Clustering of Cerebrospinal Fluid Cell Types in Patients With Neurological Sequelae

CSE 185 Final Project Report
Benjamin Hadler

Introduction

A known side effect from getting COVID-19 is the development of neurological sequelae resulting from infection of Sars-Cov-2[1]. As of September 2020, there were 901 reported COVID-19 patients with a wide array of neurological manifestations[3]. Neuro-COVID patients reported symptoms varying from headaches and dizziness to seizures, ischemic stroke, and intracerebral hemorrhage[1].

The etiology of these conditions is still unknown. There are 3 potential suggested mechanisms: (1) complications of systemic disease because of oxygen deprivation, sepsis, or hyperpyrexia, (2) direct viral damage to nervous system cells because of COVID-19, and/or (3) indirect damage due to an excessive and harmful immune activation[3]. In order to learn more about the possible causes, I want to carry out single-cell sequencing and further analysis on immune cell profiles from the cerebrospinal fluid (CSF) of COVID-19 patients who developed neurological sequelae. The Neuro-COVID patient data was compared to control data from other patients with idiopathic intracranial hypertension (IIH), brain autoimmune disease multiple sclerosis (MS), and viral encephalitis (VE).

I focus on cerebrospinal fluid because it forms a unique immune compartment that surrounds the central nervous system (CNS) and is helpful when sampled for understanding neurological diseases[4]. It is unique in the sense that $CD4^+$ T-cells are the most abundant in CSF. It is also extremely rare to find SARS-CoV-2 RNA in the CSF, which can indicate that SARS-CoV-2 may not be the direct cause of damage to nervous system cells. Applying single cell RNA sequencing to the CSF while focusing on T-cells will be helpful in figuring out the development of neurological diseases.

One important aspect of the CSF I focused on was the increased presence of T-cells. Studies have shown that patients with COVID-19 display T cell exhaustion when their blood is examined[1]. On the other hand, exhausted or dysfunctional T cells arise because of repetitive over-stimulation during chronic infection[1]. By analyzing the T cells in the CSF of Neuro COVID patients using T cell single cell sequencing, I can glean information regarding the pathways of how the patient developed neurological sequelae. Furthermore, this information could serve potential avenues for future research regarding therapeutics and targets for neurological sequela.

I found that Umaps made with both the Louvain and Leiden clustering methods grouped the T cells close to each other and separate from the other cell types. This shows that the T cells are significantly different compared to the other cells and are worth taking note of. I also found genes that were highly expressed in each of the T cells and can serve as genetic markers. This will make further analysis and identification of areas with higher / lower concentrates of T cells easier.

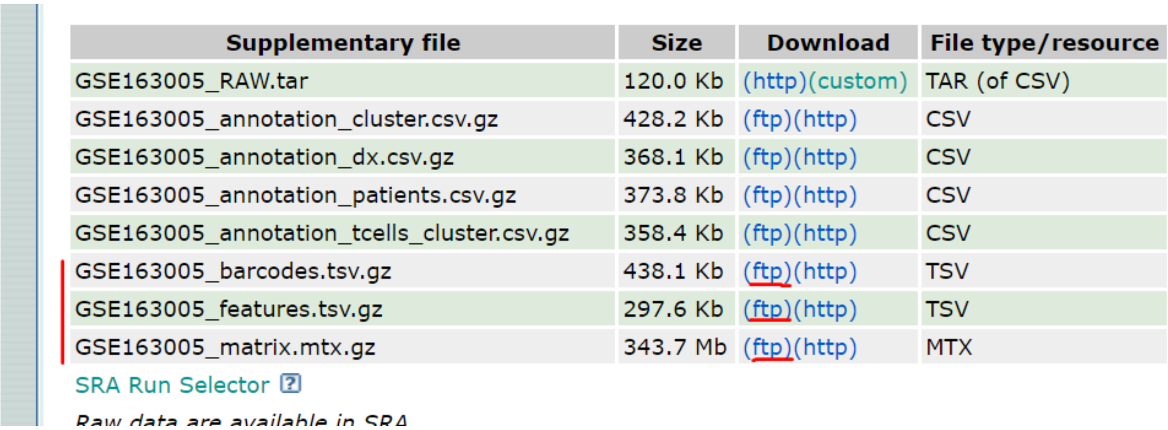
In this project, I attempted to recreate a Umap of t-cells found in the CSF of patients experiencing neurological sequelae, including those suffering from neurocovid, using both python and R (including Seurat). I was able to create a umap of all cells found in the CSF of the samples using python but I was unable to filter out all cells excluding the t-cells. Additionally, I was unable to use R packages to create the Umaps.

