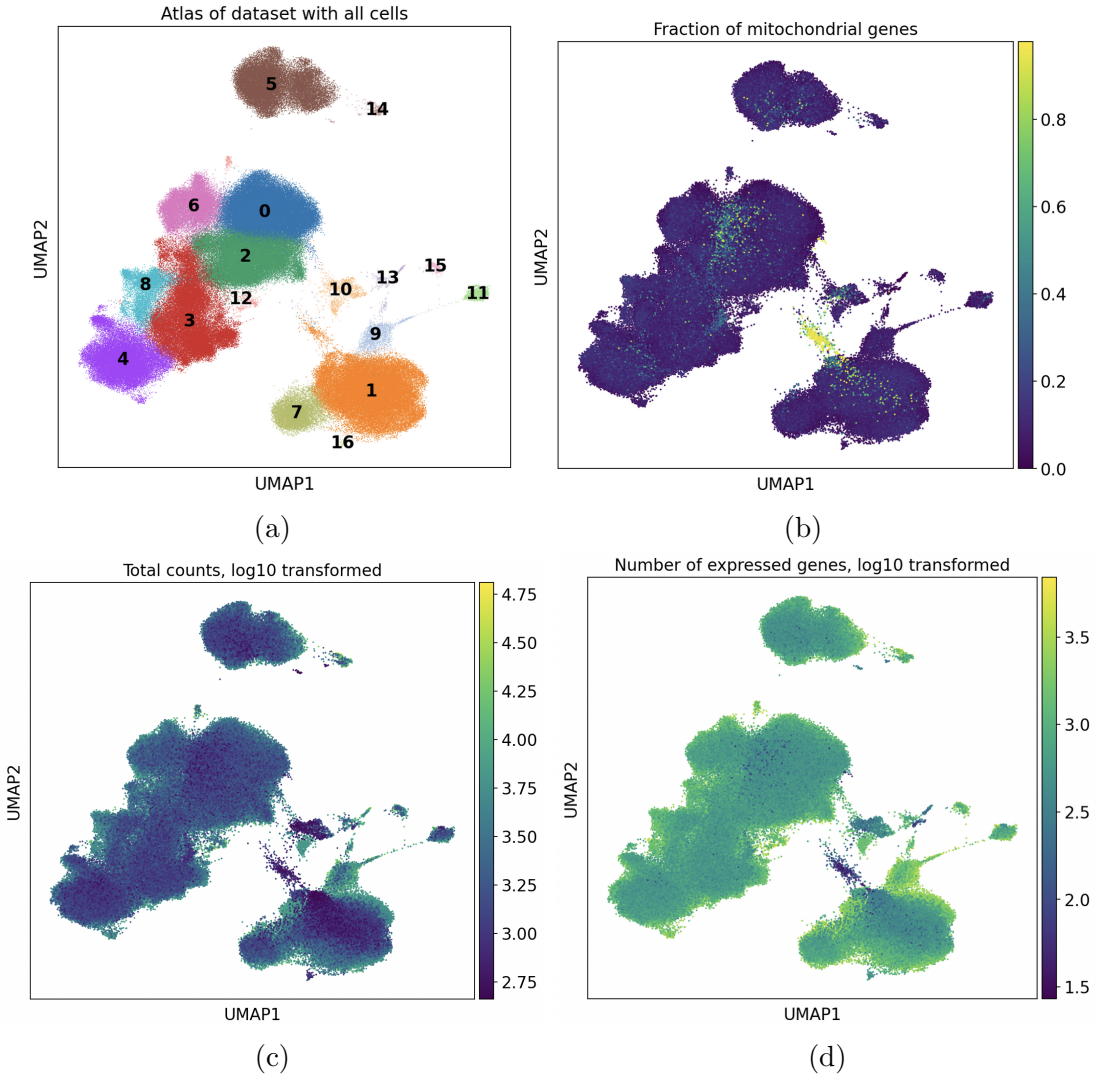
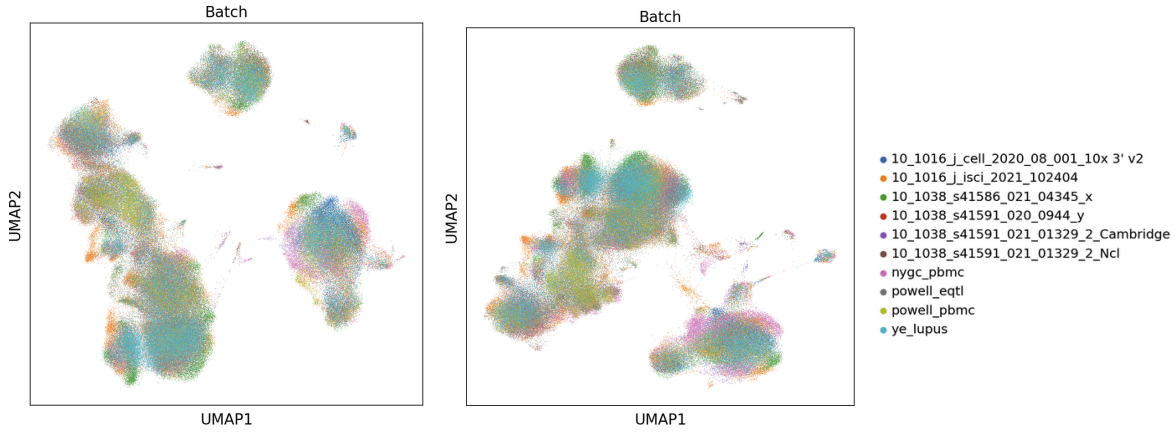


Figure 7: (a) Leiden clustering with resolution 0.5 on atlas with all cells. (b)-(d) Atlas of all cells coloured by the fraction of mitochondrial genes, log10 transformation of total counts, and number of expressed genes, respectively.

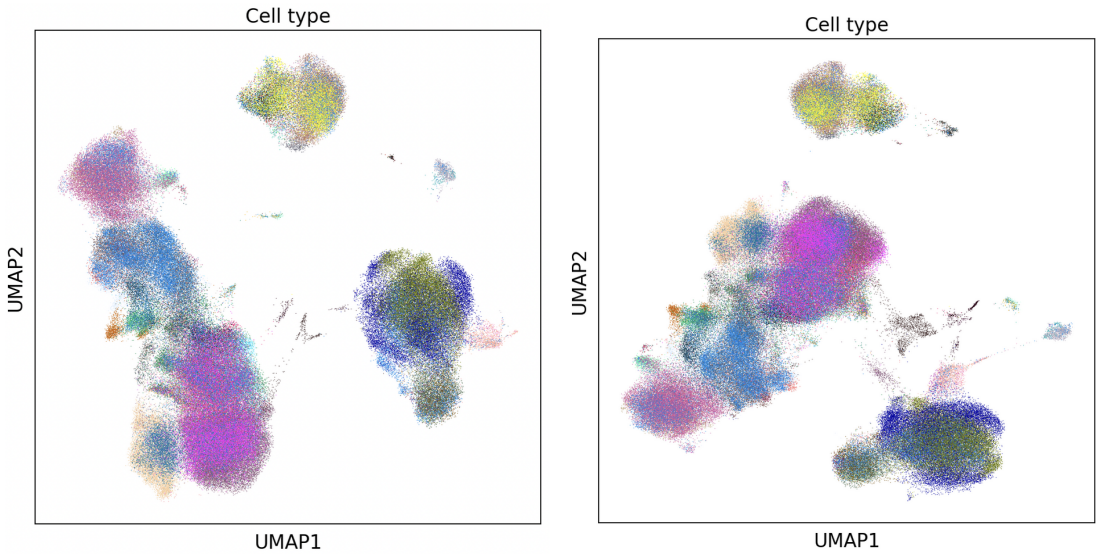
group yields a p-value of 0.2535. For both metric categories, we cannot conclude, on a significance level of 5%, that there is a difference in the averaged mean across metrics of the two atlases.

QC on the entire atlas lead to better results. Based on the clustering approach from Section 3.2.5, a total number of 7184 low quality cells were identified (see Figure 11b, which is approximately half the number of low quality cells identified before integration (see Figure 11a). Among these 7184 cells, 3352 were also identified as low quality cells before integration. Furthermore, in Figure 11 we observe that the low quality cells identified before integration are largely present in the same regions as the annotated low quality cells on atlas. In addition, the low quality cells identified on the atlas are more present at the borders of clusters or form a distinct cluster away from the other cells. There are also cells labeled as low quality cells after integration, which were not considered

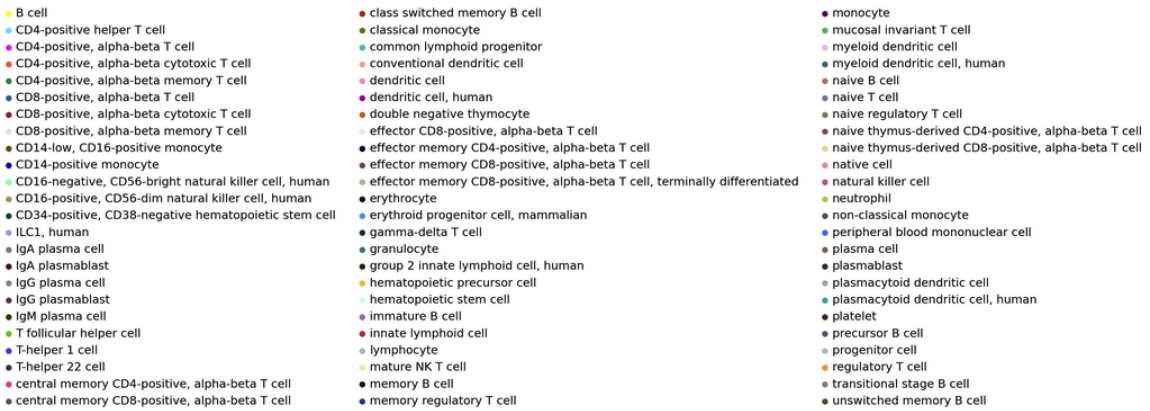


(a) (b) (c) Batch labels

Figure 8: (a): Atlas of cleaned dataset. (b): Atlas of dataset with all cells.

