ESTIMATING CELL-TYPE SPECIFIC GENE EXPRESSION IN INJURED MOUSE SPINAL CORD THROUGH DECONVOLUTION OF BULK RNA-SEQ DATA

By

DYLAN FORENZO

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Biomedical Engineering

Written under the direction of

Li Cai

And approved by

____

____

____

____

New Brunswick, New Jersey

May 2020

ABSTRACT OF THE THESIS

Estimating Cell-Type Specific Gene Expression in Mouse Spinal Cord Injury through Deconvolution of Bulk RNA-Seq Data

by DYLAN FORENZO

Thesis Director:

Li Ca

Advancements in single-cell RNA-Sequencing (scRNA-Seq) have allowed for the characterization of individual cell-type gene expression profiles. However, adult nerve cells exposed to a traumatic injury, such as cells in the spinal cord, are difficult to keep alive and viable for scRNA-Seq after isolation, making it difficult to study individual cell-type response to injury. Here, we use computational methods to deconvolve bulk RNA-Seq data obtained from mixtures of cells in the injured mouse spinal cord into individual cell types using healthy mouse scRNA-Seq Data. Through this deconvolution, we deduce that the mixtures mainly consist of neurons, oligodendrocytes, and astrocytes which make up approximately 54%, 24%, and 16% of the total cell population, respectively. These cell proportions and the differential gene expression between mixtures are then used to estimate the changes in cell-type specific gene expression between experimental conditions. The resulting gene expression profiles are then compared in a differential gene expression analysis (DGE) to provide evidence of the biological effects of gene therapies on neuron, oligodendrocyte, and microglia cell populations. Through the DGE analysis, we identified an average of 650 differentially expressed genes in neurons, 147 in oligodendrocytes, and 40 in microglia across experimental conditions. This approach provides an accessible and useful method for identifying the gene expression profiles of various cell-types after injury.

Acknowledgements

I would like to thank:

- Dr. Li Cai for his valuable advice, feedback, and guidance throughout my graduate education.
- Dr. Mark Pierce and Dr. David Shreiber for volunteering to be part of my graduate committee
- And my family for their unconditional support and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

RNA-Sequencing (RNA-Seq) is a method for quantifying gene expression levels in a sample of cells. These gene expression counts can be used to identify the biological pathways and functions taking place within the sample of cells. RNA-Seq has been shown to have a higher degree of accuracy and specificity than other quantitative gene expression assays, and continues to become more accessible over time.[1] This high-precision and accessibility have made RNA-Seq a powerful tool with applications in many different areas of biomedical research.

One major application of RNA-Seq data is differential gene expression analysis (DGE). This technique uses RNA-Seq to quantify gene expression in samples of cells under different experimental conditions, such as conditional knockout vs. wild type cells. The gene expression counts can then be compared between the samples to identify the biological pathways and functions that have different activity between the conditions, if any. Due to high variations in gene expression levels across a genome and inherent noise in the sequencing process, determining which genes are significantly differentially expressed requires statistical modelling techniques.[2] Several popular bioinformatics tools have been developed for this purpose such as DESeq2[3] and edgeR[4].

Single cell RNA-Sequencing (scRNA-Seq) is a relatively new technique that evolved from traditional bulk RNA-Seq. In this method, each cell in the sample is isolated and sequenced individually to provide quantitative gene expression information at the cellular level instead of at the sample level. This superior resolution of scRNA-Seq represents a major improvement over traditional RNA-Seq, however scRNA-Seq has its own drawbacks. Isolating cells for individual sequencing can be a difficult task, especially when studying certain types of cell populations and experimental conditions. Drop-seq is one popular procedure for isolation that encapsulates the cells into nanoliter droplets of oil and water.[5] Though Drop-Seq and similar techniques can make scRNA-Seq easier to perform, its relative inaccessibility remains a significant weakness compared to bulk RNA-Seq. Single nucleus RNA-Seq (snRNA-Seq) is a variant of scRNA-Seq where individual nuclei are sequenced instead of cells. This technique has been shown to produce similar results compared to scRNA-Seq while being easier to implement in some cases.[6] scRNA-Seq and snRNA-Seq are used interchangeably throughout this report.

In this study, we would like to determine the effects of two gene therapies on various cell types in mice after a spinal cord hemi-section injury (SCI). Since this goal requires cell-specific gene expression information, scRNA-Seq would be the preferred analysis method. However, spinal cord cells such as neurons are difficult to isolate for sequencing and existing techniques often result in lower cell yield and viability.[7] The SCI conditions also make scRNA-Seq difficult to implement as the injured cells are more susceptible to damage or cell death. Stress or cell death due to aggressive sorting methods can cause unintended gene expression responses in the cell samples compromising the gene expression data.[8] Since scRNA-Seq is not accessible for SCI cell-studies, an alternative method is needed to produce cell-type specific gene expression information.

One upcoming technique that addresses this need is bulk RNA-Seq deconvolution. In this procedure, scRNA-Seq is performed on a control group (ex. Healthy adult mice) and the resulting counts are then used to create a gene expression profile (GEP) for the cell types in the sample. This GEP is a model that contains information about which genes are expressed by a cell type and quantifies the gene expression levels under the control conditions. In bulk RNA-Seq deconvolution, these GEPs are used as references to estimate cell-type specific information from bulk RNA-Seq mixture samples. Several RNA-Seq deconvolution algorithms such as CIBERSORT[9] and MuSiC[10] have been developed that use scRNA-Seq GEPs to estimate the proportions of different cell populations in a bulk RNA-Seq mixture. These methods seek to model the bulk RNA-Seq samples as linear combinations of the cell-type GEPs that are present in the tissue of interest and compute the cell-type proportions as the linear coefficients. While these methods are useful for some applications, the models do not account for potential changes in cell-type GEPs across experimental conditions, such as those induced by SCI.

Recently, Newman et. Al., the group developing the CIBEROSRT algorithm, improved upon their original design and added a feature to estimate changes in GEPs across bulk RNA-Seq samples.[11] This new algorithm, CIBERSORTx, attempts to identify the changes in cell type GEPs that in gene expression due to experimental conditions among the bulk RNA-Seq samples. Using this novel method of bulk RNA-Seq deconvolution, quantitative cell-type specific gene expression information can be obtained without the need for scRNA-Seq of the SCI condition samples.

Here, we present an analysis pipeline for estimating cell-type specific gene expression in bulk RNA-Seq mixtures using healthy scRNA-Seq counts as reference. This method incorporates single-cell clustering, CIBERSORTx GEP estimation, and differential gene expression analysis to identify affected pathways and functions between experimental conditions. We then apply this pipeline to a mouse SCI dataset to identify DEGs in neuron, oligodendrocyte, and microglia populations between injured mice given SCI gene therapies and injured mice without treatment.