Methods

Aggregate Umap (python):

Fetching the data:

The first figure I attempted to recreate was figure 1b from the research paper. The first step I took was downloading the matrix file as well as the barcodes and features files from the dataset linked in the study. I then took the three zipped files and put them into a folder similar to the filtered feature barcode matrix in lab 6.



Supplementary file	Size	Download	File type/resource
GSE163005_RAW.tar	120.0 Kb	(http)(custom)	TAR (of CSV)
GSE163005_annotation_cluster.csv.gz	428.2 Kb	(ftp)(http)	CSV
GSE163005_annotation_dx.csv.gz	368.1 Kb	(ftp)(http)	CSV
GSE163005_annotation_patients.csv.gz	373.8 Kb	(ftp)(http)	CSV
GSE163005_annotation_tcells_cluster.csv.gz	358.4 Kb	(ftp)(http)	CSV
GSE163005_barcodes.tsv.gz	438.1 Kb	(ftp)(http)	TSV
GSE163005_features.tsv.gz	297.6 Kb	(ftp)(http)	TSV
GSE163005_matrix.mtx.gz	343.7 Mb	(ftp)(http)	MTX

SRA Run Selector

Raw data are available in SRA

Preprocessing the data:

I then used scanpy (version 1.7.2) to store the contents of the three files into an anndata object. I filtered out cells with less than 5 counts, cells that map to less than 200 genes and cells that map to more than 1200 genes. Next, I calculated the percentage of reads mapping to mitochondrial genes expressed in each of the cells. Mitochondrial genes in the features in the dataset were denoted by “MT-” in the beginning. A column for percentage of mitochondrial genes mapped by the reads was added to the observations dataframe of the anndata object. The anndata object was then filtered again in order to only contain cells that had a percentage of reads mapping to mitochondrial genes under 10%. Cells with 25000 or more reads were also filtered out. The anndata object was then normalized per cell to 10,000 reads per cell using the normalize_per_cell function in scanpy. I then log transformed the anndata object using the log1p function in scanpy. Next, I used the highly variable genes function in scanpy in order to add columns to the var dataframe of the filtered anndata object representing variance of the genes. Afterwards, I filtered the anndata object again to only include genes that are highly variable as determined by the highly variable genes function. I then regressed out effects of total counts per cell and the percentage of mitochondrial genes expressed with sc.pp.regress_out and scaled each gene to unit variance while clipping values exceeding a standard deviation of 10 via sc.pp.scale (max_value=10).

Clustering:

I computed the neighborhood graph of cells using a PCA representation of the data matrix with the neighbors functions in scanpy with 40 pcs as was done in the research article. I then used scanpy to run louvain clustering (resolution=.95) and leiden clustering (resolution=.8).

