Project 5 – Individual Project
Single Cell Analysis of Pancreatic Cells

Daisy Wenyan Han

ENG BF528 | Spring 2021

Table of Contents

Introduction

The ongoing advancements in RNA-sequencing and single-cell technologies have allowed for more in-depth studies of diverse cell types, many of which had previously only been studied via bulk RNA-sequencing. In their 2016 study, *A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter- and Intra-cell Population Structure*, Baron et al. implemented the then-ground-breaking method of inDrop single-cell sequencing to identify additional cell types within the mammalian pancreas, demarcate their molecular identifiers, and thus generate a database of novel, cell-type specific information to be used in future analyses. When applied to the human pancreas, this database would be instrumental in the search for treatments for well-known pancreatic diseases, such as Type I and II Diabetes Mellitus.

Baron et al. analyzed the pancreatic cells of four individuals. For our analysis, we will be re-examining the data from one such individual in the study and evaluating the cellular makeup of the corresponding samples in order to corroborate the initial findings using more advanced, updated techniques.

More specifically, the *Data Curator* and *Programmer* roles for this project will be performed. This includes tasks such as the processing of the raw data and the quality control of the UMI count matrix, such that downstream analyses can be undertaken without the worry of contamination or other poor-quality data influencing the final results. These tasks are important for the overall study because should any biases be introduced in any of the earlier processing or quality control steps, these errors will then propagate throughout the rest of the study, thereby leading to misconstrued or inaccurate results. It is therefore vital that these steps be performed in a clear, coherent manner, and that any results obtained be logical and reproducible.

Methods

I was originally in the *Analyst* role for Project 4, which was responsible for the cell type identification of the data produced by the *Data Curator* and *Programmer*. Therefore, I chose to attempt to reproduce the results in the *Data Curator* and *Programmer* roles, as well as implement my pre-existing *Analyst* code, to see if I could obtain the same results as those obtained by my group in our initial recreation of the Baron et al. analysis in Project 4.

Data Curator – Process Reads and Choose Barcodes

Locate Sample Metadata

A total of thirteen sequencing libraries, obtained from four post-mortem human donors, were made available to us on the Boston University Shared Computing Cluster. For our analysis, we used only the files associated with the 51-year-old female donor. In order to determine the Sample IDs corresponding to this donor, we examined the supplementary data provided by the original authors. Baron et al.'s Supplementary Table 1 details the ages, BMIs and sexes of the four human donors from whom samples were obtained. The donor corresponding to the 51-year-old female was labelled as "Donor 2". While this provided context to Baron et al.'s figures, it did not specify which sample identification numbers corresponded to this particular donor. For this information, we were therefore required to access the data online.

The datasets and corresponding metadata are publicly available via the Gene Expression Omnibus under Accession Number GSE84133. The metadata for these files were then accessed using the SRA Run Selector module on the NCBI webpage. Filtering was applied on each of the runs for the appropriate sex, age and organism, which for this project, were female, fifty-one and Homo sapiens, respectively. This provided us with three Sample IDs corresponding to our donor in question. These were Runs SRR3879604, SRR3879605 and SRR3879606, which were then located on the SCC, to be used in downstream analysis.

While we were not required to download the files, their metadata was available on the SRA Run Selector, with information pertaining to their sample collection methods and respective library sizes. Samples were obtained using the inDrop methodology developed by Klein et al. in 2015, just one year prior to the publishing of Baron et al.'s study. All three FASTQ files contained paired-end reads, run on an Illumina Hiseq 2500 machine. They ranged in size from 36.47Gbp to 56.12Gbp, with SRR3879606 as the smallest, and SRR3879604 as the largest.

Count Number of Reads by Barcodes

In order to count the number of reads assigned to each barcode in each of the three FASTQ files, we required a method by which we could iterate through each line of the three gzipped FASTQ files, without decompressing them. To achieve this, we utilized the AWK command on the command line, in conjunction with ZCAT. AWK allows for commands to be written in the form of statements which define text patterns to be searched for in each line of a particular document. In this exercise, AWK was used to isolate each UMI-barcode pair, which was located on the second line of every four-line read. The barcode consisted of the first nineteen characters of the first element of the line in question, with the UMI consisting of the six characters immediately following.

The CUT, SORT and UNIQ commands on the command line were then used to sort the barcodes alphabetically, isolate a list of barcodes, and then count the number of times each barcode appeared. This count corresponded to the number of UMIs per unique barcode, which was used to inform the generation of a whitelist of barcodes to be used in further analysis.

Whitelist Informative Barcodes

It is imperative to eliminate reads with infrequent barcodes from consideration, as lowly represented barcodes are likely to be due to noise and therefore detract from meaningfully differentiated gene expression. Therefore, a whitelist of barcodes was required in order to retain only the reads that would be most informative. To best inform our decision in choosing a threshold number of UMIs per barcode, a set of summary statistics were generated for each of the three samples, as represented in *Table 1* below.

Run	Minimum	First Quartile	Median	Mean	Third Quartile	Maximum
SRR3879604	1	3701	7726	29343	14874	1604234
SRR3879605	1	3082	6394	23285	12807	992905
SRR3879606	1	3004	6200	22092	12699	1288069

Table 1 – Summary Statistics on the number of UMIs per barcode. These values were used to inform the threshold value to be used in the generation of the barcode "whitelist". Ultimately, the mean for each sample, which fell between 22092 and 29343, was used as a lower bound for the number of UMIs per barcode to be included in the analysis.

It appears from our summary that our data is skewed rightwards, as the mean number of UMIs per barcode is greater than the median. Therefore, in order to filter out the majority of low-quality, uninformative cells, we felt it would be best to set the threshold value at the mean number of UMIs per unique barcode for each sample. This would allow for a fairly large number of barcodes to be used in later analysis, while also removing those that fall below the average. This value was approximately in line with the recommended sequencing depth by the UC Davis Bioinformatics Core, which recommends a sequencing depth of approximately 30,000 reads per barcode for RNA-poor cellular subtypes, such as PBMCs (Settles, 2017).

This whitelist of barcodes was then provided to the Salmon Alevin program, in the following step.

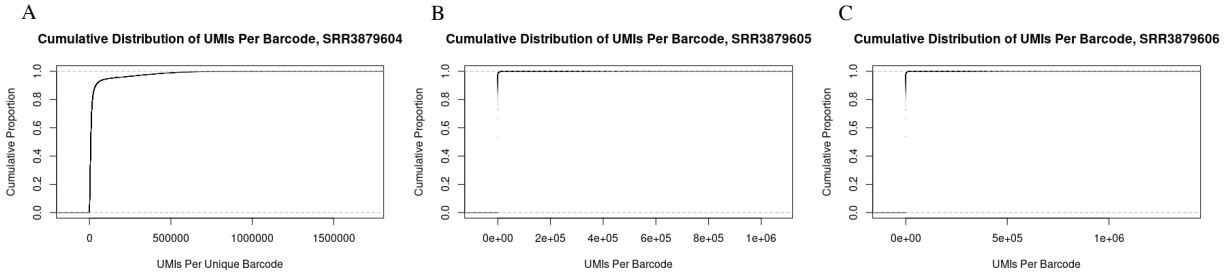


Figure 1 – Cumulative Distribution Plot of UMIs per barcode for each of the three samples. Note the steep increase in each of these samples, indicating that the distribution of reads is heavily skewed rightwards. It is therefore logical to use the mean, rather than the median, as the threshold value for generating the whitelist of barcodes.

