Meghana Peshwa

BF528 - FINAL REPORT

Section 1 – Project 4 Data Curator

Introduction

The original paper aims to determine the various subpopulations of human and mouse pancreatic cells as well as gain a better understanding of the gene expression program of these cells. Single-cell transcriptomes of more than 12,000 pancreatic cells were determined from four human donors and two mouse strains by using the inDrop method. These reads need to be quantified and the UMI counts matrix is required to perform analysis and determine the cell clusters. This section focuses on how the UMI counts matrix was generated using the provided single-cell RNA-seq samples in this study.

Methods

The sample metadata was obtained from the GEO accession number GSE84133. Only the SRR files associated with the 51-year-old female donor were selected for analysis. The corresponding RNA-seq fastq files were provided.

The paper used a complex barcoding scheme for read 1. The raw read 1 barcode was processed and provided in files with names that look like <run>_1_bc.fastq.gz. These files were first used to count the number of reads per unique barcode. The number of reads per unique barcode was found separately for all 3 samples as well as for all samples combined. Then these barcodes were whitelisted by eliminating the infrequent barcodes with the help of CDF plots. This was done to remove noninformative barcodes and potential bias due to low-quality cells.

After this, an index was generated using the salmon index command and the human reference transcriptome available on gencode. A transcript ID to gene map file was also created since it is required to generate the UMI counts matrix. Finally, using the whitelist.txt, index, map file, and the fastq files in the salmon alevin command, the UMI matrix was generated.

Results

There were a few barcodes that had lengths greater than 19. These barcodes were discarded since the expected length was 19. CDF plots were made to understand the knee point to whitelist barcodes (Figure 1 and 2). From the CDF plots for the unique barcodes of each sample, and the combined unique barcodes (of all samples), the knee was deduced to be 30. Hence, the barcodes with read counts less than 30 were filtered out. Before filtering there were 2833359 distinct barcodes, and after filtering this reduced to 315631 barcodes. This final dataframe was the whitelist.txt that was used as an input to salmon alevin.

Around 13% of noisy cellular barcodes were discarded during the salmon alevin process. The mean UMIs and genes per cell were 86 and 61 respectively. The mapping rate was found to be 42.5% which is quite low. This could be due to possible contamination or the presence of relatively short reads. Baron et al. did not mention anything about the mapping rate so this result may be what is expected.

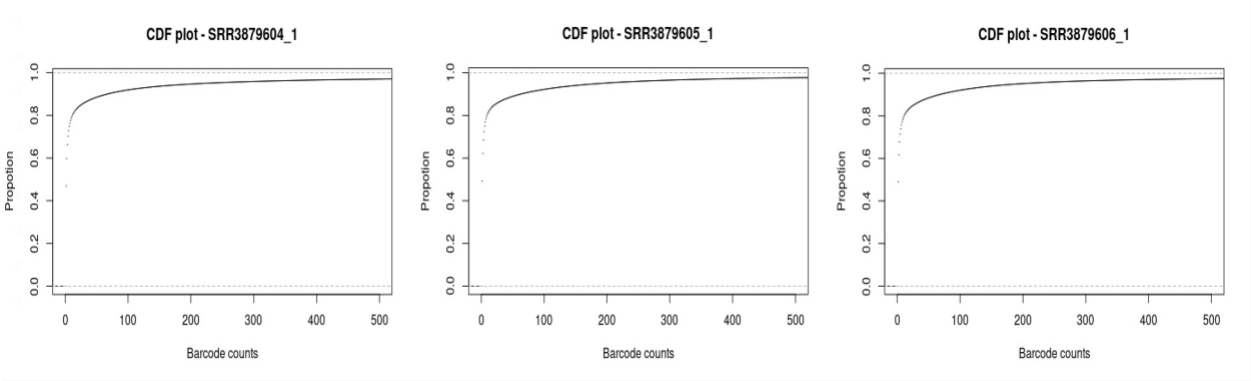

Figure 1. CDF plots for the unique barcodes of each sample

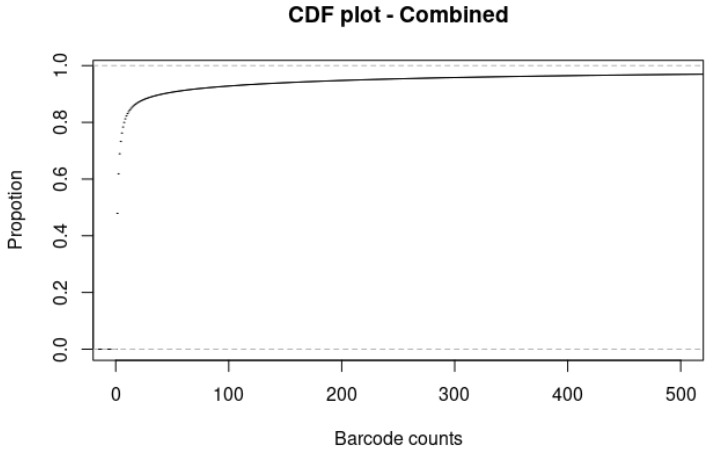

Figure 2. CDF plots for the combined unique barcodes of all samples

Table 1. Important salmon alevin statistics

Total reads	1324837961
Final number of CBs	312092
Noisy cellular barcodes	13.3267%

UMI after deduplicating	26978343
Mean UMIs per cell	86
Mean genes per cell	61
Mapping rate	**42.5413%**

Conclusion

The goal of this section was to create a UMI counts matrix from the raw reads. The counts matrix was successfully generated using salmon alevin. The mapping rate was low, but perhaps this is expected for this experiment.