Plotting:

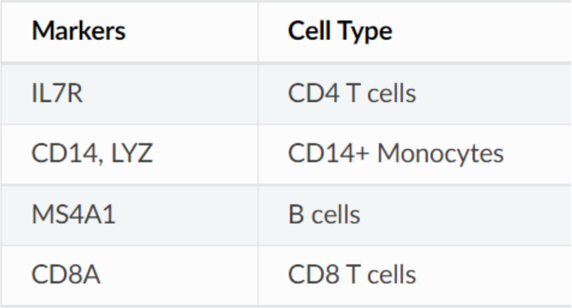
Next, I created a umap to embed the neighborhood graph in 2 dimensions. I preemptively ran three suggested lines of code recommended on the scanpy tutorials page in the case of disconnected clusters or other clustering abnormalities.

```
tl.paga(adata)
pl.paga(adata, plot=False)  # remove `plot=False` if you want to see the coarse-grained graph
tl.umap(adata, init_pos='paga')
```

I then used scanpy to create and plot the umap using the two clustering methods. (Louvain and Leiden). I chose to use both because Louvain was the one mentioned by the paper but Leiden is considered to be a more advanced clustering method(2).

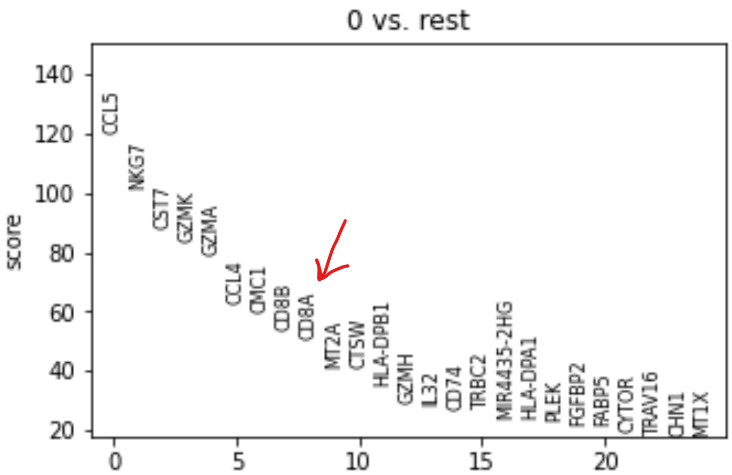
T-cell Umap(python)

After plotting the umap, we attempted to recreate figure 3a, a umap of just the two t-cells identified in the paper through the first step (CD4 and CD8). The plan to do this was to be able to identify which of the louvain clusters corresponded to the CD4 and CD8 cells and to filter the anndata object to only include those cells and then run the clustering methods again and create another umap. We tried using a list of marker genes listed on the scanpy tutorial web page (2):