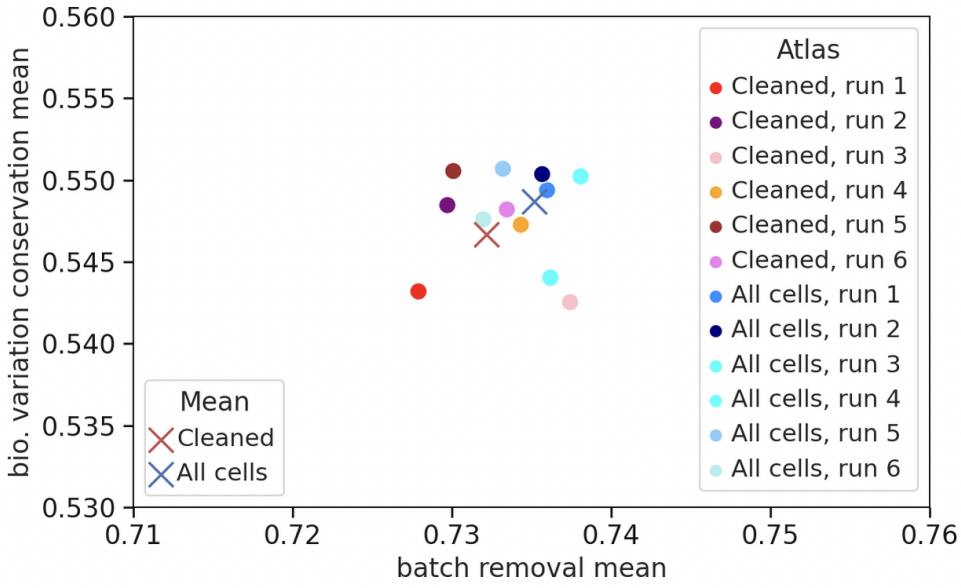


(a) (b)



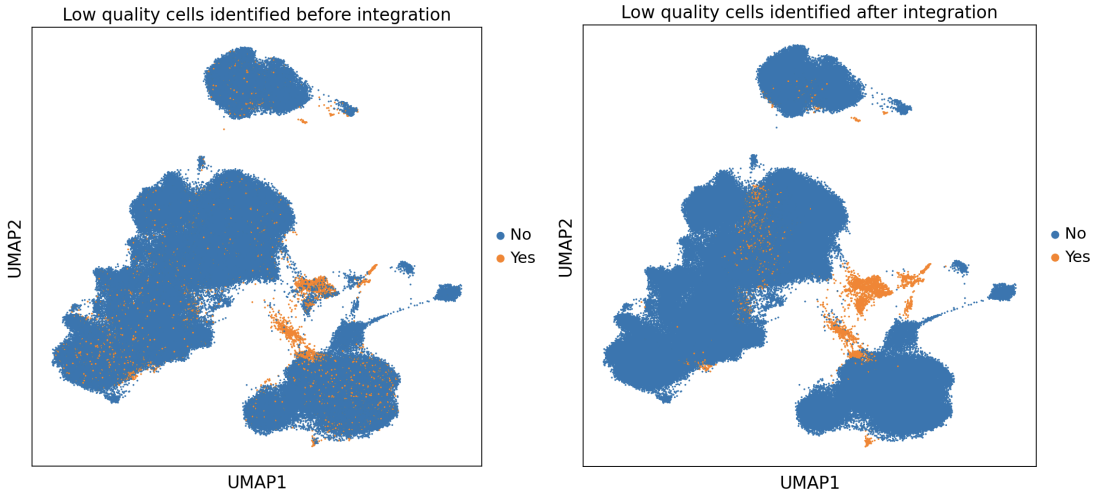
(c) Cell type labels

Figure 9: (a): Atlas of cleand dataset. (b): Atlas of dataset with all cells.



(a)

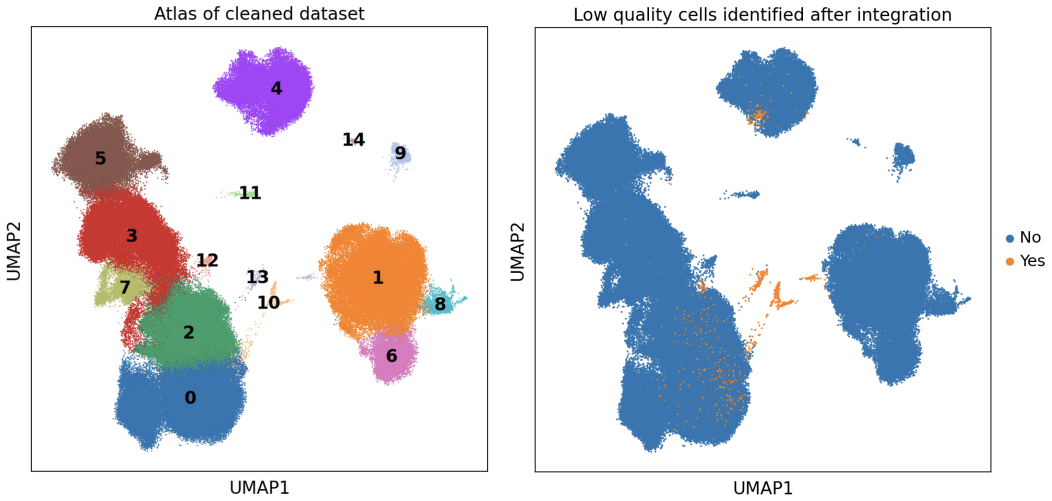
Figure 10: Values of the metrics for all integration runs.



(a) (b)

Figure 11: Low quality cells identified (a) before and (b) after integration.

as low quality cells before integration. These cells are shown in Figure 12b. In particular, we observe that these cells are mostly present in cluster 10 and 13 on the cleaned atlas in Figure 12a. These two clusters are two of the smaller clusters in the middle of the cleaned atlas, which have a larger distance to the other major clusters. This shows that performing quality control on the entire atlas after integration is potentially able to better identify the problematic low quality cells, which could hamper the downstream analysis. Particularly, performing quality control on the atlas level could also potentially help to annotate problematic low quality cells, which are difficult to be identified before integration, as the later only relies on summary statistics and not the complete transcriptome of

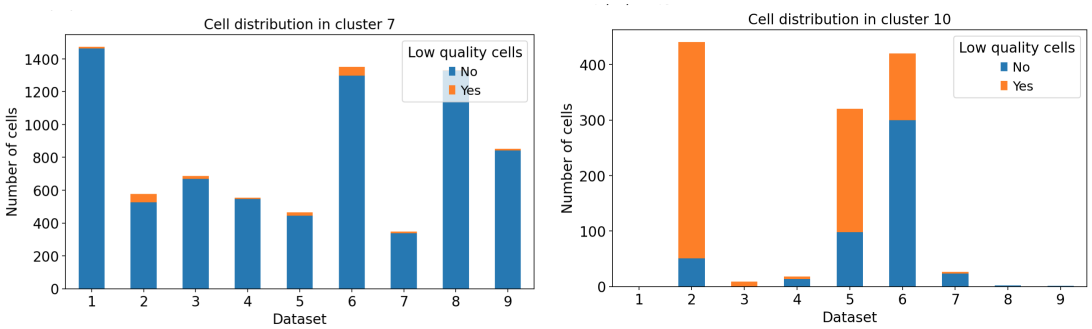


(a) (b)

Figure 12: (a) Clusters on atlas of dataset with low quality cells identified before integration removed using Leiden clustering with resolution 0.5. (b) Location of annotated low quality cells on atlas, which were not identified as low quality cells before integration.

a cell.

Low quality cells from different batches are clustered together on the joint space. We explored how the cells within the different Leiden clusters identified in Section 3.2.5 are distributed among the different datasets. This enables us to assess the intermixing of batches in the cluster. We analysed both clusters with low fractions of low quality cells, such as clusters 6, 7 and 8 as well as clusters with high fractions of low quality cells, such as clusters 10, 13 and 16.



(a) (b)

Figure 13: Cell distribution in cluster (a) 7 and (b) 10 from atlas with all cells, clustered using resolution 0.5.

Figure 13 visualizes how the cells within cluster 7 and 10 are distributed among the datasets. Thereby, the blue bars show the share of cells that were not identified as low quality cells whereas the orange bars show the share of cells that were annotated as low quality cells in Section 3.2.2. We see that the cells in the clusters originate from multiple

different datasets. The datasets within cluster 7 seem to be even more diverese than the datasets within the low quality cluster 10. However, this is likely due to the fact that with 7638 cells compared to 1237 cells, cluster 7 is significantly larger than cluster 10. Analysing the low quality clusters further, we find that previously identified low quality cells within the clusters also originate from different datasets. This allows us to conclude, that integration leads to the creation of low quality clusters across datasets. This is an important finding for the process of mapping new samples onto the atlas. It indicates that low quality cells can be mapped onto low quality clusters. This would enable automated QC during the mapping process.

3.4 Conclusion and Outlook

This study indicates that it is not necessary to perform QC before integration on each of the samples, which can be time consuming. We showed, that the alternative method of performing QC on the joint space could be superior to the state of the art approach. To verify these findings it is necessary to perform similar analyses for data from different tissue.

As this study solely considers the integration method scVI [19], it is to be investigated whether the results can also be applied to atlas creation with other integration methods. Likely, this will be the case for integration methods with similar or better integration metric scores. That is because the clustering of low quality cells is improved by high conservation of biological variance in combination with high removal of batch effects, which is indicated by a high integration score.

Moreover, QC methods not considered in this report, can be investigated to find out whether the findings of this study can be extended to QC methods in general. Doublet detection might as well profit from clustering effects and could therefore possibly be improved by the alternative QC approach. However, our hypothesis is that this is not the case for ambient gene removal as a successful integration leads to dataset-diverse clusters. If performing QC on the entire atlas via clustering turns out to be advantageous in the cases named above, a best practice for executing it needs to be created.

4 Comparative analysis of subsampling methods

Dealing with large amount of data is a fundamental issue when working with single-cell RNA-sequencing data, especially for blood. Great sample sizes restrict computational resources like memory access, therefore algorithms can take hours or even days. Some methods are not even scalable to big data. It is often necessary to create a smaller subset of representative cells from the original data to perform or accelerate computational processes. However, this will leave out some important information from the original data and can fail to represent the whole variance in the subsample. Working on a random downsampling basis is a naive approach and relatively easy to implement, but brings many drawbacks with it, such as distorting and misrepresenting this variance.