Chapter 2. Materials and Methods

2.1 Overview of Analysis Pipeline

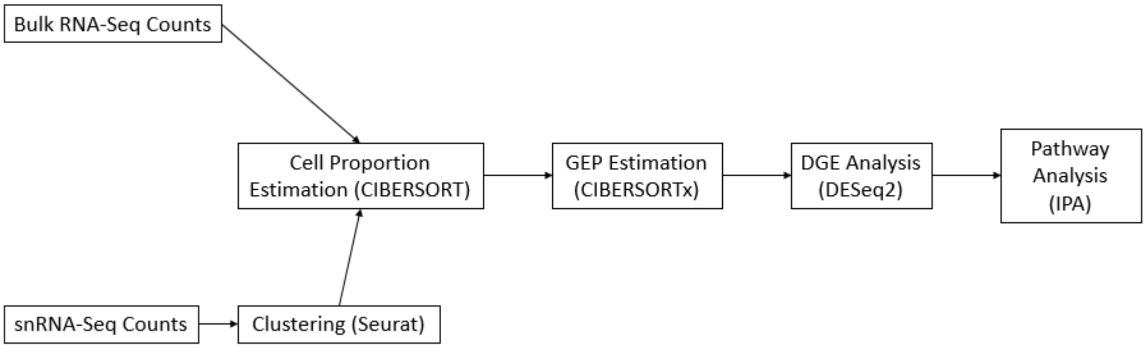


Figure 2.1.1 Flowchart of Analysis Pipeline

To produce cell-type GEPs for each injury condition, two sets of raw data are required: scRNA-Seq counts from a mixture of healthy mouse spinal cord cell populations and bulk RNA-Seq counts from mixtures of mouse spinal cord cell populations among each condition of interest. These raw data sets are displayed on the left-most side of the flowchart of Figure 2.1. The single-cell counts are first clustered using an unbiased clustering method and are assigned cell-type labels using known cell-type marker genes. Cell-type GEPs for healthy mice can then be inferred from these clusters by averaging gene expression across cluster members.

Next, these cell type GEPs are used in the CIBERSORT algorithm along with the bulk RNA-Seq mixtures to estimate the proportions of each cell population in each condition. This step of the analysis is shown in the second column of Figure 2.1 as the joining of two other blocks. The rest of the analysis follows a linear path.

The cell proportion estimates, single cell clusters, and bulk RNA-Seq mixtures are then used to estimate GEPs for each cell type in each SCI sample. Only genes that are significantly expressed and are found to have sufficient evidence of cell-type specific differential expression between experimental conditions are included in the estimated GEP for that cell type.

The estimated cell-type specific gene expressions can then be used in a DGE analysis to identify the significantly differentially expressed genes between experimental conditions. Lastly, a pathway analysis is performed on the genes found through DGE analysis and the biological pathways and functions that are affected by the experimental conditions are identified.

All statistical tests and computations were performed using the R software environment for statistical computing[12] unless noted otherwise. The various libraries and packages used for data analysis are listed throughout this chapter.

2.2 Bulk RNA-Seq Data

The motivation for this project was to apply the analysis pipeline outlined in Section 2.1 to a dataset of RNA-Seq counts generated from a mouse SCI and gene therapy study. This dataset consists of RNA-Seq counts of cells from the spinal cords of mice subjected to the following conditions: Sham (no injury), Control (hemi-section injury only), Treatment1 (hemi-section injury with gene therapy 1), and Treatment2 (hemi-section injury with gene therapy 2) taken at 3 and 35 days after a hemi-section injury as well as Control and Treatment1 taken 14 days after injury. Three replicates of each condition were sequenced except for Control and Treatment1 at 35 DPI which consisted of four replicates for a total of 32 samples. All RNA-Seq samples were used in the GEP estimation to improve statistical power, but only the Control, Treatment1, and Treatment2 samples at 3 and 35DPI were analyzed in the DGE and pathway analyis.

Condition	Number of Replicates
Sham 3 DPI	3
Control 3 DPI	3
Treatment1 3 DPI	3
Treatment2 3 DPI	3
Control 14 DPI	3

Treatment1 14 DPI	3
Sham 35 DPI	3
Control 35 DPI	4
Treatment1 35 DPI	4
Treatmen2 35 DPI	3
Total	32

Table 2.2.1 Bulk RNA-Seq Samples

2.3 Single Cell RNA-Seq Data

Single-nucleus RNA-Seq counts were obtained from the published mouse spinal cord atlas dataset by Sathyamurthy et. al.[13] This dataset is publicly available for academic use at the NCBI Gene Expression Omnibus[14] under GEO accession number GSE103892. The single-nucleus RNA-Seq counts present in this dataset were produced using a modified protocol of the Drop-Seq method[5] outlined in Sathyamurthy et. al. These counts are representative of a mixture of cell populations in a healthy adult mouse spinal cord. In the original study, several of the mice were subjected to acute pain to study potential effects on the expression of early-immediate genes. To be sure these effects were not present in this analysis, only the 6,750 nuclei from the control mice in the study were used out of the original 18,000 nuclei present in the dataset.

2.4 Clustering Analysis of scRNA-Seq Data

The remaining sequenced nuclei were clustered using the Seurat[15] package for R. The raw counts were first filtered to only pass through nuclei that expressed at least 200 unique genes and genes that were expressed by at least 3 unique nuclei. Next, the percentage of reads for each nucleus that corresponded to mitochondrial genes were computed. The remaining nuclei were then filtered again to only pass through if less than 20 percent of their reads mapped to mitochondrial genes and they expressed less than 5000 unique genes. Cells that express a high proportion of mitochondrial genes are considered to have compromised cell or nuclear membranes and are discarded from the analysis. Similarly, nuclei that express a very large number of genes are most likely doublets of nuclei captured in a single drop for sequencing and are also removed from the analysis. These preprocessing steps and cutoff values follow from the methods described in the original analysis of the snRNA-Seq data.[13] After this preprocessing, 6,556 nuclei were used for clustering. The counts from these nuclei are then normalized by the total expression per each nucleus, multiplied by 10,000, then log transformed.

Next, the top 2000 most variable genes across the nuclei were identified and used to perform a principal component analysis: a procedure that assigns orthogonal dimensions to the dataset. Jackstraw and Elbow plots were then constructed to determine the minimum number of principal components (PCs) that sufficiently describe the dataset. This number was chosen by looking for the flattening (elbow) of the curve between PCs on both plots as described by the Seurat tutorial.[15] The minimal number of PCs were then used to cluster the remaining nuclei using an unbiased clustering method based on the K nearest neighbors and Louvain[16] algorithms. The recommended resolution for clustering is between 0.4 and 1.2, with a higher resolution resulting in more clusters.[15] A resolution of 1.2 was chosen due to the large number of nuclei in the dataset and a larger number of desired clusters to identify small cell populations in the mixture. The resulting clusters were plotted in a two-dimensional plane using the UMAP dimension reduction algorithm.[17] The unbiased nature of this clustering means that Seurat clusters the nuclei without any cell phenotype data and relies only on the observed differences between the nuclei's expression. Therefore, the resulting clusters initially lack biological phenotypes and must be labelled as specific cell types using known marker genes.