UMI Counts Matrix Generation

A UMI counts matrix was generated using the Salmon Alevin program. Alevin is a tool within the Salmon software that allows for the quantification and analysis of single-cell sequencing data obtained using the Drop-Seq and 10X-Chromium protocols (Patro et al., 2017). As input, Salmon Alevin requires the raw FASTQ files, the previously generated whitelist of barcodes, an index file containing an index of the reference transcriptome, and a transcript-to-gene map, which matches each transcript identifier with a corresponding gene name.

Both the raw FASTQ files and the whitelist of barcodes were available for each of the three samples by this point in the analysis. An index of the reference transcriptome was generated using the "salmon index" command, contained in the Salmon package. This was accomplished using the GTF file of comprehensive gene annotations of version 37 of the human genome, available on the GENCODE webpage. The transcript-to-gene map was generated using a FASTA file containing the primary assembly of the same version of the human genome sequence. The AWK command was then again used to parse this large file; the transcript IDs and gene names were located in the eleventh and twelfth columns in this file, respectively. This produced a TSV file containing transcript IDs and their corresponding gene names, which was later filtered such that there were no repeats of identical transcript ID-gene name pairs.

Additional parameters, provided for the class, were also included in the Salmon Alevin command. These parameters specified a barcode length of nineteen characters, and a UMI length of six characters. This process took approximately 30 minutes, whilst running on sixteen cores. Version 1.1.0 of Salmon was used for this analysis.

Mapping Statistics

The Salmon Alevin program generated a log of summary statistics for the program run. Of the inputted data, there was a total of 4628 whitelisted barcodes. Utilizing an internal filtering method, a total of 41.6792% of the reads were thrown away due to noisy cellular barcodes. This resulted in 95949 cellular barcodes, of the possible 4251176, being passed down for further analysis. The UMIs were also filtered by the Salmon Alevin program. Of the possible 1324837961 reads, 39950 contained "N" values, and were discarded from further analysis. Therefore, only 772508229 reads were used.

A deduplicating step was then performed by the Salmon Alevin program, resulting in a total of 13393191 UMIs being retained for downstream analysis. The final mapping statistics are summarized in *Table 2* below, with more detailed data available in Supplementary Table 1.

Mean UMIs Per Cell	Mean Genes Per Cell	Mapping Rate
2893	1446	30.919392111228917

Table 2 – Summary Statistics Generated by Salmon Alevin during UMI Counts Matrix Generation. A more complete table of summary statistics generated by the Salmon Alevin program is available in the Supplementary Data section.

It was slightly concerning to note that the mapping rate of our three samples was only 30.919%, indicating that over two thirds of our data had already been excluded from further analysis at this point. However, this seemed reasonable, given that the samples were obtained from cadaveric donors. It was therefore expected that some degradation had occurred prior to cell sampling. Likewise, the mean number of genes per cell had been expected to vary due to the many different cell types expected to be present prior to sampling. Moreover, because low-quality cells are to be filtered out in the following step, a lower-than-expected mapping rate may not significantly impact the quality of the cells retained.

The mean UMIS per cell was of slight concern. As per the Harvard Chan Bioinformatics Core, the typical UMI counts per cell should be above five hundred, with counts falling lower than this value likely requiring greater sequencing depth (Harvard, 2021). While resequencing the cells was not an option for us, this was kept in mind for further analysis, in that a smaller quantity of cells could be expected to be of use in downstream analyses.

Programmer – Process UMI Counts Matrix

Filter Low-Quality Cells

The UMI counts matrix generated via Salmon Alevin was then imported into R using the tximport program, available via Bioconductor. This program was chosen, as it has been shown to be faster and less memory consuming in the loading of expression data in comparison to similar programs, which require creation and storage of BAM files (Soneson, 2015).

Prior to further analysis, we felt it beneficial to map the Ensembl gene identifiers to their respective gene symbols. This required not only mapping each of the Ensembl gene identifiers to their respective genes, but also iterating through the UMI counts matrix to ensure that no genes were duplicated or overrepresented. To achieve these tasks, the EnsemblDB package was used. EnsemblDB is an R package, available via Bioconductor, that allows for the retrieval of annotations for any of the hundreds of species available through the Ensembl and EnsemblGenomes databases, using their Perl API (Rainer et al. 2019). During this process, the initial 60232 genes was decreased to 56724 following the removal of those not matching to known genes in the Ensembl database, and then 55116 following the summation of duplicate features into a single row. This final matrix was then loaded into Seurat for further quality control. This consisted of 55116 features across 4627 cells.

Seurat is an R package commonly used for single-cell RNA analysis. It allows for a streamlined exploration and filtering of cells based on user criteria (Hao et al. 2020). Version 4.0 of Seurat, recently released in 2020, was used for this analysis. For this experiment, cells were filtered based on the number of unique genes detected in each cell, the number of molecules detected per cell, as well as the percentage of reads mapping to the mitochondrial genome. Each of these three filtering criteria are visualized below, in *Figure 2*.

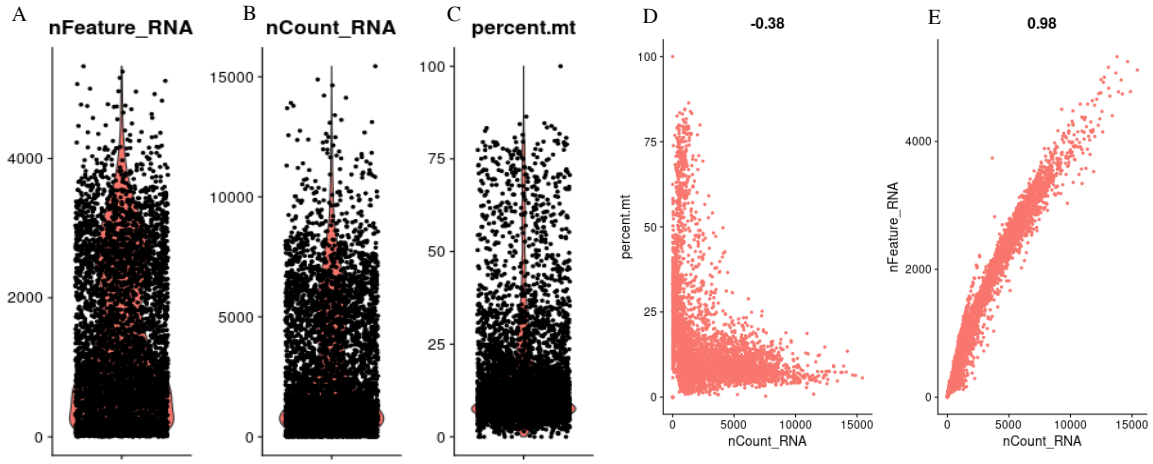


Figure 2 – Plots of common quality control statistics, including violin plots of (A) the number of unique genes detected in each cell, (B) the total number of molecules detected within each cell, and (C) the percent of mitochondrial genes per cell. Scatter plots, typically used to visualize feature-feature relationships, were also generated to visualize the correlation between (D) the number of unique genes and the mitochondrial percentage, and (E) the number of genes and the number of molecules within each cell.