Single-cell data is highly multidimensional, the location of cells in the so-called *transcriptomic space* is determined by the gene expressions. This transcriptomic space is a "genome-wide RNA profiling" for single-cell sequencing data [21]. Whereas common, more frequent cell types are gathered densely together, rarer cell types can either be grouped in smaller sets or spread more sparsely in larger, complex clusters [20]. A random downsampling captures only partly the underlying transcriptomic structure in the subset, see figure 14. Here, frequent and common cell types show also a high occurrence in the subsample, while some rare cell types and complex transcriptional structure remain unrecognized, this leads to a distorted impression of the total geometry. Thus, the biological variance is not represented correctly in the subset.

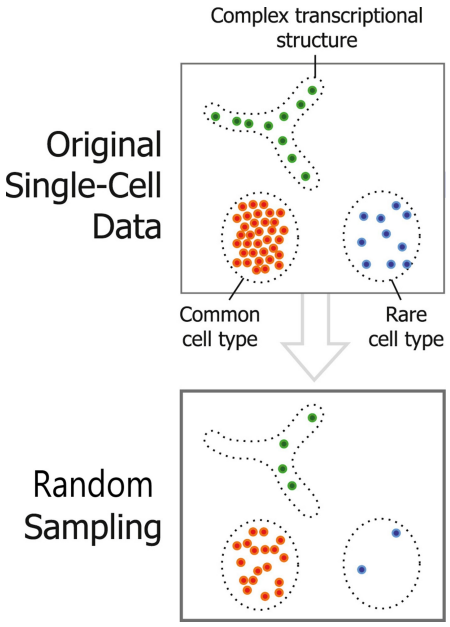


Figure 14: Weakness of random subsampling: conservation of transcriptomic landscape. Image adapted from [20]

The purpose of constructing an atlas is to represent the diversity existing in human population in a heterogeneous reference. Hereby, the total biological variance ensures a deeper understanding of how cells function and interact. Thus, this biological complexity needs to be preserved, otherwise highly relevant biological processes cannot be identified correctly. In the following, random subsampling will be compared to a cluster-dependent downsampling approach, and two further optimization methods designed to select a subsample covering the full data's topology as well as possible. A more detailed description of the different subsampling method can be found in section 4.2. To benchmark the diverse algorithms, we evaluate them in practicability and preservation of the biological variance in the subsample, see section 4.4.

4.1 Data acquisition and cleansing

Data structure and preprocessing A downsampling size of 300 000 cells from the original dataset (described in section 1.1) is chosen, which contains approximately 3.66% of the original cells. It was not possible to run all methods on the full data due to

rare cell types	common cell types
dendritic cell, pDC, granulocyte, erythroid lineage cell, hematopoietic precursor cell	B cell, T cell, monocytes, innate lymphoid cell

Table 3: Categorization of the cell types

the large sample size, so additionally it was randomly split into four subsets. Because the main goal of this work is to compare subsampling methods before integration, no quality control was performed. Different embeddings were generated using 20, 50 and 100 principal components. Based on these the later subsets are created. Various embeddings will serve as comparison since each method – except random subsampling – is based on the embedding and therefore lead to different subsamples.

Rare cell types Around 4.11% of the cell types are unlabeled. Our data includes 10 different annotated cell types, which were categorized into rare and common ones regarding the occurrence within blood data, as depicted in table 3. This classification was important for the evaluation of the methods, where we investigate to what degree rare cells are conserved and handled differently than common ones. We expect this investigation to be informative of the heterogeneity and biological variance in the subsample.

4.2 Different subsampling approaches and methods

The most intuitive way to downsample is a **random subsampling**, but it has several disadvantages which were introduced in section 4. The ultimate goal of this task is the find a method which preserves biological variance, represents the transcriptomic space in the subsample similar to the full data, and creates a heterogeneous downsample. To approach this, we implemented a **Leiden-cluster-dependent subsampling**. This cluster-dependent approach selects cells from each cluster to guarantee that also small cluster and underrepresented, rarer cells types are maintained instead of mainly keeping common, more homogeneous cells. Thus, the biological complexity is expected to be better captured than for random subsampling. However, it was found that this cluster-dependent way is not robust, and can result in an unsuitable subset for downstream analysis, as demonstrated in [20]. Furthermore, our data is quite unbalanced, meaning it contains way more common cells than rare cells. In this case it is found that clustering is often assigned incorrectly and underrepresented data observations can be clustered together even when they are not that similar and closely related [22].
More complex algorithms tailored for downsampling instead of cluster selection yield subsets, that perform more effective in downstream analyses like clustering, visualization, and integration as described in [20], [23],and [24]. These more sophisticated algorithms tackle the aforementioned problems when generating a smaller dataset by subsampling across the transcriptomic space more evenly. In the scope of this work, we will refer to the introduced methods **Sphetcher** and **scSampler** as 'smart' subsampling, because they 'smartly' aim to improve the representation of the trancriptomic space by minimizing the Hausdorff distance, which is a similarity measure, further explanations are provided in 4.3.1. This optimization results in a maximized distance and minimized similarity

between the cells in the downsample. Thus, the biological diversity should be preserved and rare cell types will likely be upsampled.

Leiden-cluster-dependent subsampling The Leiden algorithm aims to find well-connected and high quality clusters, similar cells are assumed to be clustered together [25]. Thus, we expect a rather heterogeneous subset retaining the biological variation when subsetting on cluster basis. This downsampling approach was executed on the four splitted subsets. For each subset, a kNN graph with number of neighbors per cell set to 15 was created before running the Leiden algorithm with a resolution of 20, which leads to 280–530 clusters. The number of cells within a cluster is highly varying from 1 to 32 000. All clusters with less than 500 cells are kept fully, because they are assumed to have high variations and therefore be important for the total variance. In contrast, similar cells do not contribute to the biological variance and therefore are left out. Only a fixed number of cells is kept from clusters with more than 500 cells, which is calculated as follows for each subset to get the total downsampling size of 300 000 after merging them:

$$\frac{(75\,000 - \#\text{ cells from clusters with} < 500\text{ cells})}{(\#\text{ clusters with more than } 500\text{ cells})}$$

Sphetcher [23] This algorithm creates a so-called 'sketch' taking into account the transcriptomic space by generating small, fixed size spheres that cover cells in the transcriptomic space. From each sphere one cell is selected with the goal of approximating the global geometry and get the sketch, which is expected to preserve the geometric structure, as it can be seen in figure 15. The selection of the cells from the spheres is designed as *minimax* distance design, i.e. these cells are kept which minimize the maximal Hausdorff distance to the next nearest cell [26]. Here, Pearson correlation distance is used as underlying distance metric. In opposite to the other methods, Sphetcher is not implemented in Python but in `C++`, the input has to be a csv file of the latent space.

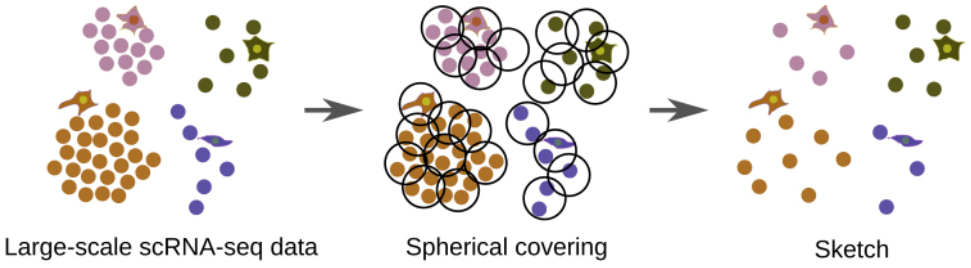


Figure 15: Visualization of Sphetcher algorithm taken from [23]

scSampler This downsampling method minimizes the Hausdorff distance with Euclidean norm as distance metric. In contrast to Sphetcher, scSampler is based on the *maximin* distance design, which allows a better separation of the cells with the downside of being computationally more expensive [24]. To enable faster computations scSampler provides a variant splitting the sample into k subsets. For this work, scSampler was performed on three variants – 20, 50, and 100 random splits – on the three embeddings to better understand the performance and detect possible differences in the outcome and conservation of biological variance.

4.3 Evaluation metrics

In finding a proper downsampling method, essential factors for the evaluation of the different methods include *computational effort*, *running time*, and *scalability*. In addition to the technical part, another focus of this comparative analysis of subsampling methods concentrates on the biological aspect. To approximate the general heterogeneity of the subsample in comparison to the original data, we look at rare cell types. The percentage of retained cells for each cell type is calculated, this provides insights into the effect of subsampling on the *transcriptional structure* and *general heterogeneity*. We expect common cell types to be more homogeneous and therefore hold less biological variety. Thus, a higher preservation of rare cell types indicates a higher conservation of the *biological variance*. A more sophisticated metric for this biological evaluation is the Hausdorff distance, as explained in the following section.

4.3.1 Hausdorff distance

In quantifying the representation of the transcriptomic space, the Hausdorff distance serves to compare how good the original data X is represented in the subsample S. This means, a subsample more similar to the original data also contains more biological variance and heterogeneity. This tool for measuring the similarity between two sets of points can be calculated by :

$$d_H(S;X) = \max_{x\in X}\{\min_{s\in S} d(x,s)\}, \qquad \text{where } d \text{ can be any distance metric [27]}$$

For the later benchmarking, we used Euclidean, Manhattan, and Cosine distance. A lower Hausdorff distance indicates that there is a high similarity between the two sets of points (in this case, full data and subsample). This means: the smaller d_H, the more comprehensively S covers the transcriptomic space of X. In contrast, a higher Hausdorff distance is evidence that there are transcriptional structures in X which are not well represented in S. Thus, the transciptomic space is not captured well in S and some of the biological variance from the original data is lost in the subsampling process.

4.4 Benchmarking the subsampling methods

To compare the four subsampling methods (random downsampling, a cluster-dependent approach and two optimization algorithms), we ran each method five times to control for possible random effects, caused by random subsampling and splitting.

4.4.1 Computational effort and time

Regarding the technical aspects, random subsampling was the quickest method, it finishes in under ten seconds, it was simple to run, and worked on 8 million cells. The Leiden-cluster-dependent subsampling is also relatively straightforward and quick, with a total running time of approximately 15 minutes, where the kNN graph and Leiden algorithm were run on GPU. It is scalable for the subset of two million cells and can be extended to the full dataset.
Due to memory issues it was neither possible to run Sphetcher on the full dataset, nor on half of the data (approximately four million cells). The four splitted subsets were

subsampled on average in 24 minutes by the algorithm – independent of the number of principal components. Finally, scSampler was on average the slowest method. The running time highly varied and depended on the number of principal components and random split, see figure 16. It took the longest time to finish between 40 minutes and 20 hours.