Cluster labelling was performed using the Seurat package to report the descriptive genes for each resultant cluster and matching these genes to known cell-type marker genes. The descriptive genes for a cluster were found through Seurat using the Wilcoxon rank sum test[18] under the constraints that the gene was a positive marker for the cluster and was expressed by at least 50 percent of the nuclei in that cluster. These descriptive genes were then matched with cell-type marker genes for each major cell type outlined in the original mouse spinal cord atlas.[13] Marker genes for neurons, oligodendrocytes, astrocytes, vascular cells, meningeal cells, microglia, Schwann cells, and a mixture of precursor cells were taken from the original single nucleus RNA-Sequencing study and an online database of cell-type gene markers at CellMarker[19]. Each cluster was compared against marker genes from each cell-type of interest following equation 2.4.1 and the cell type with the highest percent match was taken to be the label for that cluster.

$$\%Match = 100 * \frac{Number\ of\ Cell\ Type\ Markers\ Also\ Present\ in\ the\ Descriptive\ Genes\ for\ that\ Cluster}{Number\ of\ Cell\ Type\ Markers}$$

Equation 2.4.1 Percent Match Calculation

For example, if there are 10 known Schwann cell marker genes and 2 of them are also descriptive genes for Cluster 1 then Cluster 1 has a 20% match with the Schwann cell-type:

$$\%Match = 100 * \frac{2}{10} = 20\%$$

Equation 2.4.2 Example Percent Match Calculation

Several clusters were found to have no significant matches to known cell types or to have a high percent of mitochondrial genes as descriptive genes. These clusters were assumed to not be representative of a cell phenotype and were removed from further analysis. The remaining nuclei were then labelled with their corresponding cell phenotypes and the proportions of each cell-type in the dataset are calculated. The average gene expression across nuclei in a cluster is interpreted as a GEP for the cell-type associated with that cluster.

2.5 Cell Proportion Estimation

Cell proportion and GEP estimations were performed using the CIBERSORT and CIBERSORTx web-based tools available online at cibersortx.stanford.edu.[11] To estimate cell proportions in the bulk RNA-Seq mixtures, the mixture samples and healthy mouse GEPs derived from single cell clustering are used in a linear system model. This model is shown in Equation 2.5.1 where G is an *i* by *j* matrix containing the GEPs of *i* genes in *j* cell types as column vectors. F is a *j* by *k* vector containing the *j* cell type proportions in the *k* sample mixtures provided by the bulk RNA-Seq data. B is an *i* by *k* matrix containing the bulk RNA-Seq counts of *i* genes for each mixture sample as *k* column vectors. The CIBERSORT algorithm seeks to solve for the cell fractions matrix F given the GEP and bulk data matrices G and B using a machine learning approach called nu-support vector regression[20].

$$\boldsymbol{G} * \boldsymbol{F} = \boldsymbol{B}$$

Equation 2.5.1 Cell Proportions Systems of Linear Equations Model

The bulk RNA-Seq counts described in Section 2.2 and the GEP matrix constructed in Section 2.4 were passed to CIBERSORT to impute the proportions of each cell-type in each spinal cord mixture. The quantile normalization option was disabled (as recommended for RNA-Sequencing data)[11] and 100 permutations were performed. These results were used both for quality control and downstream analysis.

2.6 Cell Type GEP Estimation

CIBERSORTx is an improvement upon the original CIBERSORT tool that allows for the estimation of GEPs in mixture samples in addition to cell-type proportion estimates. This method attempts to decompose the B matrix featured in Equation 2.5.1 into a modified G matrix for each cell type. These modified G matrices are the estimated GEPs for each cell type in each bulk mixture. This matrix decomposition is accomplished by identifying differences in the bulk sample gene expression levels and modifying the reference cell-type GEPs (G matrix) to account for these differences. The CIBERSORTx algorithm is described in detail under the Supplemental Information section of the original publication.[11]

At this time, the high-resolution tool used for GEP estimation is limited by CIBERSORTx to process only up to 1000 genes at a time due to high computational loads for the web-based application. To further reduce computation time and unnecessary complexity, genes in the bulk RNA-Seq samples with total counts of less than 0.5% of the number of clustered nuclei were filtered out. This number was chosen so that genes with very low expression would be excluded from further analysis, but significant genes expressed only by the smaller cell-type clusters would not be lost. Since the least common cell type was microglia with a proportion of 1%, 0.5% was chosen so that genes expressed in at least half of microglia cells would be included.

After filtering, 15,013 genes remained for analysis. To analyze the full transcriptome of the bulk RNA-Seq mixtures, the remaining genes were split into 16 groups based on alphabetical order with each group having a maximum of 1000 genes. The resulting GEPs were then concatenated to display the full transcriptome of each cell-type. The CIBERSORTx tool was run in high-resolution mode with the GEP matrix constructed in Section 2.4, and the bulk samples matrix developed in Section 2.2. No batch correction was enabled for any of the runs and quantile normalization was disabled as recommended for RNA-Seq data.[11]

2.7 Differential Gene Expression Analysis (DGE Analysis)

DGE analysis was performed using the DESeq2 R package.[3] Chosen cell-types of interest were analyzed individually, and comparisons were made between the gene therapies and control mice at 3 and 35 DPI. DGE results were obtained using a standard alpha value of 0.05, and only genes that were found to be significantly differentially expressed with a p-value of less than 0.05 were kept for pathway analysis. Each comparison between conditions was performed individually between two conditions at a time (i.e. Treatment1 at 35DPI vs. Control at 35 DPI). The resulting differentially expressed genes were written out as a table containing the expression, log2 fold change, and p-value for each remaining gene.

2.8 Ingenuity Pathway Analysis

Pathway analysis was performed using Qiagen's IPA application (QIAGEN Inc., https://www.qiagenbioinformatics.com/products/ingenuity-pathway-analysis). Each DGE contrast was uploaded and analyzed individually. The expression values, log2 fold change, and significance values were all used to generate an IPA dataset. Core analysis was run on each dataset individually with all included genes to identify the pathways and directionality associated with the differentially expressed genes found in Section 2.7.

Chapter 3. Results

3.1 Clustering Analysis of Single-Nucleus RNA-Seq Data

After the preprocessing described in Sections 2.3 and 2.4, 6,556 nuclei were analyzed for clustering. A principal component analysis (PCA) was run on the dataset to fit orthogonal dimensions to the nuclei. JackStraw and Elbow plots were generated for 30 PCs to visualize the significance of each PC in the dataset. These plots are shown in Figure 3.1.1 and Figure 3.1.2 respectively.