Section 2 – Project 4 Programmer

Introduction

This section is based on the same paper as the last section. Baron et al. followed a custom methodology for performing analysis on the single cell RNA-seq data. For this project, the tutorial written by the authors of the Bioconductor package Seurat was followed instead. This section mainly deals with performing quality control, preprocessing, and clustering using the UMI counts matrix generated in the previous section. These tasks are essential for further analysis such as finding marker genes, and labeling clusters based on cell type.

Methods

The UMI counts matrix was loaded into R using the tximport library. The emsembl gene IDs were renamed to their respective gene symbols using BioMart. There were challenges in doing this as not all gene IDs will have gene symbols. The gene IDs were retained for those that did not map to a gene symbol. One way of overcoming this would have been to use gene symbols in the data curation section when generating the map file.

Initial filtering of the genes was done to make sure the genes have nonzero counts in at least 10 cells. This ensures that the low expression genes are eliminated. After this, the Seurat object was created. To filter cells, a few QC metrics were adapted. One of these was filtering cells based on the percentage of reads that map to the mitochondrial genome. Cells with mitochondrial counts greater than 0.9% were filtered out. Based on Figure 3, the cells that have nCount_RNA values less than 500 and nFeature_RNA values not between 200 and 3500 were filtered out too. These were probably low-quality cells, dead cells, or reads with possible contamination.

After this, the Seurat object was normalized using log normalization. This is required to find the highly variable features. Feature selection was then performed which returned 2000 most variable features. The number of features used for filtering was 2000 since 2000 features tend to work well with most datasets and this is Seurat's default value. The object was then scaled by linear transformation. This pre-processing step is required to perform PCA by RunPCA(). Figure 5 shows an elbow plot that depicts the variance explained by each principal component.

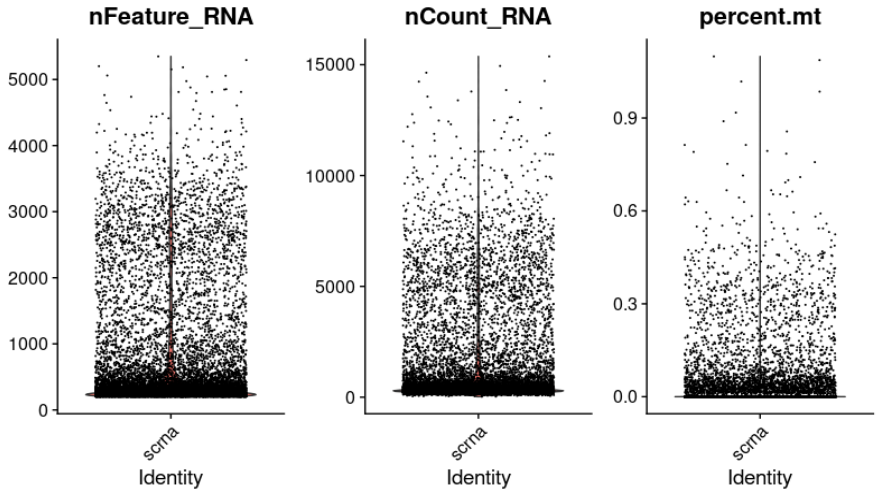

Figure 3. QC metrics visualized as violin plots

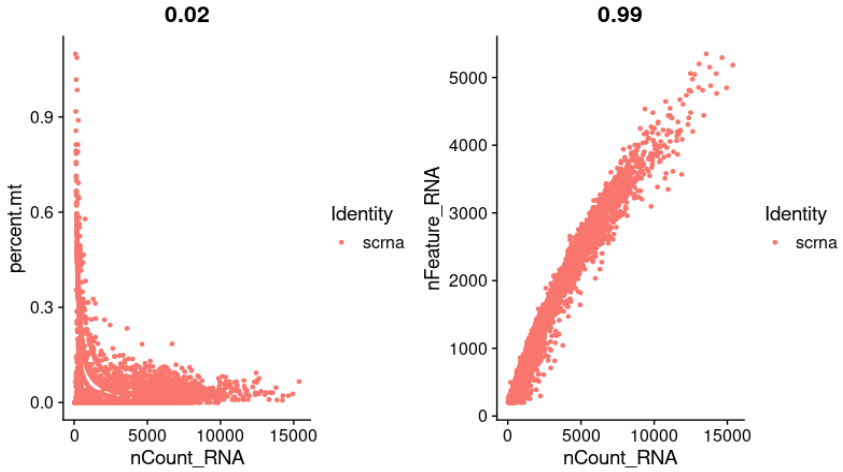

Figure 4. Scatter plots to visualize QC metrics relationships.