The number of unique genes detected per cell is an important quality control metric, as low-quality cells or empty droplets will often have very few genes. On the contrary, cell doublets and multiplets may have an abnormally high number of genes. Therefore, it is important that cells with both aberrantly high and low numbers of genes be removed prior to further analysis. The number of molecules per cell follows the same principles and is often highly correlated with the number of unique genes, as per *Figure 2E*. It is for these

reasons that lower and upper limits of 500 and 4000 respectively were selected as bounds for the number of unique genes detected per cell. The lower bound of 500 eliminate the dense cluster of lowly expressed genes visible in *Figure 2A*, while the upper bound of 4000 eliminates the "tail" of genes observed in *Figure 2E*.

Unlike the nCount and nFeature metadata, which was precomputed by Seurat, the percentage of mitochondrial RNA had to be calculated before it could be visualized and filtered by. This was done using the PercentageFeatureSet function, contained within the Seurat package. This command calculates the percentage of counts, originating from a set of features, for each cell. Mitochondrial genes were those with gene names starting with "MT-". This is an important quality control metric, as low-quality or dying cells will often exhibit extensive mitochondrial contamination. Of note, because our samples were obtained from a cadaveric donor, an elevated mitochondrial percentage was expected for all our cells. An upper limit of 15% was set for the percentage of mitochondrial genes present based on *Figures 2C* and *D*. As per *Figure 2C*, it appears that the majority of the cells have mitochondrial percentages below 20%. This threshold was then lowered upon examination of *Figure 2D*, which shows a greater dispersion of cells in the x-direction, corresponding to a greater number of cells with larger numbers of unique molecules, once below the 15% threshold.

Filter Low Variance Genes

In order to identify variable genes that are likely to be informative in downstream analysis, the counts matrix must first be normalized to allow cells to be compared to one another. This is done using the NormalizeData command, contained in the Seurat package. This command applies a global-scaling normalization method that normalizes the expression of each cell by the total expression, multiplies this by the default scale factor of 10,000, and then log transforms the scaled result.

The above process allows for the cells to be directly compared, and for the most informative genes to be selected for further analysis. This gene selection is performed because analyses on features exhibiting high cell-to-cell variation aids in highlighting biological differences in single-cell datasets (Brennecke et al., 2013). The FindAllFeatures function, contained in the Seurat package, was therefore utilized to identify these highly variable features. The default value of 2,000 variable features per dataset was used for our analysis, as visualized in *Figure 3*. This is the same number of variable features used by Baron et al. in their clustering analysis as well.

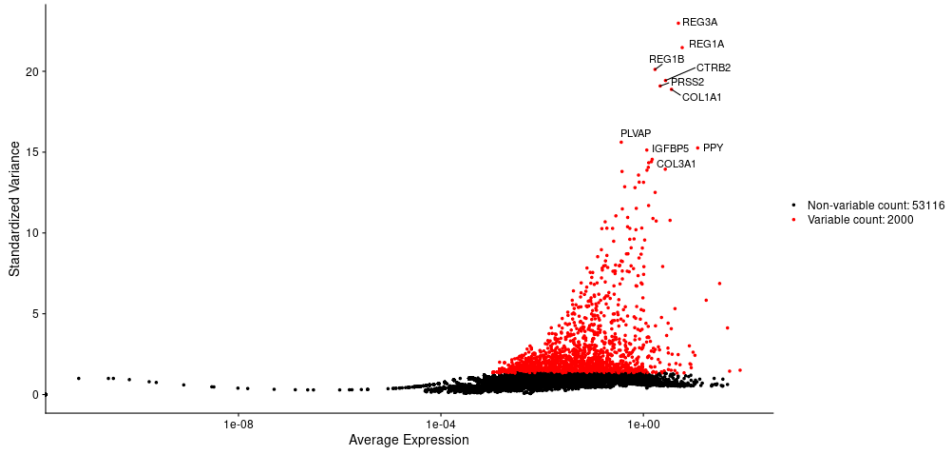


Figure 3 – Scatter Plot highlighting the 2,000 variable features selected for further analysis. The variable features are represented in red, with the ten most highly variable features labelled.

The final Seurat object, after filtering, contained 53116 features across 2771 cells, with 2000 identified variable features.

Identify Clusters of Cell Type Subpopulations

Prior to dimensional reduction techniques, the data must first be linearly transformed. This was achieved using Seurat's ScaleData function, which scales the expression of each gene such that variance across genes is one, and the mean expression of each gene across all cells is zero. This step ensures that highly expressed genes do not skew downstream results. By default, this was carried out on the 2000 variable genes, identified above.

Linear dimensionality reduction was performed by running Principal Component Analysis (PCA) on the scaled data using Seurat's RunPCA command. The dimensionality of the dataset was then determined by examining *Figure 4*, below.

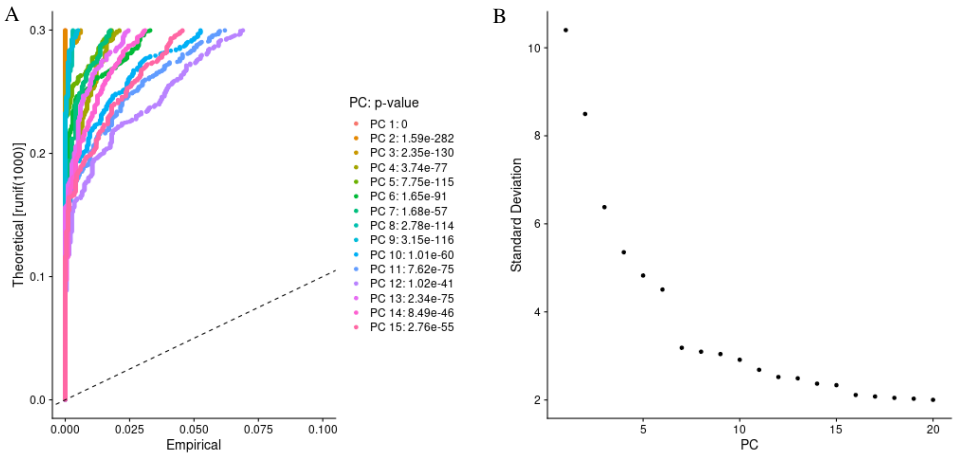


Figure 4 – (A) Jackstraw and (B) Elbow plots visualizing the first fifteen and twenty principal components of the dataset. These plots were used in the determination of principal components to be used in further analysis.

From *Figure 4A*, there does not appear to be a significant drop-off in significance within the first fifteen PCs visualized, though the significance of each subsequent PC does appear to slowly decrease. This is more clearly visualized in *Figure 4B*, in which the standard deviation of the PCs drops steeply after the fifth PC, with what the Seurat developers described as an "elbow" occurring near the fifteenth. Therefore, fifteen dimensions were used in further analyses.

A non-linear dimensionality reduction was then performed to cluster the cells into their respective cell type subpopulations. There are several non-linear reduction techniques available through Seurat that can be used to visualize and explore this dataset. We chose to visualize our data using a UMAP rather than a tSNE, as UMAP projections preserve not only local, but also global data structure (McInnes et al., 2018). Therefore, we believed this would be more representative of the data, which consists of many closely related, and highly similar, cell types. The clustering of the cells is visualized below in *Figure 5*, which displays the eleven distinct clusters identified. More data about the individual clusters, including the number of cells contained within each cluster, can be found in the Supplementary Data.