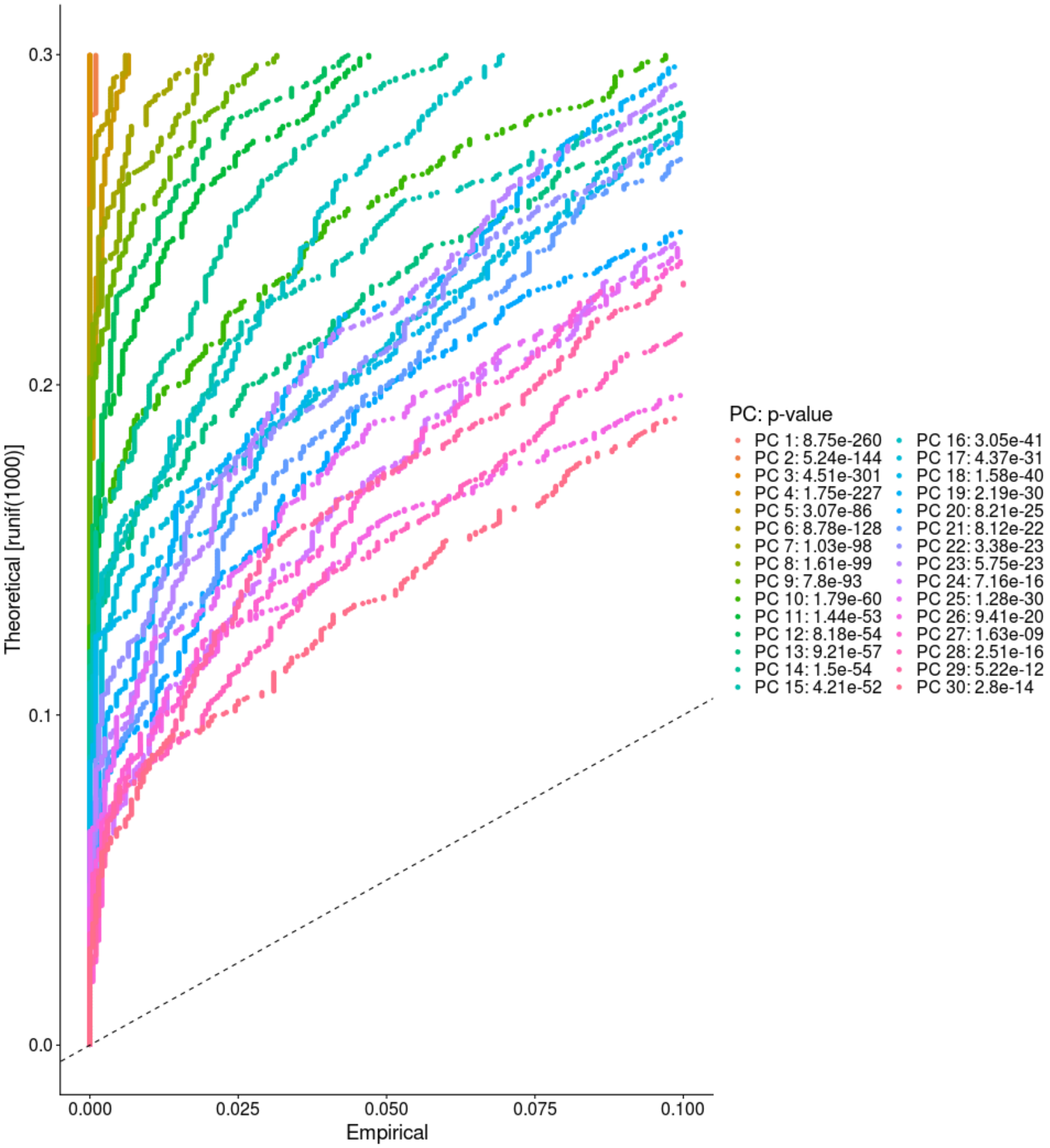


Figure 3.1.1 JackStraw Plot of Principal Components

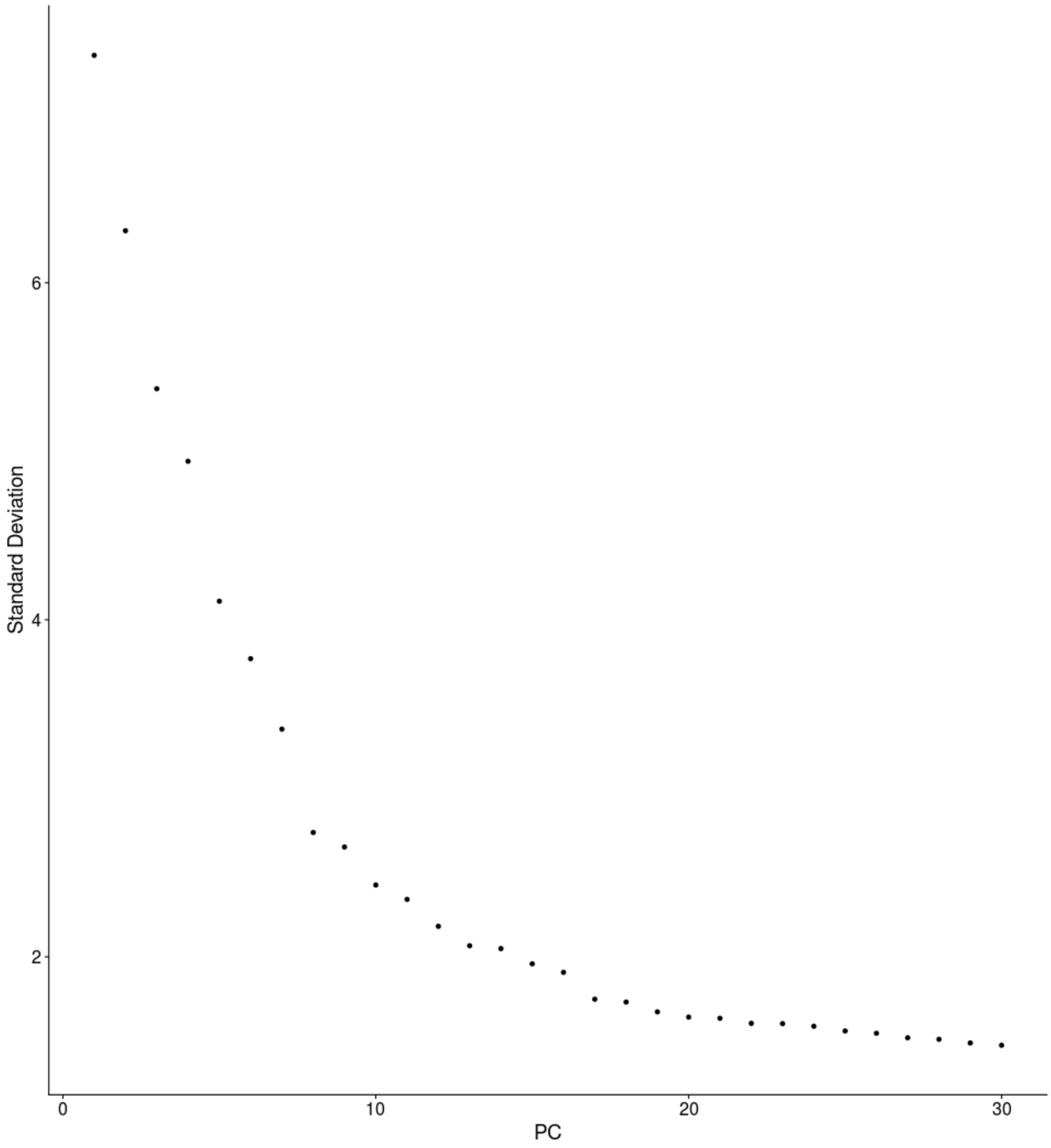


Figure 3.1.2 Elbow Plot of Principal Components

Both Figure 3.1.1 and Figure 3.1.2 plot the principal components of the single cell dataset (horizontal axis) vs. imputed statistical significance (vertical axis). The information in these plots is used to choose the number of minimum number of dimensions to describe the dataset in down-stream analysis. Using more dimensions could potentially improve clustering accuracy, but with diminishing returns and at the cost of computational burden. Here, the top 25 dimensions were chosen for further analysis as the 25th PC marks where the significance of the PC's