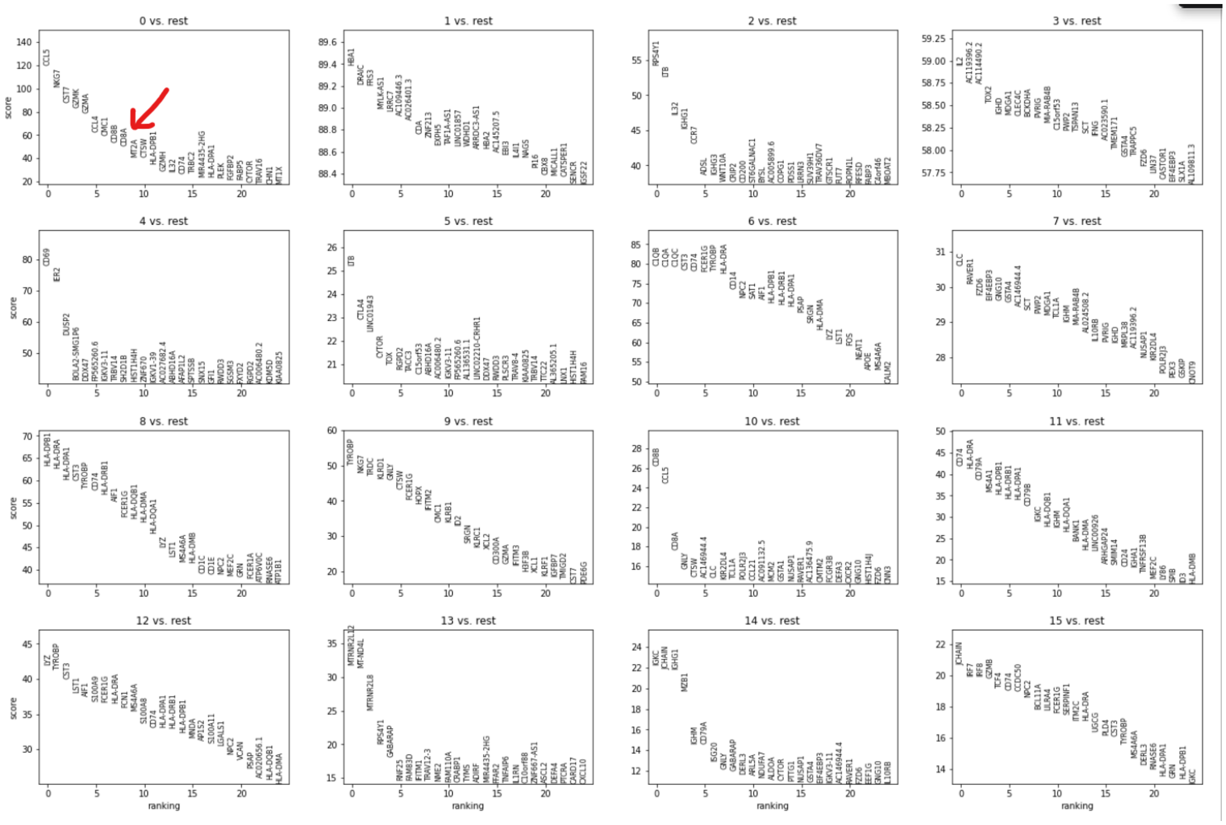


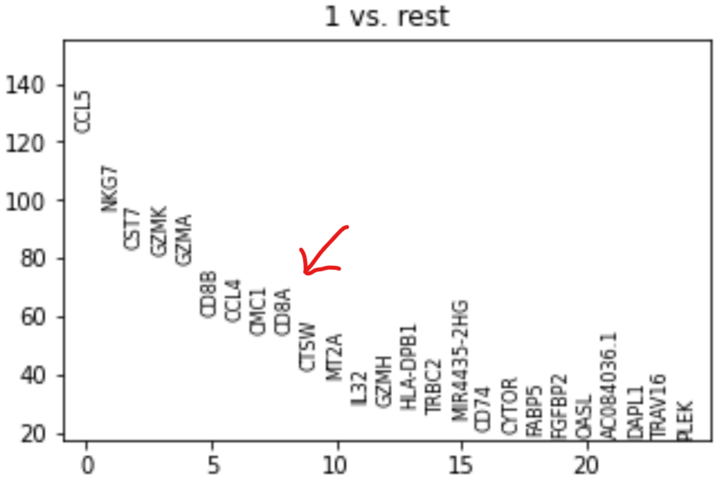
Markers	Cell Type
IL7R	CD4 T cells
CD14, LYZ	CD14+ Monocytes
MS4A1	B cells
CD8A	CD8 T cells

In order to name the clusters, I computed rankings for highly differential genes across the 16 clusters and saw if any of the clusters contained the highly ranked genes. Using the wilcoxon method on the leiden and louvain clusters with the function rank_gene_groups on scanpy I were able to find a cluster associated with CD8A, the marker gene for CD8 T cells (cluster 0).