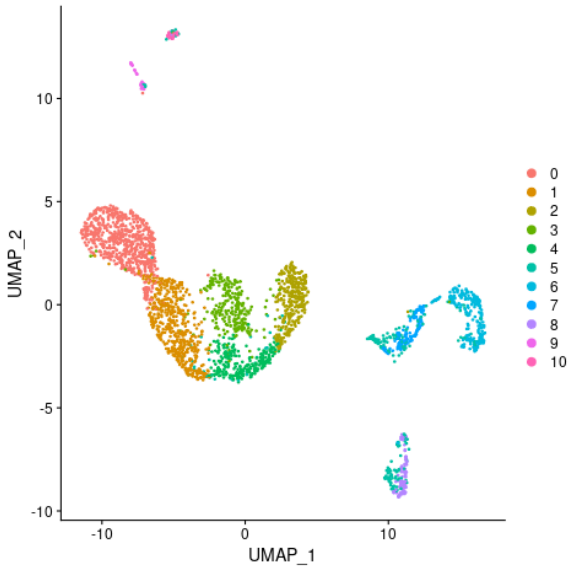


Figure 5 – UMAP Projection of the dataset. Eleven distinct clusters of cell type subpopulations are identified. Further information about the number of cells in each of the clusters is available in the Supplementary Data.

Analyst – Cluster Marker Genes

Identify Marker Genes

Marker genes were identified for each cluster using the FindAllFeatures command included in the Seurat package. This command automatically identifies differentially expressed genes in each cluster, compared to all other clusters. The default differential expression test of Seurat was used: the Wilcoxon Rank Sum Test. This is a nonparametric test used to compare outcomes between two groups, often interpreted as the comparison of the medians between the two populations tested.

A minimum percent threshold value of 0.25 was placed during the generation of the biomarkers to be used for cell type classification. This corresponds to a feature being detected at a minimum of 25% in either of the two groups of cells in order to be considered informative. The top ten marker genes for each of the twelve clusters, by average $\log_2$FoldChange, were then exported for further analysis. This corresponded to the top ten most differentially up- or down-regulated genes per cluster.

Using the above-generated marker genes, clusters were assigned their corresponding cell types. These were visualized via UMAP projection, as detailed in the Results section.

Results

Label Clusters by Cell Type Based on Marker Genes

Marker genes for each of the eleven clusters were identified using the FindAllFeatures command in Seurat, as detailed in the Methods section above. This resulted in a total of 6275 features being selected as "marker" genes. Statistics such as the p-value, adjusted p-value, average log2FoldChange, and cluster of origin were also automatically generated. The top ten marker genes, by most significant average log_2FoldChange, were exported for each of the clusters, and used to generate *Figure 6*, below.

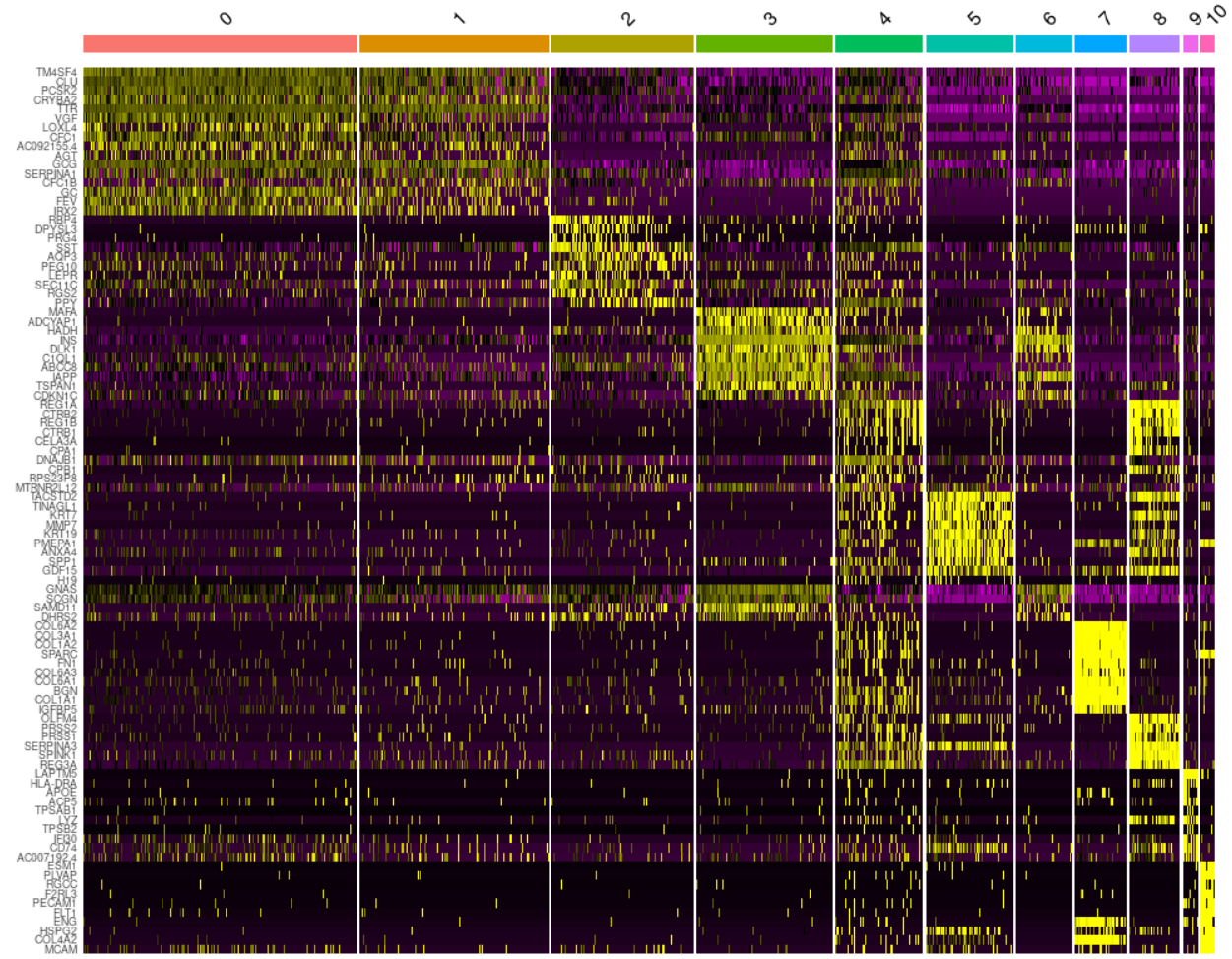


Figure 6 - Clustered heatmap of log normalized UMI counts, showing top ten differentially expressed genes in each of the eleven clusters. Genes are obtained from the top ten most differentially expressed genes in each cluster, by their average Log2FoldChange. Clusters are labelled as the color bar, ranging from Cluster 0 to Cluster 10.

The above figure was generated using the top ten most differentially expressed genes for each of the clusters, by their Log_2FoldChange. Because this figure was generated prior to labelling cells by their cell types, the "bars" present in the heatmap were used to inform various decisions in the cell type assignment process. For example, Clusters 0 and 1 appear quite similar in their differentially expressed genes. It would therefore be plausible that these two clusters belong to the same cell type. This same principle can also be applied to Clusters 3 and 6, as well as Clusters 4 and 8.