start to level out. In the JackStraw plot, this point can be seen as the PCs after PC 25 have a much lower p-value than those before PC 25. The same point is also visualized in the Elbow Plot where PC 25 appears to be in the middle of the plateau.

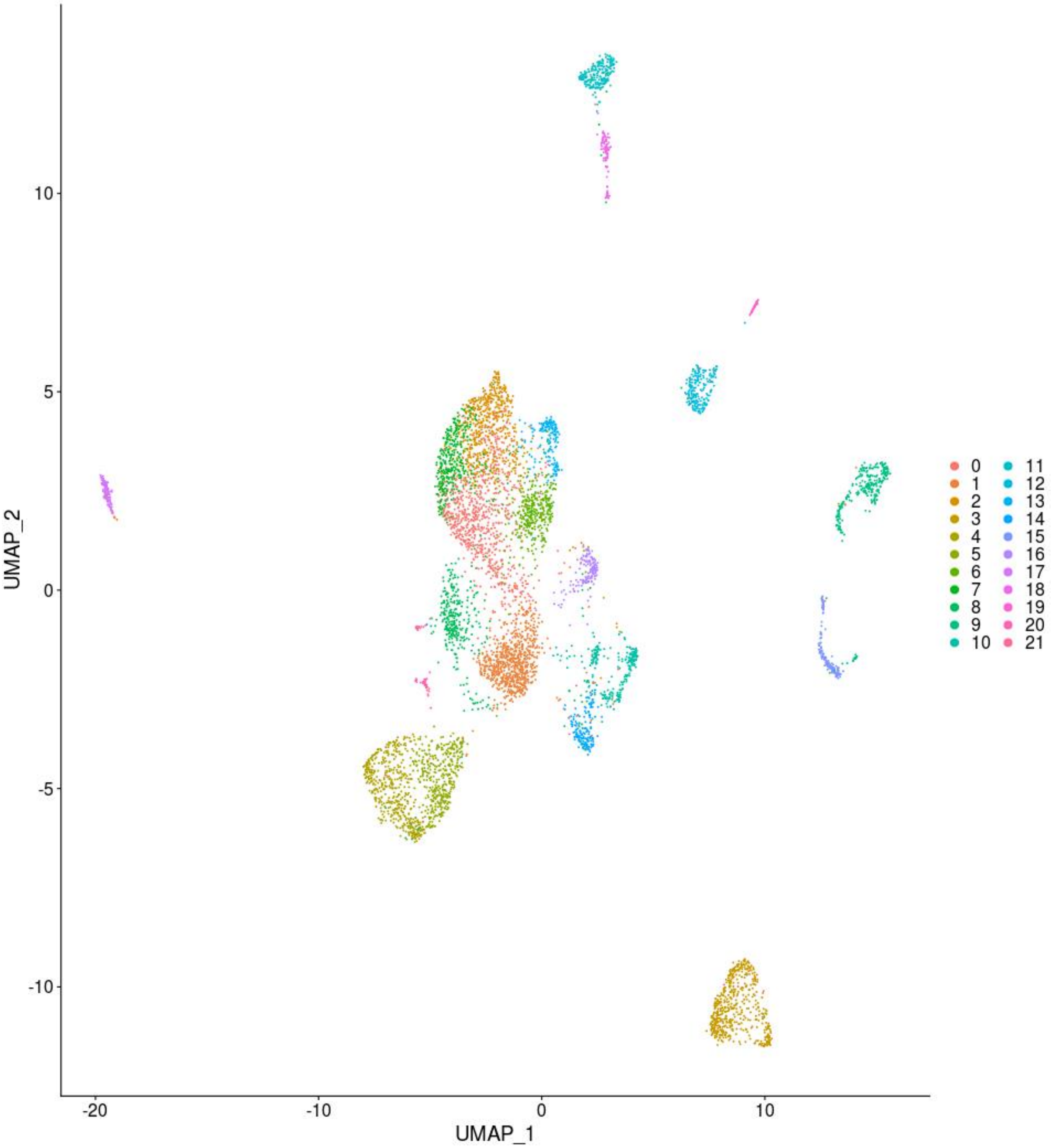


Figure 3.1.3 Raw Clusters in UMAP Plot

The Seurat clustering algorithm was run with the top 25 dimensions and a resolution of 1.2. The resulting 21 clusters are displayed in Figure 3.1.3 using the UMAP dimensional reduction. The clusters were then labelled using known cell-type gene markers as described in Section 2.4 (Clustering Analysis of scRNA-Seq Data). Figure 3.1.4 shows the percent matches between clusters and known cell-type gene markers.