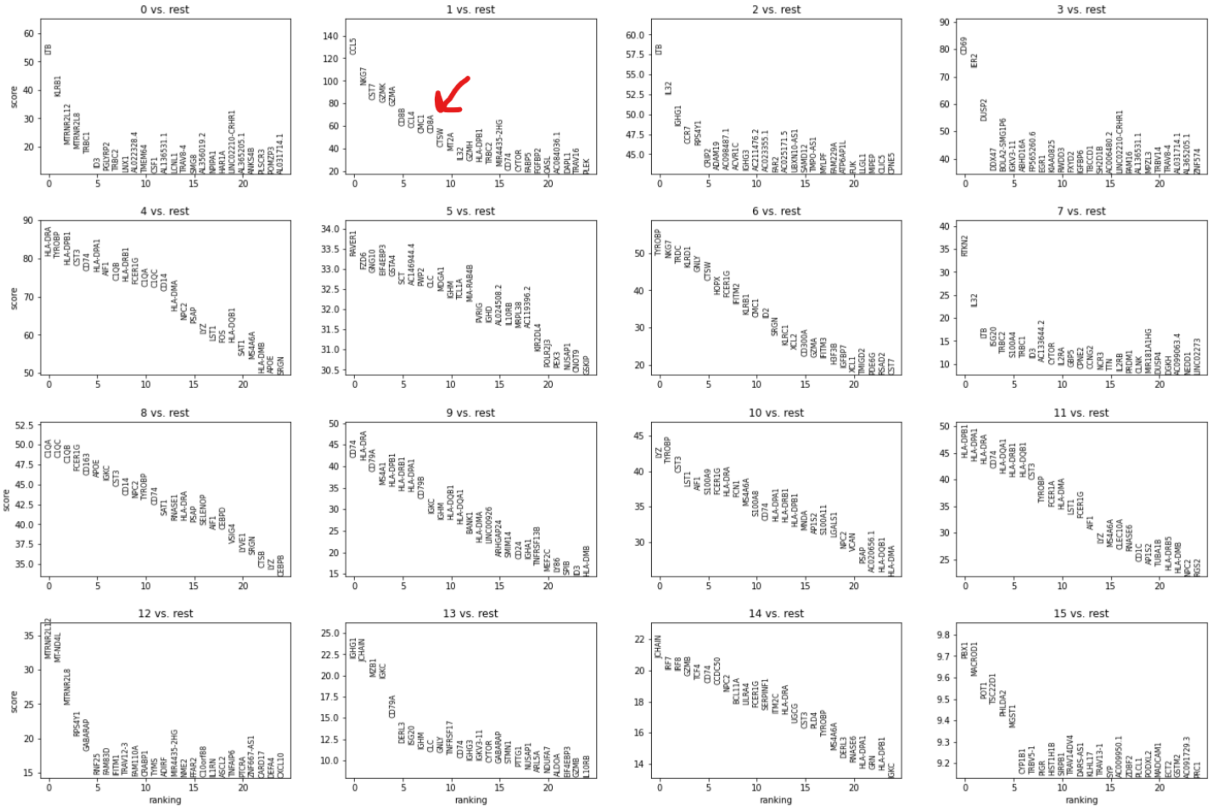


(Louvain, wilcoxon)





(Leiden, wilcoxon)



Using both wilxocon and other tests such as t-test, I was reliably able to find clusters associated with CD8 T Cells but I was never able to find any clusters differentially expressing IL7R. Thus, I was unable to proceed with the pipeline and reproduce figure 3a.