4.4.2 Conservation of biological variance

To approximate the general heterogeneity and estimate how well the biological variance is preserved, we calculated the percentage of each cell type kept in the subsample. Random subsampling keeps around the same fraction of each cell type (3.68%). This value can be used as a reference to evaluate the other methods. When the kept percentage is larger than this value, then cell types are upsampled. We want to archieve an upsampling in the rare cell types, because we expect them to hold a crucial part of the biological variance, a maintance promises a more heterogeneous dataset.

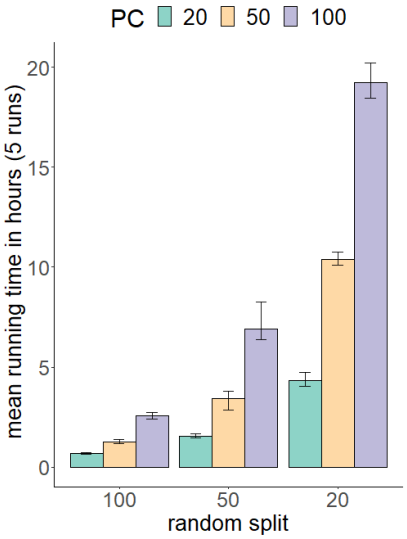


Figure 16: scSampler's running time

It is notable that for 100 principal components, scSampler is not performing as expected, as depicted in figure 17. We find that it rather upsamples common cell types and keeps a higher percentage of them, which we imagine to result in a lower biological variance, because exactly these cells are more homogeneous and therefore contain less variation. We suppose that the number of principal components is too high, because for the lower 20 and 50 principal components the results are more as expected. For further evaluations, scSampler with 100 principal components is therefore excluded.

In general, we see that the 'smart' methods preserve rare cell types with the exception of hematopoietic precursor cells, see figure 17. The Leiden-cluster-dependent subsampling is the only one handling the hematopoietic precursor cells in an upsampling way.

Taken together, the Leiden-cluster-dependent subsampling on the 20 principal components embedding seems to work best, because it upsamples all the rare cell types. This means more variation is retained on the subsample. On the contrary, Sphetcher is most conservative algorithm in upsampling, since it keeps mostly a smaller amount of cells compared to scSampler and the Leiden-cluster-dependent approach. However, it provides – except for the hematopoietic precursor cells – solid results. Gennerally, the embedding of 20 principal components appears to have a better influence on the subset regarding the heterogeneity, because the percentage of kept cells is larger and therefore the biological variety from the original data is expected to be better conserved.

Fewer common cell types are kept, especially B and T cells. We assume them to hold in general less biological variance, since the cells are more frequent and their clusters more homogeneous. The subsampling methods recognize this and 'punish' them by downsampling in favor of rare cell types. In the appendix more detailed plots of cell types preservation for each methods can be found.

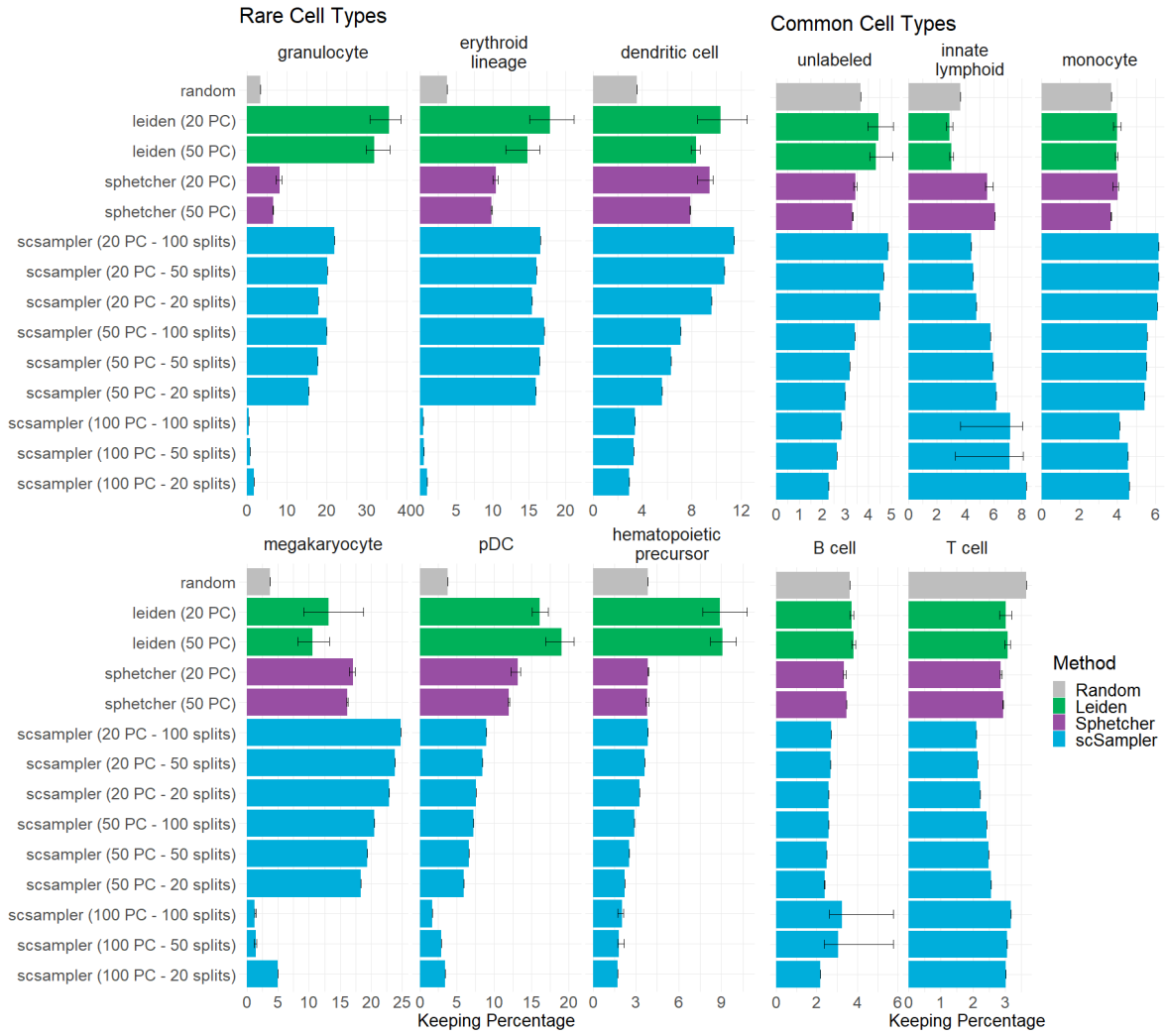


Figure 17: Overview: percentage of rare cell types kept from the different subsampling methods

4.4.3 Hausdorff distance

To quantify the preservation of the original heterogeneity in the subsample, we calculated the Hausdorff distance. A lower distance indicates that the biological variation is rather conserved. In this case, the subset created from the Sphetcher algorithm covers the transcriptomic space the most evenly, since it results consistently in the lowest Hausdorff distance for various distance metrics and principal components, as depicted in figure 18. We see a higher variability in the boxplots of the Leiden-dependent subsampling method, this may gives a hint for the previously mentioned imbalance and lacking robustness. In general, the Hausdorff distance of the Leiden-dependent subsampling and scSampler are very close, except for the cosine distance, where the scSampler has nearer results to Sphetcher. In general, we can conclude that in regard of similarity to the original data, Sphetcher has the best results and preserves the heterogeneity and biological variance best.

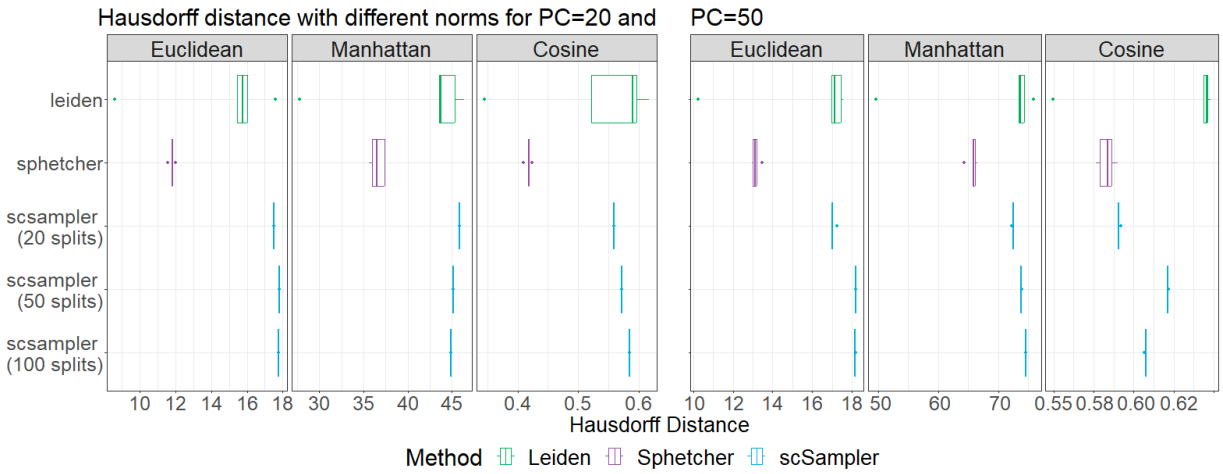


Figure 18: Comparison of Hausdorff distances for the three smart subsampling methods using Euclidean, Manhattan, and Cosine distance for 20 and 50 priniciple components

4.5 Conclusion: Effect of subsampling

For a complete comparison of the subsamples created by four different subsampling methods, which are either based on a random, cluster-dependent or optimization approaches, we evaluated them in technical and biological aspects. The fastest and simplest to run method is the random subsampling. However, in the creation of subsets for single-cell RNA-sequencing data it is crucial to ensure the preservation of biological variance and conservation of the original transcriptomic heterogeneity, which is not given for the random subsampling. For a better accounting, the cluster-dependent approach and the 'smart' methods are compared in rare cell type conservation, where the Leiden-cluster-dependent subsampling yielded the best upsampling, whereas for the Hausdorff distance, the consistently lower results from Sphetcher indicate that this methods conserves the biological complexity best. Therefore, Sphetcher and the Leiden-cluster-dependent seem to be the most promising methods for our data, they can be investigated in more detail in a further analysis.