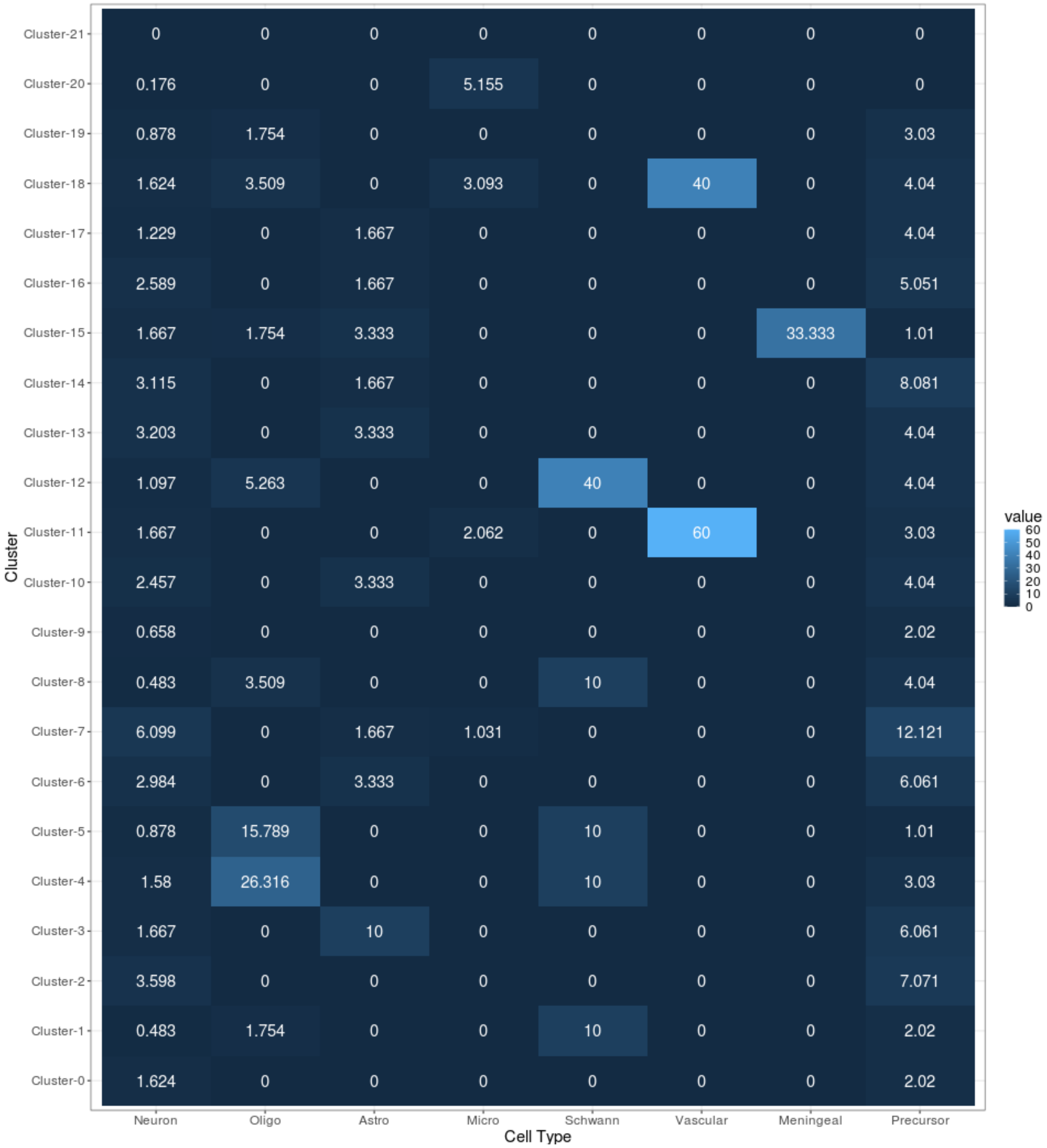


Figure 3.1.4 Raw Gene Marker Heatmap (values given as % of Matching Marker Genes)

There were many more known marker genes available for neuron and precursor cells, as these cell groups are well-studied and contain various known cell type sub-populations. The large number and variation of neuron and precursor marker genes resulted in at least several matching genes for nearly all clusters. To account for this, a cluster was labelled as a neuron population only if no other cell types were significantly matched to the cluster and a relatively large number of neuron marker genes were matched. Similarly, a cluster was labelled as a precursor cell population only if no other cell types were matched and the ratio between precursor and neuron matches was relatively large (defined as >2.5). In addition, cluster-9 was labelled as a group of meningeal cell-types due to its low matching with other cell types, proximity to an identified meningeal cluster on the UMAP plot, and the expression of some precursor cell markers which have been shown to be also be expressed in mammalian meningeal cells.[21] A table of the final identified cluster labels is shown in Table 3.1.2.

Cluster Number	**Cell-Type Label**
0	Neuron
1	Oligodendrocyte
2	Neuron
3	Astrocyte
4	Oligodendrocyte
5	Oligodendrocyte
6	Neuron
7	Neuron
8	Oligodendrocyte
9	Meningeal
10	Neuron
11	Vascular

12	Schwann
13	Neuron
14	Precursor
15	Meningeal
16	Neuron
17	Precursor
18	Schwann
19	Precursor
20	Microglia
21	N/A

Table 3.1.2 Cluster Cell-Type Labels

As shown in Figure 3.1.4 and Table 3.1.2, Cluster 21 had no matches to any of the known cell-type marker genes. Similarly, Cluster 19, although fitting the requirements to be labelled as a precursor group, had low matches to all known cell-type markers, including precursor cells. For these reasons, Clusters 19 and 21 were removed from further analysis. An updated map of all the original clusters with labels is shown in Figure 3.1.5. Here, clusters 19 and 21 are labelled “NA” and are shown in magenta.