Results

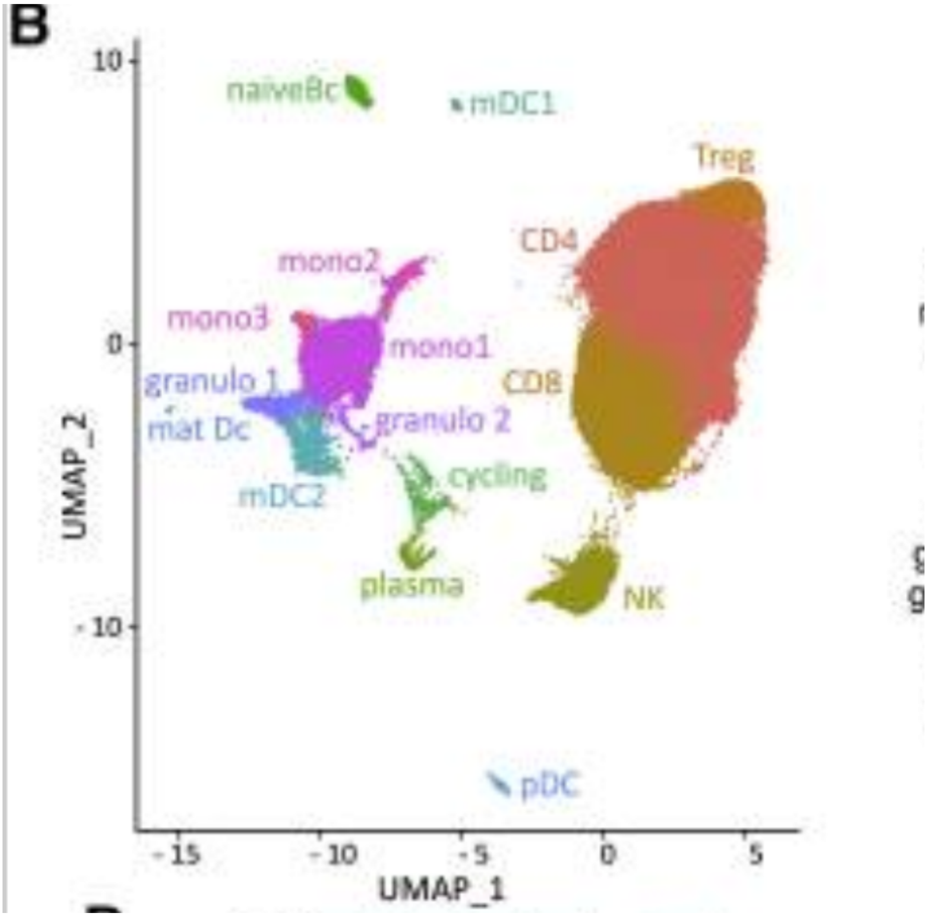


Figure 1: Figure 1B from the Neurological Manifestations of COVID-19 Feature T Cell Exhaustion and Dedifferentiated Monocytes in Cerebrospinal Fluid paper. This figure shows the UMAP plot showing 16 color-coded cell clusters of 80,919 raw single-cell transcriptomes from CSF cells from N-COVID (n = 8), idiopathic intracranial hypertension (IIH) (n = 9), multiple sclerosis (MS) (n = 9), and viral encephalitis (VE) (n = 5) patients.