4.5.1 Outlook

Since there are some concerns about cluster-dependent subsampling [22], it is recommended to research and perform more experiments in this regard as proposed in [20]. Moreover, for a deeper inspection of the biological variance preservation, the methods can also be executed on other datasets for a comprehensive comparison, especially to evaluate the robustness [20].
Moreover, integration of the subsampled data reveals further insights and metrics for evaluation and benchmarking. Since integration is also highly dependent on similar transcriptional structures [28], metrics evaluating batch mixing and cell type preservation – as proposed in [4] – in the integrated downsampled data give information of similarity of the trancriptomic space compared to the original dataset.

5 Preserving biological diversity after single-cell data integration

5.1 Background

Single cell atlases offer valuable insights into the transcriptomic differences between diseased and healthy cells. However, the complexity of these atlases, which can comprise millions of cells collected through various protocols in different laboratories, can introduce batch effects that obscure the true biological variation. To truly understand the biology of disease, it is necessary to create a single cell atlas that accurately reflects biological variation and eliminates technical artifacts.

The success of removing batch effects and preserving biological variation in single cell atlases relies heavily on the data integration method employed. There are several approaches for mitigating batch effects, and in this section, we will explore two of these methods: SCVI (Single-cell Variational Inference [4]), which uses a variational auto encoder, and ScArches (Single-cell Architecture Surgery [29]), which employs transfer learning to map query datasets onto a reference model. When integrating a new dataset into an existing reference model, traditional integration methods typically train all weights. ScArches, on the other hand, utilizes a different approach, only allowing the training of a small subset of weights while keeping the rest fixed [29].

In this section, we will study whether either approach can preserve true biological variances between healthy and diseased cells and *how well* each method can preserve these variances.

5.2 Dataset and Methods

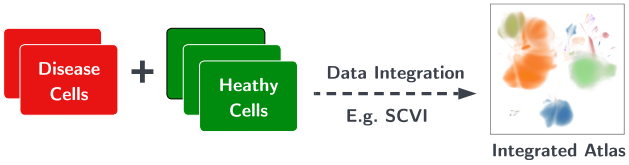
Dataset and embeddings We examine the Peripheral Blood Mononuclear Cells (PBMC) dataset of over 8 million (8195299) cells as introduced in Section 1.1. As seen in Figure 1b, 88.4% of the cells are labeled either as "COVID-19" or "normal". To simplify our discussion, we will primarily focus on these two cell types. Additionally, we were given two pre-trained lower-dimensional embeddings of the cells as shown in Figure 19. With pre-trained lower-dimensional cell embeddings, our focus will be on determining if these embeddings can preserve the biological differences between healthy and diseased cells.

Methods In particular, following techniques are utilized for visualization and clustering:

(i) **Visualization**: We used UMAP (Uniform Manifold Approximation and Projection [30]), which is widely used for visualizing single-cell data, particularly for its ability to handle large amounts of data [31]. Unlike other methods that preserve global distances, UMAP focuses on preserving the *topological structure* of the data by computing a neighborhood graph from existing embeddings. To improve computation efficiency, we utilized Rapids GPU environment [7] and observed a significant performance improvement. For more information, refer to previous discussions.

(ii) **Clustering**: We performed Leiden clustering [25] on the monocyte subsets of both embeddings. Leiden clustering is a method for detecting communities or clusters in a network, which has been adapted for use with single-cell data. It is a variation of

(Full Integration) In this embedding, both healthy and disease cells are included during the integration phase.



(Mapped after Integration) In this embedding, only healthy cells are included during the integration phase. Disease cells are mapped into the atlas after the integration phase.

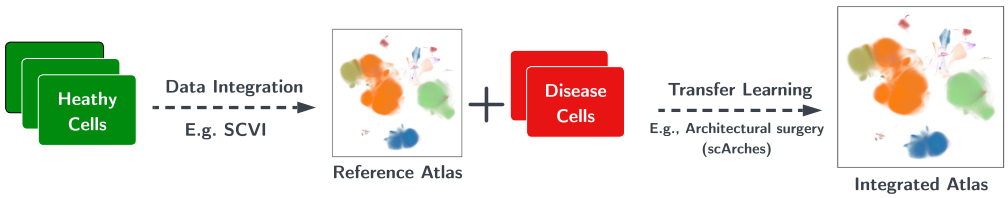


Figure 19: Techniques for obtaining lower-dimensional cell embeddings. Our discussion focuses on two pre-trained embeddings, which were derived using these methods and contain identical cell samples. Each embedding has a dimensionality of 30.

the Louvain community detection method [32] and is known for its ability to handle large-scale single-cell data efficiently. To improve performance, we again utilized the Rapids GPU environment [7].

5.3 Visualization and Subsetting

In this section, we aim to gain an initial understanding of both embeddings by visualizing them. Through this process, we hope to identify specific cell types where we observe differences in transcription between healthy and diseased cells by examining the separations in our two-dimensional representations. However, it's still important to keep in mind that the original topology of the data may be under-represented in the two-dimensional representation. To address this, in the next subsection, we will further examine the two embeddings by performing clustering using their original neighborhood graphs, with the goal of capturing more crucial information.

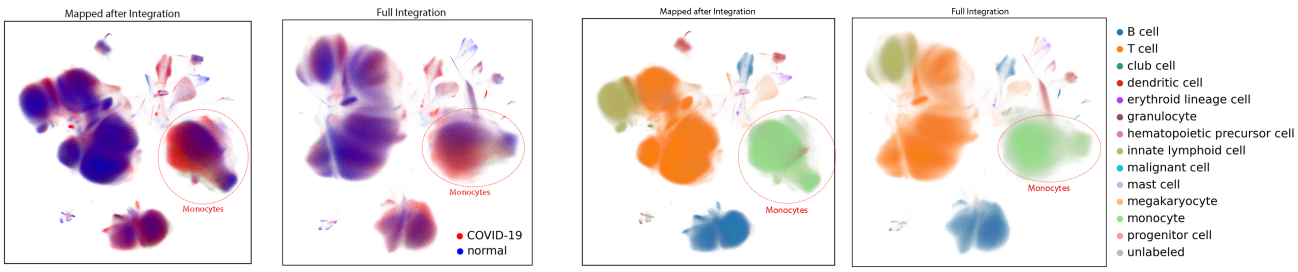
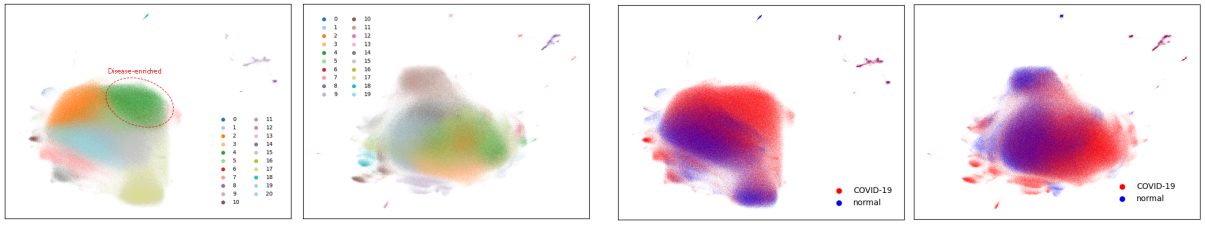


Figure 20: Visualization of the entire dataset using UMAP, colored by disease status and cell type.

Visualization results The visualization results are shown in Figure 20. We find that both embeddings separate disease and healthy cells mainly within monocytes. Therefore, we focus on the subset of 1365262 monocytes in the following analysis. While it may be desirable to further subset the data into specific monocyte sub-types to observe a more significant separation between cells, some datasets lack the necessary annotations. To have an adequate number of cells, focusing on the monocyte subset is considered as a suitable solution.

5.4 Identifying disease-specific clusters

As a reminder, our goal is to find out if the lower-dimensional embeddings can preserve the differences between healthy and diseased cells from a biological perspective. To accomplish this, we will use clustering algorithms to uncover cell clusters that are specific to the disease. Our first step is to identify these disease-specific cell clusters and separate them from the normal cell clusters. In this section, we will focus on whether we can successfully recover these disease-specific cell clusters. The next section will look into whether the cell clusters we've identified are biologically significant.



(a) Cells colored by Leiden cluster (b) Cells colored by disease

Figure 21: Identifying disease-specific clusters. Left: Mapped after Integration Right: Full Integration

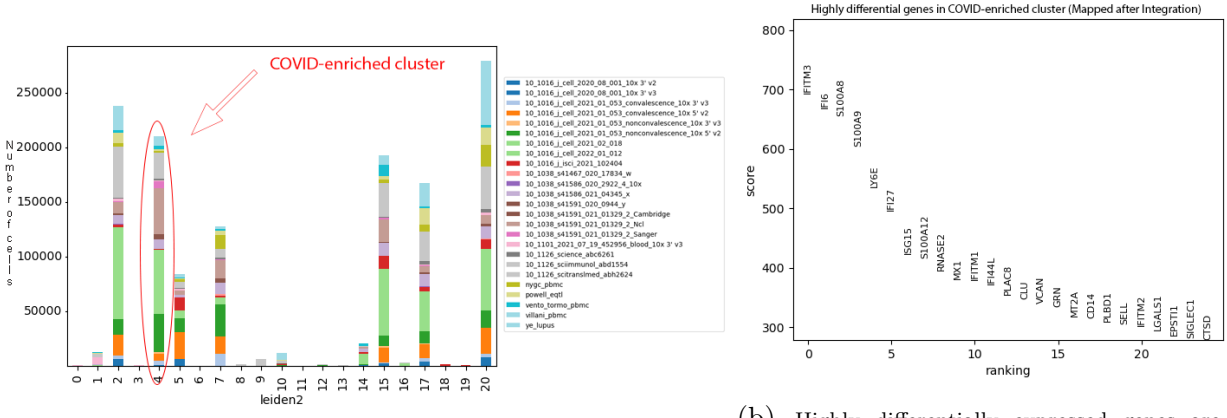
Identifying disease-specific clusters For both embeddings, we conducted Leiden algorithms using different resolutions selected uniformly and independently from $[\frac{1}{2}, 1]$, with the goal of obtaining resolution-independent results. Our objective was to determine if disease-specific clusters could be identified. At a resolution of 0.6, we identified a single disease-enriched cluster for the mapped after integration embedding. This cluster consisted of 203,496 cells, and 96.8% of those cells were labeled as "COVID-19". This specific cluster is circled in Figure 21a. Although other smaller clusters with similarly high percentages of disease cells were also identified, they were not considered in our analysis as they contained cells from only a very limited number of batches (< 3) and could be attributed to batch effects. In the case of the full integration embedding, despite trying different resolutions, we were unable to identify a similar disease-specific cluster. Clusters with similarly high percentages of disease cells tended to have fewer cells and those cells were from very few batches. Therefore, we were unable to use clustering algorithms to recover disease-specific clusters from the full integration embedding and determine the ability of that method to capture the biological differences between disease and healthy cells.