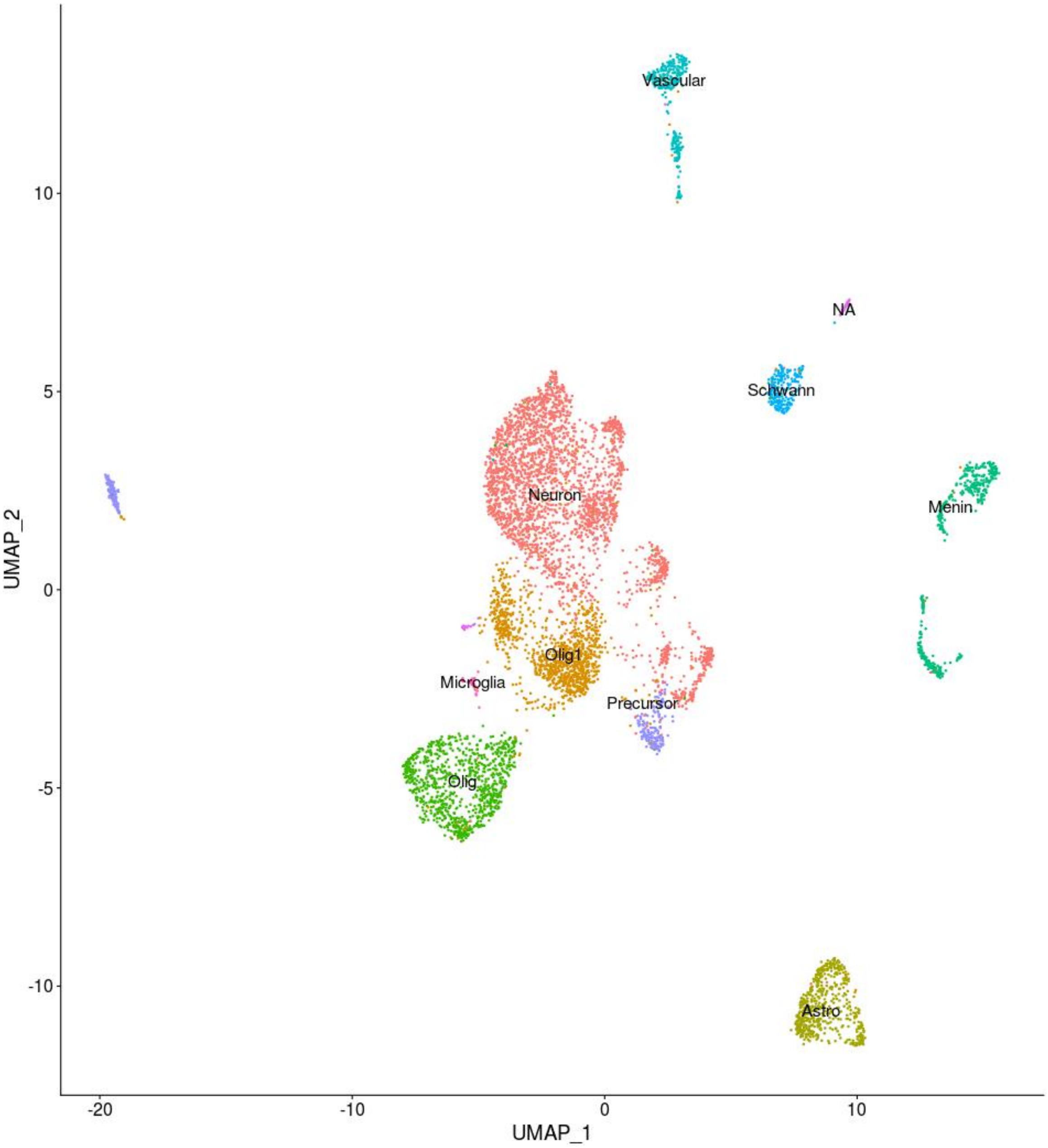


Figure 3.1.5 Labelled Clusters UMAP Plot

Further study of each cluster's descriptive genes showed that clusters 1 and 9 were characterized mostly by mitochondrial genes and genes highly expressed across all cell types. These clusters were originally labelled as "Oligo1" because they contained some of the oligodendrocyte marker genes but had different expression profiles from the other group of oligodendrocyte cells. Clusters 1 and 9 were removed from further analysis since the mitochondrial and universally expressed descriptive genes indicate clusters that do not accurately portray a biological phenotype. The criteria for removing clusters from further analysis was also taken from the methods of the original snRNA-Seq dataset study.[13] Figure 3.2.6 shows a map of the final cluster used in downstream analysis with Clusters 1, 9, 19, and 21 removed. Tables 3.1.3 and 3.1.4 show the proportions of each cluster and cell type in the dataset respectively.

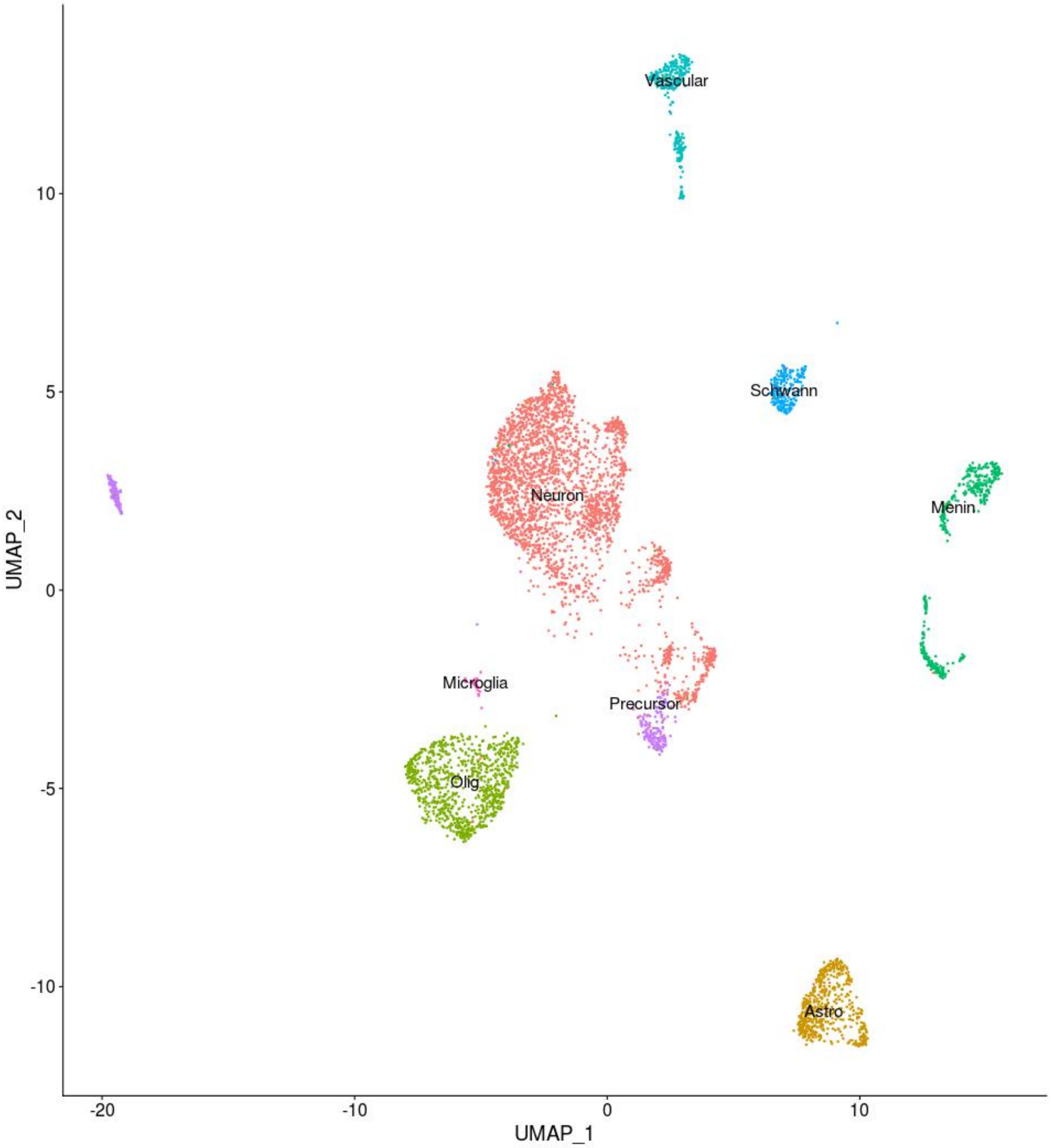


Figure 3.1.6 Final Clusters UMAP Plot

Cluster (Renumbered)	Percentage of Population	Cell Type
0	14.618	Neuron
1	11.024	Neuron
2	9.423	Astrocyte
3	9.199	Oligodendrocyte
4	7.11	Oligodendrocyte
5	6.87	Neuron
6	6.79	Neuron
7	5.400	Meningeal
8	4.749	Neuron
9	4.060	Vascular
10	3.799	Schwann
11	3.147	Neuron
12	3.110	Precursor
13	2.868	Meningeal
14	2.775	Neuron
15	2.402	Precursor
16	1.862	Vascular
17	0.782	Microglia

Table 3.1.3 Proportions of Each Final Cluster in snRNA-Seq Dataset

Cell Type	Percentage of Population

Neuron	49.981
Oligodendrocyte	16.313
Astrocyte	9.423
Meningeal	8.268
Vascular	5.922
Precursor	5.512
Schwann	3.799
Microglia	0.782