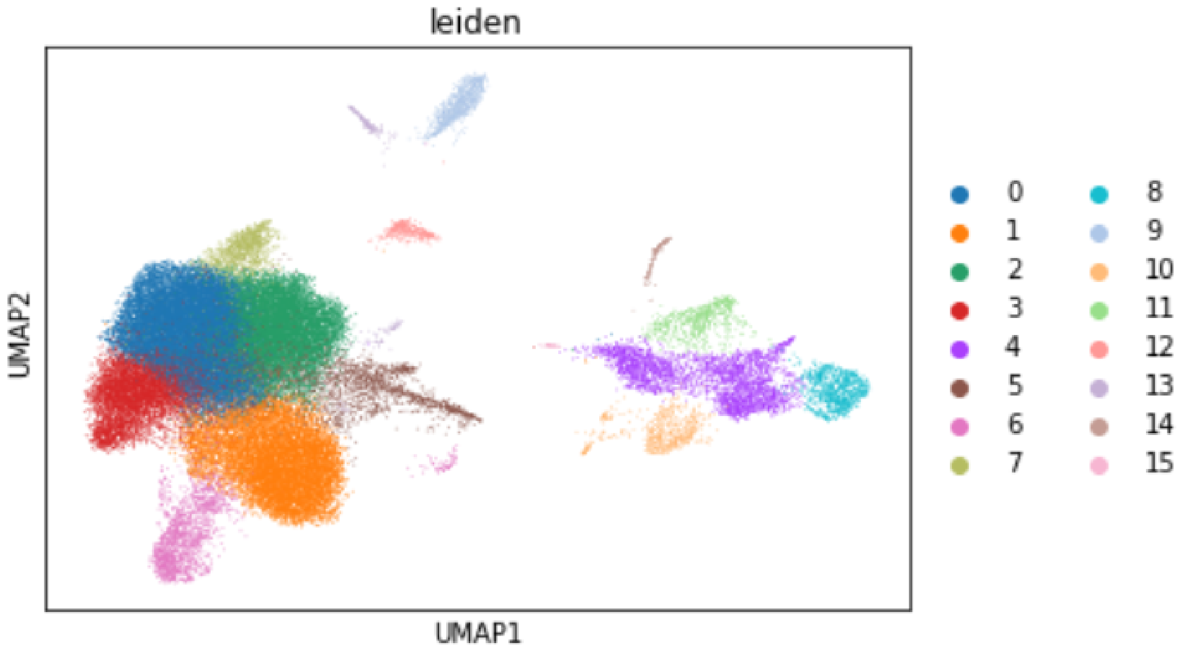


Figure 2: This is the figure I produced to replicate figure 1. I used the leiden method to generate the clustering. 1 represents CD8. However, I was unable to label any other clusters.

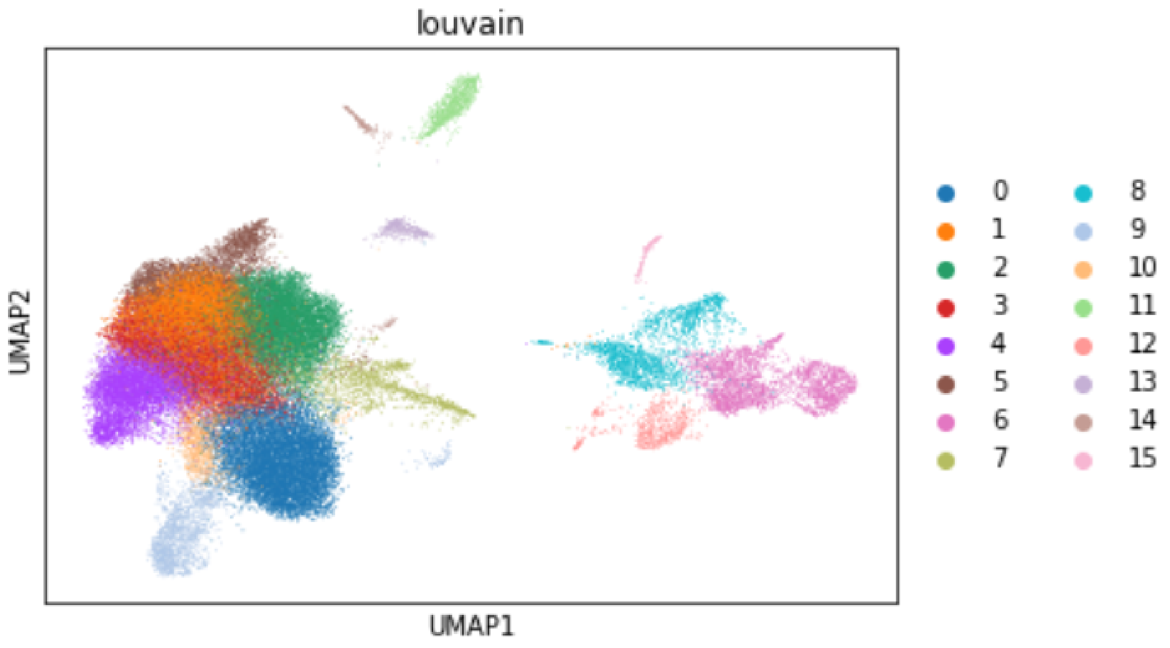


Figure 3: This is a figure I produced to replicate figure 1. I used the louvain method to generate the clustering. 0 represents CD8. However, I was unable to label any other clusters.