In order to best replicate the cell type identification performed by Baron et al., the table of marker genes used for cell identification during their analysis was downloaded from their supplementary data (Baron et al., 2016). Violin plots and feature plots were then generated for each of the Baron et al. genes to determine whether they matched with a particular cluster in our dataset. A complete set of violin plots and feature plots can be found in the Supplementary Data.

Of the eleven clusters identified, we were able to identify eight distinct cell types using only the marker genes provided by Baron et al. Most of these marker genes were present in distinct areas, making for clear identification of cell type. For example, the PDGFRB gene, indicative of stellate cells, was found only in Cluster 7, as confirmed by *Figure 7* below.

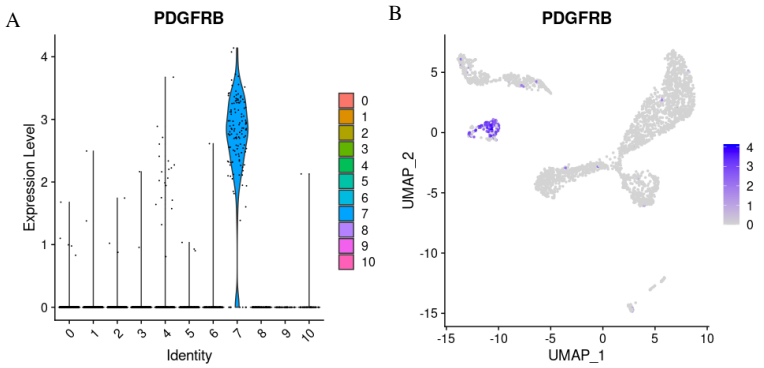


Figure 7 – Plots used in the identification of the stellate cells. Both the (A) violin plot, and (B) feature plot show that the PDGFRB gene appears to be most highly expressed in Cluster 7. As there are no conflicting clusters with this same upregulation of PDGFRB, Cluster 7 was labelled as consisting of stellate cells.

Other cell types were identified using corroborating evidence obtained from the heatmap in *Figure 6*. Clusters 4 and 8, for example, appear to be highly correlated, as per the heatmap. It was therefore unsurprising that the CPA1 gene, the marker gene for acinar cells, was highly expressed in Clusters 4 and 8. Therefore, both of these clusters were labelled as belonging to the acinar cell type. This methodology, as demonstrated in *Figure 8* below, was also used in the identification of Clusters 0 and 1, and Clusters 3 and 6, which belonged to pancreatic alpha and pancreatic beta cells, respectively.

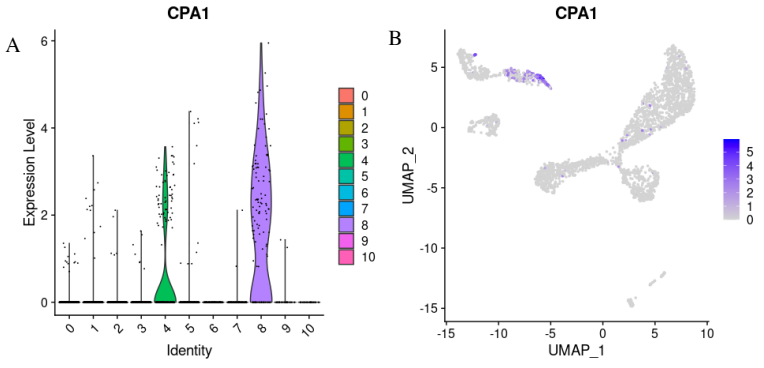


Figure 8 – Plots used in the identification of acinar cells. From Figure 6, we were aware that Clusters 4 and 8 may share the same cell type. It was therefore unsurprising that they both showed upregulation of the CPA1 gene, allowing both clusters to be labelled as acinar cells.

Using the abovementioned methods, ten of the eleven clusters were easily identified. Cluster 2, however, proved more difficult to classify into a single cell type, as it appeared that two distinct cell types were

present within this cluster. Both the SST and PPY genes, the marker genes for pancreatic delta and gamma cells, respectively, were highly enriched in Cluster 2, with both appearing to be localized in the area of the UMAP corresponding to Cluster 2, as demonstrated in *Figure 9*. Moreover, when examining the enrichment of these two genes on the UMAP projection, they appear to separate to distinct sides of the cluster, perhaps indicating that Cluster 2 is, in fact, comprised of two distinct cell types. Upon further research, it was determined that delta cells comprise approximately 10% of pancreatic islet cells, while gamma cells comprise only 5% (Elayat et al., 1995 & Brissova et al., 2005). Therefore, while it is likely that both cell types are present, we felt it best to label Cluster 2 as belonging to the delta cell type, as the majority of the cluster is expected to be comprised of delta cells.

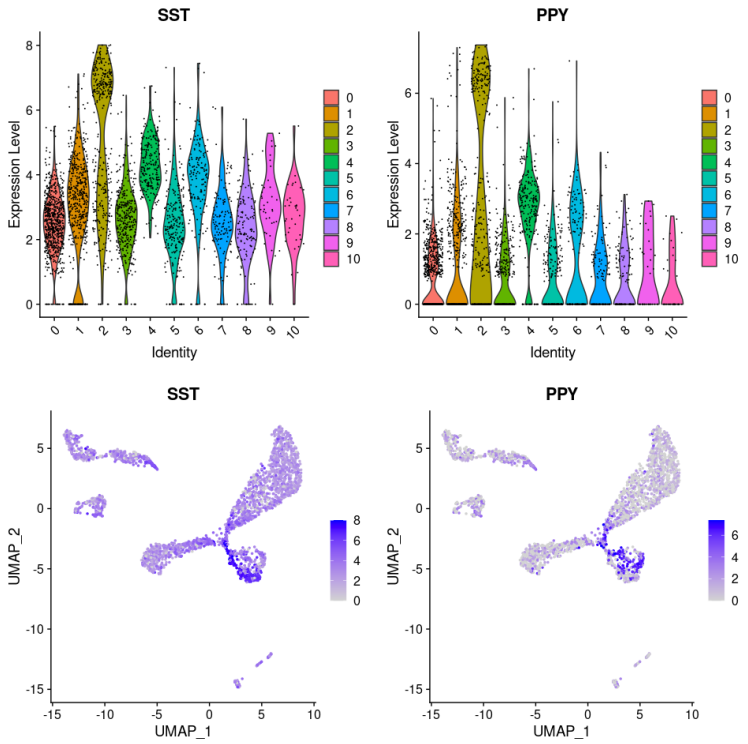


Figure 9 – Plots used in the identification of Cluster 2. Both the marker genes SST and PPY, identifiers of pancreatic delta and gamma cells, respectively, appear to be highly enriched in this cluster. The appear to be localized in different areas of the cluster on the UMAP projection, suggesting that this cluster is, in fact, comprised of two distinct cell types.

Visualize Clustered Cells

The data was then visualized on a UMAP projection, with the assigned cell types labelled, as shown in *Figure 10*. This UMAP looks quite different from the Baron et al.'s original tSNE plot, where they were able to identify fourteen distinct cell types. We were able to identify only eight of these fourteen cell types, with much less clear distinction between each one. However, it is notable that the global structure of this figure appears to be quite logical. All of the pancreatic islet cells, including the pancreatic alpha, beta and delta cells, are located closer to each other than they are to the other cells present.