5.5 Validating the biological relevance of identified disease-specific clusters

We have discovered a cluster specific to COVID-19 in the mapped after integration embedding. Our next step is to confirm if this cluster is actually due to biological differences across transcriptomes. To accomplish this, we will conduct two forms of validation

(i) It's important to note that the separation between cells can also occur simply because they originate from different batches. To rule out this possibility, we will verify that cells in the cluster are distributed evenly across all available batches, as depicted in Figure 22a. This will ensure that the cluster is not simply driven by batch effects.

(ii) To verify the biological significance of the cluster, we conduct statistical tests and compare the highly differentially expressed genes in the disease-specific cluster against those in other clusters, as shown in Figure 22b. The results show that the highly differentially expressed genes, such as IFITM3 and IFI6, are indeed associated with inflammation and closely related to COVID-19. These findings are supported by previous studies [33, 34, 35].



(a) Cells stem from different batches

(b) Highly differentially expressed genes are disease-related

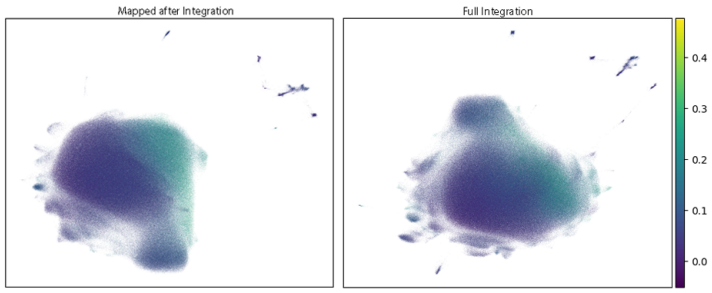
Figure 22: Validation of the identified disease-specific cluster

In light of the results from our validation, we are confident that the identified disease-specific cluster is not due to technical artifacts, but rather has a biological basis that enables us to determine cell identity. In conclusion, our results demonstrate that the mapping after integration method effectively captures and retains biological transcriptomic differences across cells.

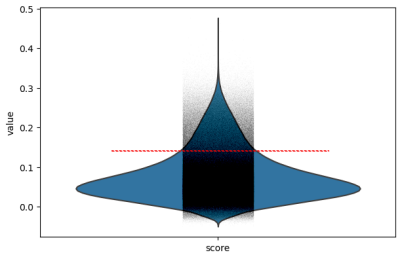
5.6 Cluster-independent evaluation

So far, we have verified that disease-specific clusters can be recovered from the mapped after integration embedding, but we have not been able to draw any conclusions about the full integration embedding. As clustering can be influenced by other factors, including resolution, we aim to compare the embeddings independent of clustering in this subsection.

Scoring interferon genes Interferon genes play a crucial role in the intensity of the immune response to specific pathogens and the severity of diseases [36]. Therefore, we use this fact and consider the set of interferon genes as the ground truth for the disease. Next, we score the set of interferon genes for each cell, as shown in Figure 23a. The score is calculated as the difference between the average expression of interferon genes and the average expression of a randomly selected reference set of genes [37]. Our aim is to assess how well cells with high interferon scores are separated from cells with low interferon scores.



(a) UMAP of cell embeddings colored by interferon genes scores



(b) Violin plot showing a potential cutoff between high- and low-interferon scores

Figure 23: Separating cells according to interferon genes scores

It's worth mentioning that the scoring results are consistent with our previous findings: cells in the identified disease-specific cluster (as shown in Figure 21a) tend to have higher interferon gene scores. To quantitatively evaluate this, we can set a natural cut-off between high- and low-interferon scores and then use the Local Inverse Simpson's Index [38] to measure both the separation between high- and low-interferon score cells and the integration of different batches simultaneously. Unfortunately, due to the large number of cells (over a million), the calculation could not be completed within a reasonable timeframe.

5.7 Discussion

In this section, we evaluated the capability of "full integration" and "mapping after integration" methods in capturing the biological differences between healthy and disease cells through lower-dimensional embeddings. Our results showed that disease-specific clusters were successfully recovered from the "mapped after integration" embedding and were confirmed to have a biological significance. In conclusion, our findings suggest that the mapping after integration approach is highly effective in preserving and highlighting biological transcriptomic diversity among cells. Future efforts could include calculating the Local inverse Simpson's index (LISI) as a quantitative measure of the separation between cells with high and low interferon scores.

6 Conclusion

During the TUM Data Innovation Lab in cooperation with Helmholtz Munich, we worked on several aspects to improve the atlas creation of a single-cell blood atlas. Our results can be applied to other large-scale single-cell atlases or serve as basis for future investigations. Firstly, we analyzed GPU implementations of state of the art data analysis methods with regards to their computational performance. Furthermore, we developed a parallel version of SCRAN, which enables normalization of large, atlas-sized single cell RNA datasets.
Our subsequent investigation of QC in the workflow of atlas creation indicates that performing QC on the joint space after dataset integration could be advantageous to the state of the art process. While it does not harm the integration performance, our results suggest that it could lead to an improved detection of low quality cells. Based on these findings similar analyses of different tissues and for further QC methods need to be performed.
For a reduction of the large sample size, we benchmark random subsampling against a cluster-dependent downsampling approach, and two further optimization algorithms. In the creation of subsets for single-cell RNA sequencing data it is crucial to ensure the preservation of biological variance and conservation of the original transcriptomic heterogeneity. For this purpose, the subsampling methods were compared in Hausdorff distance and percentage of rare cell types kept. It was found that all methods except random subsampling upsample rare cell types, the Leiden-cluster-dependent subsampling performs best in this regard. The lowest Hausdorff distance and therefore the most heterogeneous subsample is yield by the Sphetcher algorithm. Taken together, each method has its strengths and weaknesses, but for our data the Leiden cluster depending subsampling and Sphetcher on a 20 principle components latent space are the most promising ones.
Finally, we evaluated the preservation of transcriptomic heterogeneity between healthy and diseased cells in lower-dimensional embeddings generated by widely used data integration methods like scVI and scArches. Our findings show that the “mapping after integration” embedding produced by scArches has the ability to maintain this biological diversity and we also confirmed its biological significance.

Bibliography

[1] Ashraful Haque et al. "A practical guide to single-cell RNA-sequencing for biomedical research and clinical applications". In: *Genome Medicine* 9.1 (Aug. 18, 2017), p. 75. ISSN: 1756-994X. DOI: 10.1186/s13073-017-0467-4. URL: https://doi.org/10.1186/s13073-017-0467-4.

[2] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. "Single-cell RNA sequencing technologies and bioinformaticspipelines". In: *Experimental & Molecular Medicine* 50.8 (Aug. 1, 2018), pp. 1–14. ISSN: 2092-6413. DOI: 10.1038/s12276-018-0071-8. URL: https://doi.org/10.1038/s12276-018-0071-8.

[3] Malte D Luecken and Fabian J Theis. "Current best practices in single-cell RNA-seq analysis: a tutorial". In: *Molecular Systems Biology* 15.6 (June 2019). ISSN: 1744-4292, 1744-4292. DOI: 10.15252/msb.20188746. URL: https://onlinelibrary.wiley.com/doi/10.15252/msb.20188746 (visited on 01/27/2023).

[4] Malte D. Luecken et al. "Benchmarking atlas-level data integration in single-cell genomics". In: *Nature Methods* 19.1 (Jan. 2022), pp. 41–50. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-021-01336-8. URL: https://www.nature.com/articles/s41592-021-01336-8 (visited on 01/27/2023).

[5] L Sikkema et al. *An integrated cell atlas of the human lung in health and disease*. preprint. Cell Biology, Mar. 11, 2022. DOI: 10.1101/2022.03.10.483747. URL: http://biorxiv.org/lookup/doi/10.1101/2022.03.10.483747 (visited on 01/27/2023).

[6] Caltech-UW TMC Cai Long lcai@ caltech. edu 21 b Shendure Jay 9 Trapnell Cole 9 Lin Shin shinlin@ uw. edu 2 e Jackson Dana 9 et al. "The human body at cellular resolution: the NIH Human Biomolecular Atlas Program". In: *Nature* 574.7777 (2019), pp. 187–192.

[7] Corey Nolet et al. *Accelerating single-cell genomic analysis with GPUs*. preprint. Bioinformatics, May 28, 2022. DOI: 10.1101/2022.05.26.493607. URL: http://biorxiv.org/lookup/doi/10.1101/2022.05.26.493607 (visited on 01/28/2023).

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

[9] Aaron T. L. Lun, Karsten Bach, and John C. Marioni. "Pooling across cells to normalize single-cell RNA sequencing data with many zero counts". In: *Genome Biology* 17.1 (Dec. 2016), p. 75. ISSN: 1474-760X. DOI: 10.1186/s13059-016-0947-7. URL: http://genomebiology.biomedcentral.com/articles/10.1186/s13059-016-0947-7 (visited on 01/28/2023).

[10] Ka Yee Yeung and Walter L. Ruzzo. "Principal component analysis for clustering gene expression data". In: *Bioinformatics* 17.9 (2001), pp. 763–774.

[11] Ayshwarya Subramanian et al. "Biology-inspired data-driven quality control for scientific discovery in single-cell transcriptomics". In: *Genome Biology* 23.1 (Dec. 27, 2022), p. 267. ISSN: 1474-760X. DOI: 10.1186/s13059-022-02820-w. URL: https://doi.org/10.1186/s13059-022-02820-w.

[12] Matthew D Young and Sam Behjati. "SoupX removes ambient RNA contamination from droplet-based single-cell RNA sequencing data". In: *GigaScience* 9.12 (Nov. 30, 2020), giaa151. ISSN: 2047-217X. DOI: 10.1093/gigascience/giaa151. URL: https://doi.org/10.1093/gigascience/giaa151 (visited on 06/02/2023).