Discussion

In the very beginning, I tried to implement the pipeline using both R and python. I then faced some challenges in visualization when using R. Some of us had trouble with loading the data or had totally different looking figures. Therefore, I decided to switch to python to process and visualize the data. After changing to python, I was able to get the data and visualize them. However, I was still unable to replicate the figure 1 exactly. The potential limitations would be the parameters. The paper had a lack of clarity regarding the parameters they used. I used the basic parameters using the guide from “Preprocessing and Clustering 3k PBMCs” and lab 6 from the class in order to circumvent this. I referenced the tsne visualization where I used scanpy to run louvain clustering (resolution=.95) and leiden clustering (resolution=.8). Also, I regressed out effects of total counts per cell and the percentage of mitochondrial genes expressed with sc.pp.regress_out and scaled each gene to unit variance while clipping values exceeding a standard deviation of 10 via sc.pp.scale (max_value=10). In addition, I possibly used different versions of tools. For example, I used scanpy (version 1.7.2) to store the contents of the three files into an anndata object while the paper used Seurat and R for all its downstream analysis. This paper was published in 2020. So, it is possible that the paper used different versions of tools before the update.

The paper mentioned that researchers filtered the data on a sample by sample basis. Certain cutoffs were given as ranges and as a result, I was forced to generalize and use one cutoff. Examples of this include the filtering of genes in the beginning of the pipeline. The paper mentions “Each sample was filtered individually to remove cell doublets and low-quality cells with few genes (< 200) high genes (> 1200-6000) or high mitochondrial percentages (5%–20%).”[1]. I was forced to use 1200 as a cutoff for the number of genes considered too high and 10% for the cutoff of what I considered too high of a percentage of reads mapping back to mitochondrial genes.

One other deviation from the paper occurred during the regression analysis. I regressed the number of reads coming from each cell in addition to the percentage of mitochondrial reads, but the paper only mentioned regressing based on mitochondrial reads in the cells.

Additionally, I set the max value to 10 when using the scale function in scanpy which removed all values exceeding a standard deviation of 10, but this was not mentioned in the paper.

Lastly, I used different resolutions than were listed in the paper. They used Louvain clustering only and set the resolution to .3. They also performed the clustering only in R, which possibly could have performed differently. I chose the resolution using a guess and check method in order to get a number of clusters that matched what the paper got (16 clusters).

All these deviations from the paper, most importantly the difference in software (R and seurat vrs Python), may have contributed to my inability to reproduce the figures. I do believe that if I was able to filter sample by sample the same way they did in the study I would be able to get more similar results.

References

1. Heming, Michael et al. “Neurological Manifestations of COVID-19 Feature T Cell Exhaustion and Dedifferentiated Monocytes in Cerebrospinal Fluid.” Immunity vol. 54,1 (2021): 164-175.e6. doi:10.1016/j.immuni.2020.12.011

2. “Preprocessing and Clustering 3k PBMCs.” *Preprocessing and Clustering 3k PBMCs - Scanpy Documentation*, scanpy-tutorials.readthedocs.io/en/latest/pbmc3k.html.

3. Ellul, Mark A et al. “Neurological associations of COVID-19.” The Lancet. Neurology vol. 19,9 (2020): 767-783. doi:10.1016/S1474-4422(20)30221-0

4. Ransohoff, Richard M, and Britta Engelhardt. “The anatomical and cellular basis of immune surveillance in the central nervous system.” Nature reviews. Immunology vol. 12,9 (2012): 623-35. doi:10.1038/nri3265