As mentioned in the previous Project 4 report, particular cells, such as the pancreatic epsilon cells, were not expected to be seen, as they are not typically present in adults (Baron et al. 2016). Per Baron et al., pancreatic epsilon cells make up only 0.1% of a complex mixture of cell types, and were not expected to be seen in high concentrations (Baron et al., 2016). Cytotoxic T-cells and mast cells also appear to be missing from our analysis, perhaps indicating a lack of immune response at the time that the sample was

collected. However, the "islet preparation methods" used in the Baron et al. study are known to deplete non-islet cells, perhaps explaining the absence of these cell types in our samples.

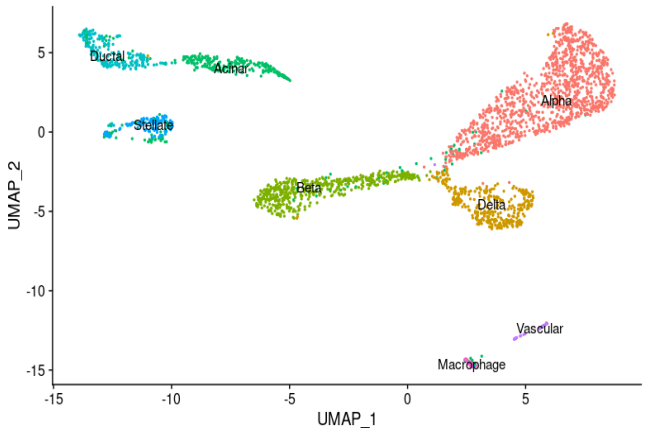


Figure 10 – UMAP projection of the dataset. Eight distinct cell types can be clearly visualized. Of note, all of the pancreatic islet cells appear to be clustered together, while the cells typically present outside the pancreatic islets, such as the vascular and macrophage cells, are further away.

Visualize Top Marker Genes Per Cluster

The top marker genes for each of the cell types is visualized in *Figure 11* below as a clustered heatmap of log normalized UMI counts for the most differentially expressed genes in each of the clusters, by their average Log_2FoldChange. It is reassuring to see clearly demarcated "bands" of gene expression across each cell type, indicating genes that are significantly up- or down-regulated in each cluster.

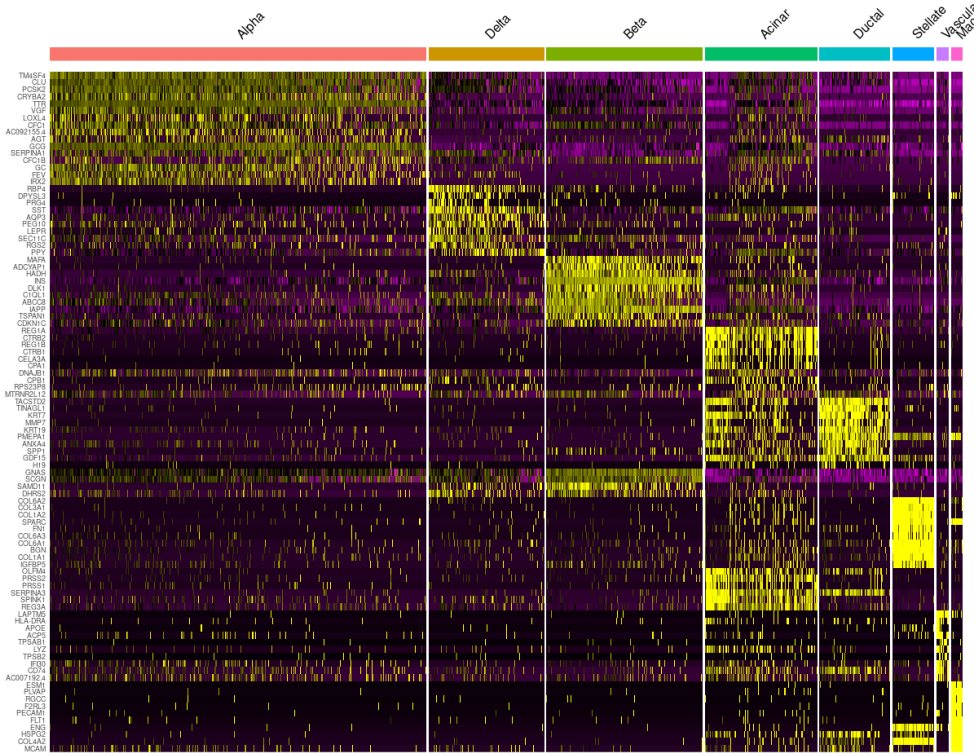


Figure 11 – Clustered heatmap of top differentially expressed genes in each of the eight cell types. Genes selected by most significant average Log_2FoldChange.

Find Novel Marker Genes

Aside from the marker genes provided by Baron et al., many additional genes were shown to be indicative of a particular cell type. *Figure 12* illustrates these genes as distinct "bands" along the x-axis. While additional testing is required to determine whether the genes present in this figure are indicative of the cell types in which they are enriched, Baron et al. were able to show promising results regarding the more detailed classification of pancreatic cell types using single-cell analysis.