[13] Stephen J. Fleming, John C. Marioni, and Mehrtash Babadi. "CellBender remove-background: a deep generative model for unsupervised removal of background noise from scRNA-seq datasets". In: *bioRxiv* (Jan. 1, 2019), p. 791699. DOI: 10.1101/791699. URL: http://biorxiv.org/content/early/2019/10/03/791699.abstract.

[14] Shiyi Yang et al. "Decontamination of ambient RNA in single-cell RNA-seq with DecontX". In: *Genome Biology* 21.1 (Mar. 5, 2020), p. 57. ISSN: 1474-760X. DOI: 10.1186/s13059-020-1950-6. URL: https://doi.org/10.1186/s13059-020-1950-6.

[15] Aaron T. L. Lun et al. "EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data". In: *Genome Biology* 20.1 (Mar. 22, 2019), p. 63. ISSN: 1474-760X. DOI: 10.1186/s13059-019-1662-y. URL: https://doi.org/10.1186/s13059-019-1662-y.

[16] Samuel L. Wolock, Romain Lopez, and Allon M. Klein. "Scrublet: Computational Identification of Cell Doublets in Single-Cell Transcriptomic Data". In: *Cell Systems* 8.4 (Apr. 24, 2019), 281–291.e9. ISSN: 2405-4712. DOI: 10.1016/j.cels.2018.11.005. URL: https://www.sciencedirect.com/science/article/pii/S2405471218304745.

[17] Christopher S. McGinnis, Lyndsay M. Murrow, and Zev J. Gartner. "DoubletFinder: Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest Neighbors". In: *Cell Systems* 8.4 (Apr. 24, 2019), 329–337.e4. ISSN: 2405-4712. DOI: 10.1016/j.cels.2019.03.003. URL: https://www.sciencedirect.com/science/article/pii/S2405471219300730.

[18] Adam Gayoso and Jonathan Shor. "GitHub: DoubletDetection". In: *Zenodo* (2019).

[19] Romain Lopez et al. "Deep generative modeling for single-cell transcriptomics". In: *Nature Methods* 15.12 (Dec. 2018). Number: 12 Publisher: Nature Publishing Group, pp. 1053–1058. ISSN: 1548-7105. DOI: 10.1038/s41592-018-0229-2. URL: https://www.nature.com/articles/s41592-018-0229-2 (visited on 01/31/2023).

[20] Brian Hie et al. "Geometric Sketching Compactly Summarizes the Single-Cell Transcriptomic Landscape". In: *Cell Systems* 8.6 (June 2019), 483–493.e7. ISSN: 24054712. DOI: 10.1016/j.cels.2019.05.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S2405471219301528 (visited on 01/27/2023).

[21] Rickard Sandberg. "Entering the era of single-cell transcriptomics in biology and medicine". In: *Nature Methods* 11.1 (Jan. 2014), pp. 22–24. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/nmeth.2764. URL: http://www.nature.com/articles/nmeth.2764 (visited on 02/10/2023).

[22] Patrick G. Meirmans. "Subsampling reveals that unbalanced sampling affects Structure results in a multi-species dataset". In: *Heredity* 122.3 (Mar. 2019), pp. 276–287. ISSN: 0018-067X, 1365-2540. DOI: 10.1038/s41437-018-0124-8. URL: http://www.nature.com/articles/s41437-018-0124-8 (visited on 02/10/2023).

[23] Van Hoan Do, Khaled Elbassioni, and Stefan Canzar. "Sphetcher: Spherical Thresholding Improves Sketching of Single-Cell Transcriptomic Heterogeneity". In: *iScience* 23.6 (June 2020), p. 101126. ISSN: 25890042. DOI: 10.1016/j.isci.2020.101126. URL: https://linkinghub.elsevier.com/retrieve/pii/S2589004220303114 (visited on 01/27/2023).

[24] Dongyuan Song et al. "scSampler: fast diversity-preserving subsampling of large-scale single-cell transcriptomic data". In: *Bioinformatics* 38.11 (May 26, 2022). Ed. by Olga Vitek, pp. 3126–3127. ISSN: 1367-4803, 1460-2059. DOI: 10.1093/bioinformatics/btac271. URL: https://academic.oup.com/bioinformatics/article/38/11/3126/6569076 (visited on 01/27/2023).

[25] V. A. Traag, L. Waltman, and N. J. van Eck. "From Louvain to Leiden: guaranteeing well-connected communities". In: *Scientific Reports* 9.1 (Mar. 26, 2019), p. 5233. ISSN: 2045-2322. DOI: 10.1038/s41598-019-41695-z. URL: https://www.nature.com/articles/s41598-019-41695-z (visited on 02/02/2023).

[26] M.E. Johnson, L.M. Moore, and D. Ylvisaker. "Minimax and maximin distance designs". In: *Journal of Statistical Planning and Inference* 26.2 (Oct. 1990), pp. 131–148. ISSN: 03783758. DOI: 10.1016/0378-3758(90)90122-B. URL: https://linkinghub.elsevier.com/retrieve/pii/037837589090122B (visited on 02/08/2023).

[27] Abdel Aziz Taha and Allan Hanbury. "An Efficient Algorithm for Calculating the Exact Hausdorff Distance". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.11 (Nov. 1, 2015), pp. 2153–2163. ISSN: 0162-8828, 2160-9292. DOI: 10.1109/TPAMI.2015.2408351. URL: http://ieeexplore.ieee.org/document/7053955/ (visited on 02/05/2023).

[28] Jun Ren et al. "A downsampling method enables robust clustering and integration of single-cell transcriptome data". In: *Journal of Biomedical Informatics* 130 (June 2022), p. 104093. ISSN: 15320464. DOI: 10.1016/j.jbi.2022.104093. URL: https://linkinghub.elsevier.com/retrieve/pii/S1532046422001095 (visited on 02/10/2023).

[29] Mohammad Lotfollahi et al. "Mapping single-cell data to reference atlases by transfer learning". In: *Nature Biotechnology* 40.1 (Jan. 2022), pp. 121–130. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/s41587-021-01001-7. URL: https://www.nature.com/articles/s41587-021-01001-7 (visited on 02/02/2023).

[30] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. Sept. 17, 2020. arXiv: 1802.03426[cs,stat]. URL: http://arxiv.org/abs/1802.03426 (visited on 02/02/2023).

[31] Etienne Becht et al. "Dimensionality reduction for visualizing single-cell data using UMAP". In: *Nature Biotechnology* 37.1 (Jan. 2019), pp. 38–44. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/nbt.4314. URL: http://www.nature.com/articles/nbt.4314 (visited on 02/02/2023).

[32] Vincent D Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 9, 2008), P10008. ISSN: 1742-5468. DOI: 10.1088/1742-5468/2008/10/P10008. URL: https://iopscience.iop.org/article/10.1088/1742-5468/2008/10/P10008 (visited on 02/02/2023).

[33] Caterina Prelli Bozzo et al. "IFITM proteins promote SARS-CoV-2 infection and are targets for virus inhibition in vitro". In: *Nature Communications* 12.1 (July 28, 2021), p. 4584. ISSN: 2041-1723. DOI: 10.1038/s41467-021-24817-y. URL: https://www.nature.com/articles/s41467-021-24817-y (visited on 02/02/2023).

[34] Kavitha Mukund et al. "Immune Response in Severe and Non-Severe Coronavirus Disease 2019 (COVID-19) Infection: A Mechanistic Landscape". In: *Frontiers in Immunology* 12 (Oct. 13, 2021), p. 738073. ISSN: 1664-3224. DOI: 10.3389/fimmu.2021.738073. URL: https://www.frontiersin.org/articles/10.3389/fimmu.2021.738073/full (visited on 02/02/2023).

[35] Stephanie Pfaender et al. "LY6E impairs coronavirus fusion and confers immune control of viral disease". In: *Nature Microbiology* 5.11 (July 23, 2020), pp. 1330–1339. ISSN: 2058-5276. DOI: 10.1038/s41564-020-0769-y. URL: https://www.nature.com/articles/s41564-020-0769-y (visited on 02/02/2023).

[36] Leonid Gozman et al. "A Role of Variance in Interferon Genes to Disease Severity in COVID-19 Patients". In: *Frontiers in Genetics* 12 (Sept. 17, 2021), p. 709388. ISSN: 1664-8021. DOI: 10.3389/fgene.2021.709388. URL: https://www.frontiersin.org/articles/10.3389/fgene.2021.709388/full (visited on 02/02/2023).

[37] Rahul Satija et al. "Spatial reconstruction of single-cell gene expression data". In: *Nature Biotechnology* 33.5 (May 2015), pp. 495–502. ISSN: 1087-0156, 1546-1696. DOI: 10.1038/nbt.3192. URL: http://www.nature.com/articles/nbt.3192 (visited on 02/02/2023).

[38] Ilya Korsunsky et al. "Fast, sensitive and accurate integration of single-cell data with Harmony". In: *Nature Methods* 16.12 (Dec. 2019), pp. 1289–1296. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-019-0619-0. URL: http://www.nature.com/articles/s41592-019-0619-0 (visited on 02/02/2023).

[39] Kyle J. Travaglini et al. *A molecular cell atlas of the human lung from single cell RNA sequencing*. preprint. Genomics, Aug. 27, 2019. DOI: 10.1101/742320. URL: http://biorxiv.org/lookup/doi/10.1101/742320 (visited on 02/06/2023).

[40] L Sikkema et al. "An integrated cell atlas of the human lung in health and disease". In: *bioRxiv* (Jan. 1, 2022), p. 2022.03.10.483747. DOI: 10.1101/2022.03.10.483747. URL: http://biorxiv.org/content/early/2022/03/11/2022.03.10.483747.abstract.

[41] Malte D. Luecken et al. "Benchmarking atlas-level data integration in single-cell genomics". In: *Nature Methods* 19.1 (Jan. 1, 2022), pp. 41–50. ISSN: 1548-7105. DOI: 10.1038/s41592-021-01336-8. URL: https://doi.org/10.1038/s41592-021-01336-8.

[42] Simon Anders and Wolfgang Huber. "Differential expression analysis for sequence count data". In: *Genome Biology* 11.10 (Oct. 27, 2010), R106. ISSN: 1474-760X. DOI: 10.1186/gb-2010-11-10-r106. URL: https://doi.org/10.1186/gb-2010-11-10-r106.