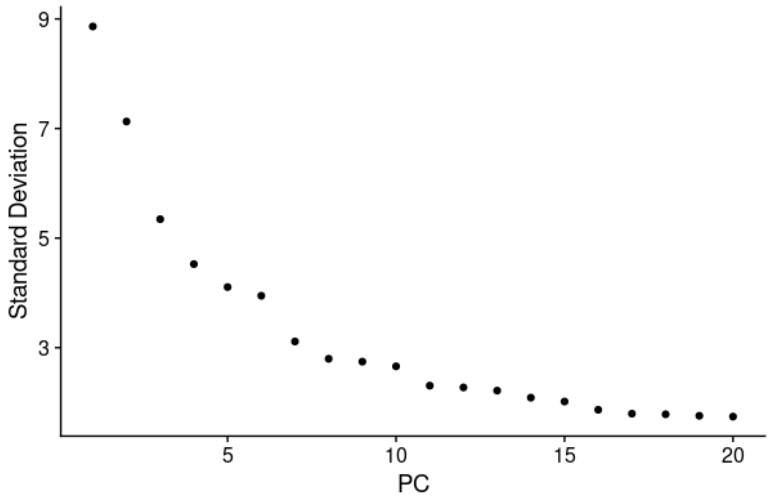

Figure 5. Elbow plot where the elbow is seen at PC7

Finally, to cluster cells, k-nearest neighbor method was applied using the FindNeighbors() function followed by FindClusters() which uses a modularity optimization technique to iteratively cluster cells. A bar graph was plotted to visualize the cell numbers in each cluster.

Results

Initially, there were 287952 cells and 61125 genes. After performing filtering there remained 24814 cells and 4449 genes. Figure 6 shows the number of cells present in each cluster. Totally there were 12 clusters identified. Cluster 0 has the highest number of cells (>650 cells, 19%). Cluster 11 has the lowest number of cells (around 40 cells, 1%). Every cluster has fewer cells compared to the previous cluster.

When compared to the original paper, Baron et al. reported 15 clusters. This difference could be due to several reasons. Firstly, a smaller subset of samples (only the 3 51-year-old female donor samples compared to 13 human samples in Baron et al.) was analyzed in this case. Also, there could have been differences in the way the UMI counts matrix was generated. Lastly, the authors had a custom pipeline for preprocessing and analyzing the data, which is different from the methodology used here.

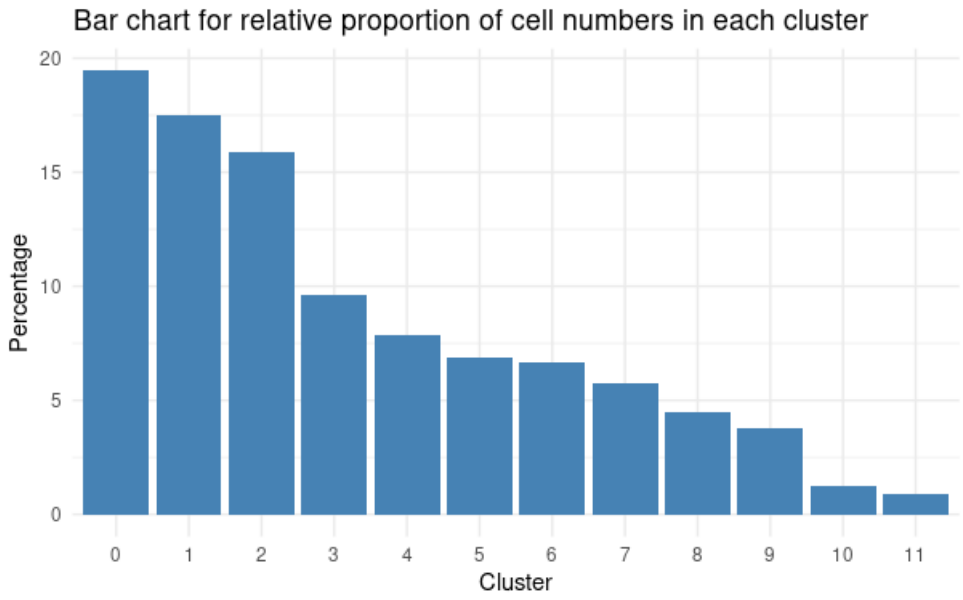

Figure 6. Bar chart representing the relation proportion of cell numbers in each cluster.

Conclusion

Overall, the Seurat object underwent QC analysis and was filtered based on meaningful criteria to make it ready for further analysis. 12 clusters were identified. These can further be labeled based on cell type, and marker genes can also be found.

References

1. Baron M, Veres A, Wolock SL, et al. A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure. *Cell Syst*. 2016;3(4):346-360.e4. doi:10.1016/j.cels.2016.08.011
2. Alevin documentation, https://salmon.readthedocs.io/en/latest/alevin.html
3. "Seurat – Guided Clustering Tutorial", April 17, 2020, https://satijalab.org/seurat/archive/v3.1/pbmc3k_tutorial.html