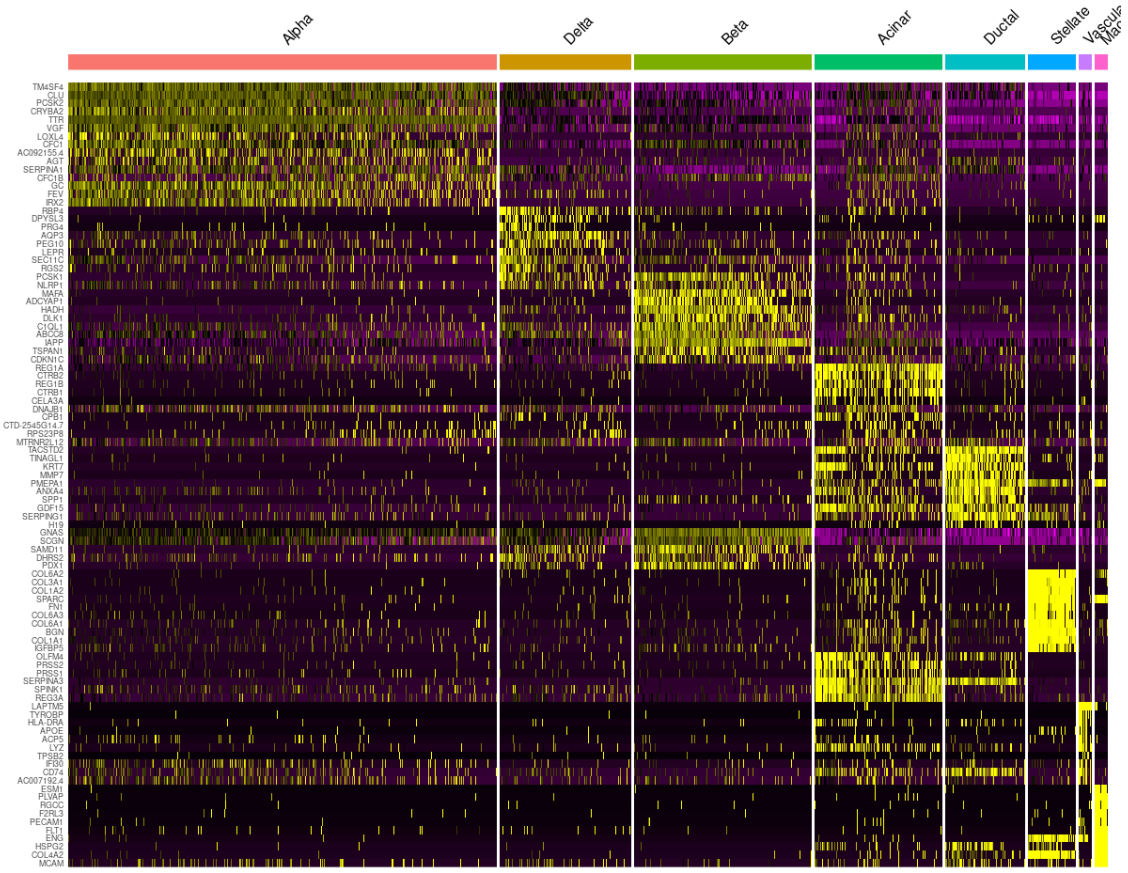


Figure 12 – Clustered heatmap of log normalized UMI count matrix, grouped by cell type. The "novel" genes represented are those with the most significant average Log2FoldChange in each of the cell types but was not included in the pre-determined "marker" genes by Baron et al. in their Supplementary Data.

Discussion

The original goal of this study was to apply newer, more advanced single-cell analysis tools to the data obtained by Baron et al. in an attempt to replicate the results of their 2016 study. To do so, the original raw FASTQ files were obtained, and underwent a series of transformations and quality control checks before clustering analysis. Newer programs and packages, such as Salmon Alevin, released 2017, and Seurat, first released in 2015 and later updated in 2020, were used in lieu of the original analysis tools (Patro et al. 2017 & Hao et al., 2020).

While we were unable to obtain the fourteen different distinct cell types seen by Baron et al., we were able to differentiate between eight different cell types using the original marker genes provided, as well as identify additional novel genes to be evaluated for their efficacy at indicating particular cells.

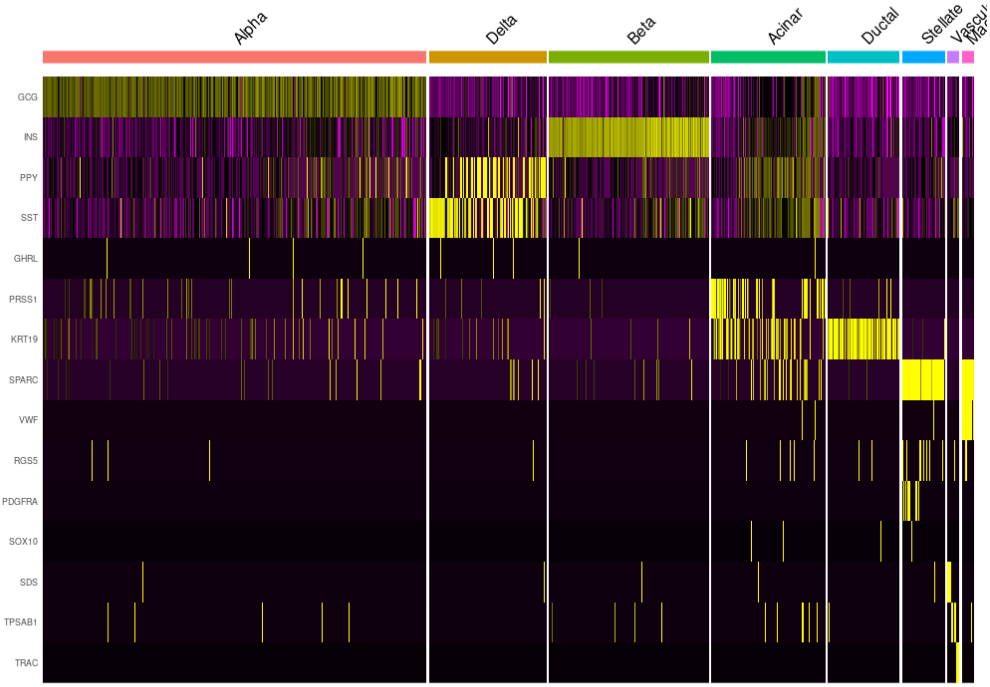


Figure 13 – Recreation of the clustered heatmap in Baron et al.'s Figure 1D. The genes along the left of the figure are the same as those in Baron et al.'s figure. However, the distinct "bars" that they observed across their cell types for each of these marker genes does not appear to be present.

Figure 13 above illustrates our recreation of Baron et al.'s *Figure 1D*. The same genes were used, in the same order, along the left side of the clustered heatmap. Some of the cell types, such as the pancreatic alpha and beta cells, appear to have clear, distinct "bands" in the heatmap along the x-axis, indicating the enrichment of their respective marker genes. We also see again the merging of the PPY and SST genes, indicative of the pancreatic delta and gamma cells, within the pancreatic delta cell cluster. It is likely that additional data is required for Seurat, or other single-cell analysis programs, to clearly distinguish between these two cell types. Other cell types also appear to require additional data in order to determine their distinctive marker genes. The acinar cells, for example, appear to be enriched across three different genes.

While the exact results from Baron et al. could not be recreated, it is reassuring to see that despite using the data from only one of their four donors, a number of different cell types could be identified. This experiment also re-affirmed the results obtained from our group project, in which we were also only able to distinguish a small subset of the genes noted by Baron et al.

References

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

Brennecke P, Anders S, Kim J. et al. Accounting for technical noise in single-cell RNA-seq experiments. Nat Methods. 2013;10:11093-1095. doi:10.1038/nmeth.2645

Brissova M, Fowler MJ, Nicholson WE, et al. Assessment of Human Pancreatic Islet Architecture and Composition by Laser Scanning Confocal Microscopy. *Journal of Histochemistry & Cytochemistry*. 2005;53(9):1087-1097. doi:10.1369/jhc.5C6684.2005

Elayat AA, el-Naggar MM, Tahir M. An immunocytochemical and morphometric study of the rat pancreatic islets. *J Anat*. 1995;186(3):629-637.

Hao Y, Hao S, Andersen-Nissen E et al. Integrated analysis of multimodal single-cell data. bioRxiv. 2020. doi: 10.1101/2020.10.12.335331

Harvard Chan Bioinformatics Core. Single-cell RNA-seq analysis workshop. *GitHub Repository*. 2021. https://github.com/hbctraining/scRNA-seq

McInnes L, Healy J et al. UMAP: Uniform Approximation and Projection for Dimension Reduction. *Journal of Open Source Software*. 2018;3(29):861. doi: 10.21105/joss.00861

Patro R, Duggal G, Love MI, Irizarry RA, Kingsford C. Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods*. 2017;14(4):417-419. doi:10.1038/nmeth.4197

Rainer J, Gatto L, Weichenberger CX. ensembldb: an R package to create and use Ensembl-based annotation resources. *Bioinformatics*. 2019;35(17):3151-3153. doi:10.1093/bioinformatics/btz031

Settles M. Single Cell Transcriptomics. Lecture Presented at Genome Center University of California, Davis; June 2017.