Appendix

Quality control

Table 4 and 5 contain the values for the nine metrics evaluated for each of the atlases in Section 3.3

Metrics measuring removal of batch effect					
Atlas	ASW label/batch	PCR batch	graph connectivity	iLISI	mean
All cells, 1	0.9037	0.9636	0.7588	0.3176	0.7359
All cells, 2	0.9019	0.9636	0.7555	0.3215	0.7356
All cells, 3	0.9050	0.9657	0.7584	0.3231	0.7380
All cells, 4	0.9038	0.9641	0.7559	0.3208	0.7361
All cells, 5	0.9046	0.9643	0.7401	0.3237	0.7332
All cells, 6	0.9062	0.9621	0.7408	0.3186	0.7319
Cleaned, 1	0.8990	0.9625	0.7411	0.3088	0.7279
Cleaned, 2	0.8982	0.9632	0.7459	0.3114	0.7297
Cleaned, 3	0.9020	0.9701	0.7608	0.3167	0.7374
Cleaned, 4	0.9003	0.9647	0.7649	0.3073	0.7343
Cleaned, 5	0.9000	0.9627	0.7542	0.3034	0.7301
Cleaned, 6	0.9036	0.9649	0.7578	0.3074	0.7334

Table 4: scIB metrics, which measure how well the batch effects are removed.

Metrics measuring conservation of biological variance						
Atlas	NMI cluster/label	ASW label	isolated label F1	isolated label silhouette	cLISI	mean
All cells, 1	0.5550	0.4675	0.1853	0.5623	0.9766	0.5493
All cells, 2	0.5656	0.4671	0.1783	0.5636	0.9769	0.5503
All cells, 3	0.5601	0.4683	0.1836	0.5626	0.9763	0.5502
All cells, 4	0.5405	0.4704	0.1770	0.5556	0.9764	0.5440
All cells, 5	0.5622	0.4702	0.1866	0.5581	0.9761	0.5506
All cells, 6	0.5569	0.4706	0.1738	0.5597	0.9767	0.5475
Cleaned, 1	0.5655	0.4672	0.1430	0.5633	0.9767	0.5431
Cleaned, 2	0.5702	0.4664	0.1713	0.5571	0.9770	0.5484
Cleaned, 3	0.5641	0.4676	0.1393	0.5646	0.9768	0.5425
Cleaned, 4	0.5602	0.4675	0.1731	0.5584	0.9769	0.5472
Cleaned, 5	0.5666	0.4687	0.1772	0.5626	0.9774	0.5505
Cleaned, 6	0.5595	0.4671	0.1795	0.5576	0.9771	0.5481

Table 5: scIB metrics, which measure how well the biological variance is conserved.

Comparative analysis of subsampling single cell sequencing data

The following figures 24-26 visualize the performance of the Leiden-cluster-dependent subsampling and the two 'smart' subsampling methods (Sphetcher, scSampler) for either 20

or 50 principle components. The key insight is that rare cell types are mostly upsampled, while common cell types are downsampled. The dotted line indicates the random subsampling fraction (3.68%), it can be used as a reference to evaluate the other methods. If the percentage in the plots is larger, then the cell type is upsampled. On the contrary, when the percentage is smaller, the cell type is downsampled.

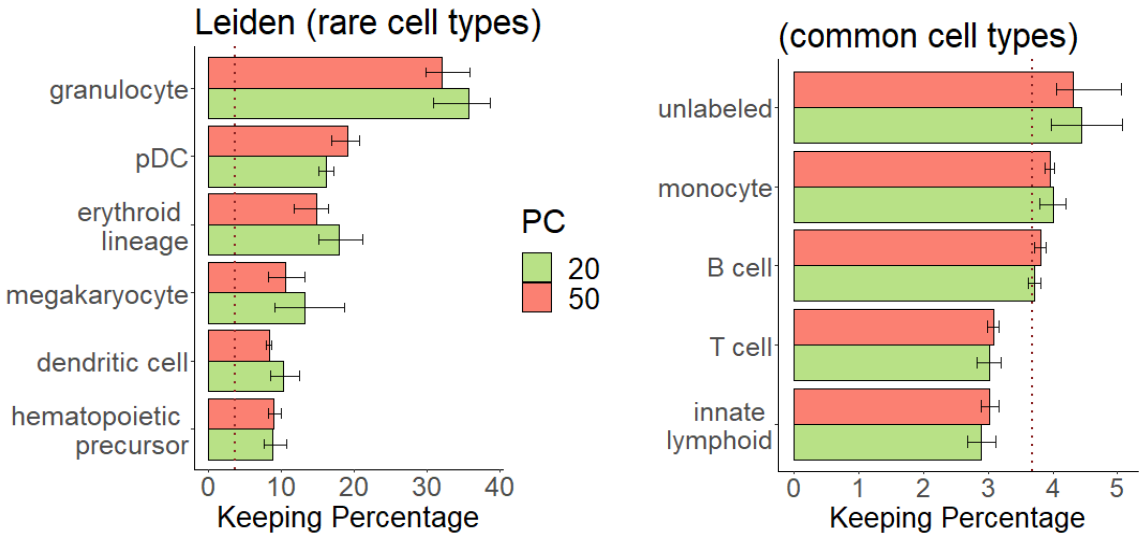


Figure 24: Percentage of cell types kept from the Leiden-cluster-dependent subsampling

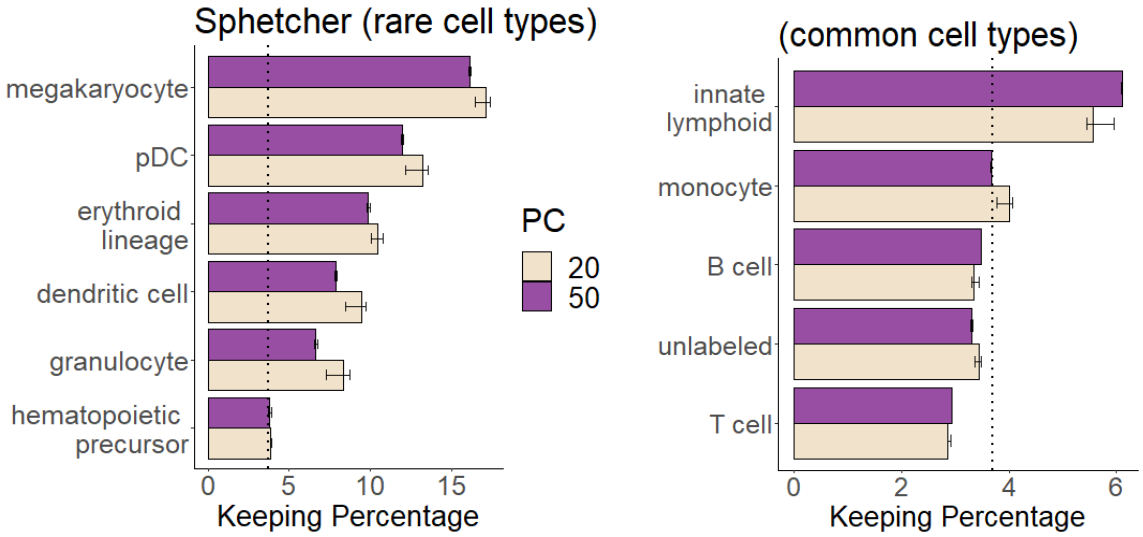


Figure 25: Percentage of cell types kept from the Sphetcher algorithm

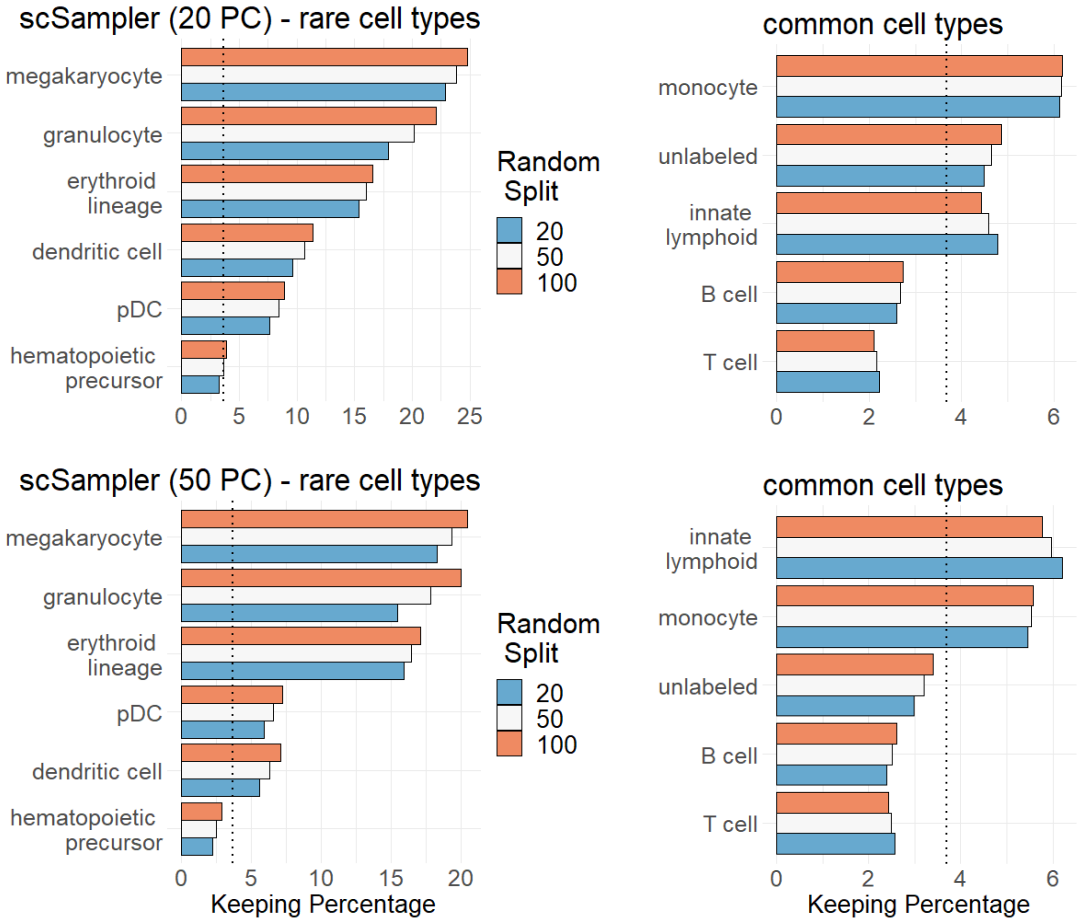


Figure 26: Percentage of cell types kept from the scSampler algorithm for 20 and 50 PC