Soneson C, Love MI and Robinson MD. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Research*. 2015, 4:152. doi:10.12688/f1000research.7563.1

Data Availability

All code used to generate the data in this report can be found at github.com/daisyhan97/bf528
The raw data referenced can be accessed on via GEO, Accession GSM2230758.

Software Documentation

Salmon – Version 1.1.0 of Salmon was used for this project. The data curator portion of this project making use of Salmon had a run-time of approximately 30 minutes.
Seurat – Version 4.0.1 of Seurat was used for this project. The programmer and analyst portions of this project making use of Seurat had a run-time of less than three minutes, running on eight cores on the BU SCC.

Supplementary Data

Supplementary Tables 1A and B – Summary statistics generated during the UMI counts matrix generation by Salmon Alevin, detailing the values regarding the (A) reads and UMIs, as well as the (B) cellular barcodes.

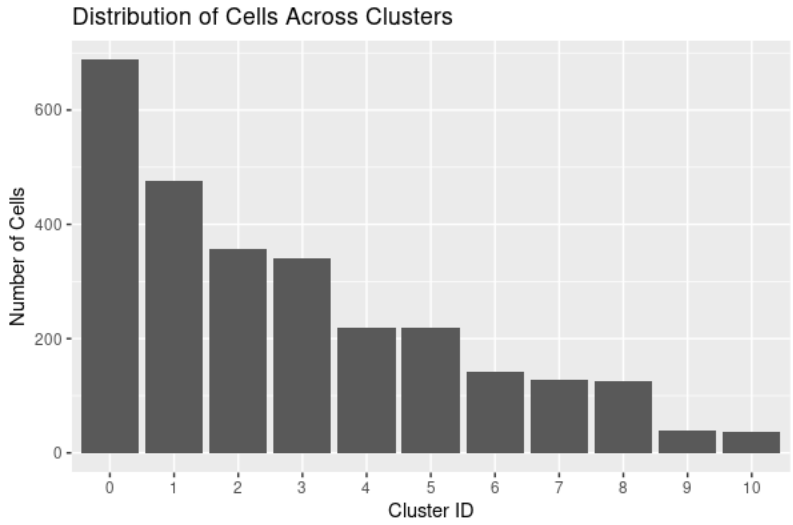
Supplementary Table 1A

Total Reads	Reads with "N"	Noisy UMI Reads	Deduplicated UMIs	Used Reads
1324837961	67930	39950	13393191	772508229

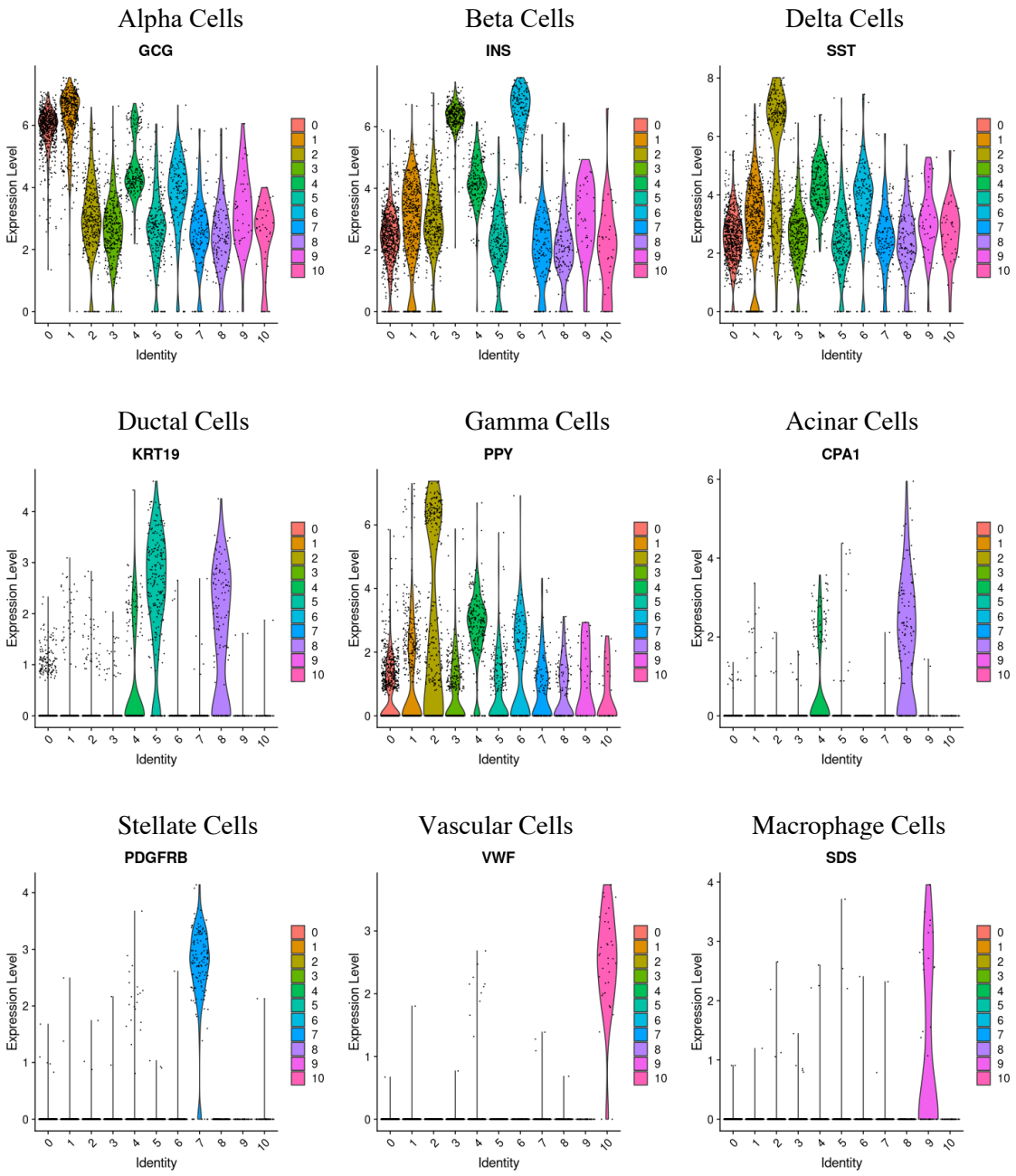
Supplementary Table 1B

Total Unique Barcodes	Total Whitelist Barcodes	Noisy Barcode Reads	Used Barcodes	Final Number Barcodes
4251176	4628	552181902	95949	4628

Supplementary Figure 1 - Distribution of Cells, Across Clusters. The number of cells in each cluster do not appear to be uniform across the clusters; Cluster 0 appears to have the most cells, while Cluster 10 has the least.



Supplementary Figure 2 – Violin plots of all marker genes provided by Baron et al., labelled by their corresponding cell type. Some of the marker genes appeared only in one or two clusters, allowing for simple cell type identification. Others appeared in multiple clusters, possibly allowing for error or misclassification. Marker genes that were not present in any of the cells by a significant amount are not represented by a violin plot.



Supplementary Figure 3 – Feature plots of all marker genes provided by Baron et al. Unlike the previous figure, all genes provided by Baron et al. are present in this figure, including those not expressed in significant levels.