Table 3.1.4 Proportions of each Cell Type in Single Nucleus Dataset

3.2 Estimation of Cell Type Proportions in Bulk RNA-Seq Data

Sample	**Neuron**	**Precursor**	**Menin**	**Oligo**	**Schwann**	**Vasc**	**Astro**	**Microglia**
ShamT3.1	51.5	0	0.6	24.3	3.2	4.6	15.4	0.3
ShamT3.2	55.2	0	0.1	23.2	2.1	4.2	15	0.2
ShamT3.3	55.8	0	0.6	23.1	1.8	4.5	13.8	0.4
ShamT35.1	52.5	0	0	26.1	0.2	3.3	17.6	0.3
ShamT35.2	54.2	0	0	25	0	3	17.5	0.3
ShamT35.3	57.2	0	0	23.8	0	2.5	16.4	0.2

Table 3.2.1 Sham Cell Type Proportion Estimates (Percent of Sample)

The cell-type GEPs produced from snRNA-Seq clustering were then fed analyzed using CIBERSORT to estimate the cell type proportions in each bulk RNA-Seq mixture as discussed in Section 2.5. Table 3.2.1 shows the estimated cell type proportions as the percent of cells that belong to that cell group out of the entire sample. Each row represents a Sham bulk RNA-Seq mixture and each column describes a single cell type.

3.3 Bulk Sample GEP Estimation

Next, CIBERSORTx was used to estimate the gene expression profiles of each cell type among each mixture sample. The CIBERSORTx high resolution tool used for this step is currently only able to run up to 1000 genes at a time. To comply with this requirement, the genes present in the bulk mixtures were split alphabetically into 16 groups of genes so that groups 1-15 contained 1000 genes each and group 16 contained the remaining 14 genes. The results of each run of CIBERSORTx produced a 1000 row by 32 column matrix for each cell type (14x32 for group 16) where each entry I,J contains the estimated expression of gene I in sample J. Table 3.3.1 shows a cropped portion of the resulting matrix for gene group 5 expressed by neurons to illustrate the format.

Gene Symbol	ShamT3	ShamT3.1	ShamT3.2
Ganc	945.405	945.405	945.405
Gm26899	1.459	1.459	1.459
Gatm	13534.573	13534.573	13534.573
Gm14004	53.065	35.898	47.138
Galk2	631.896	663.281	810.669

Table 3.3.1 Example Portion of Estimated Neuron Gene Expression Matrix

Figure 3.3.1 provides another visualization of estimated GEP matrices. Here, the estimated expression of group 5 genes by neurons as a heatmap where blue indicates less expression, black neutral expression, and yellow greater expression. The rows containing all black entries correspond to genes that were either found to similar expression levels across all the samples or did not have enough statistical power to be estimated reliably. This lack of statistical power could be either that the gene was not sufficiently expressed by that cell type in the single-cell derived GEP, or that the gene was not differentially expressed among the bulk RNA-Seq mixtures.

All 16 expression matrices for each cell-type were concatenated into one cumulative table containing all the gene expression information for that cell-type. These concatenated cell type tables have the same format as a bulk RNA-Seq dataset but are composed of cell-type specific information. This format allows for DGE analysis to be performed in the same procedure as in a traditional bulk RNA-Seq study.

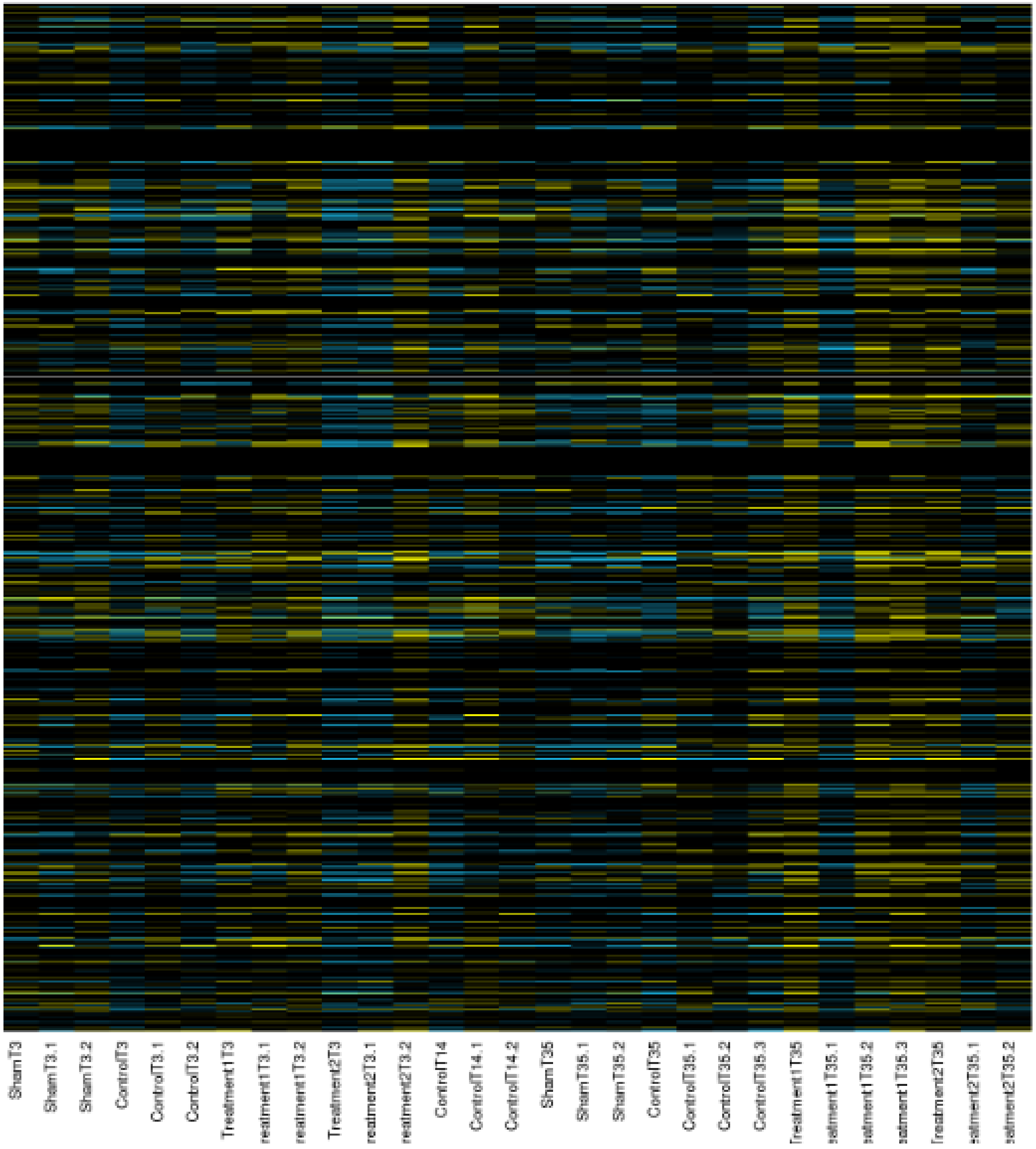


Figure 3.3.1 Heatmap of the Expression of 1000 genes in Neurons

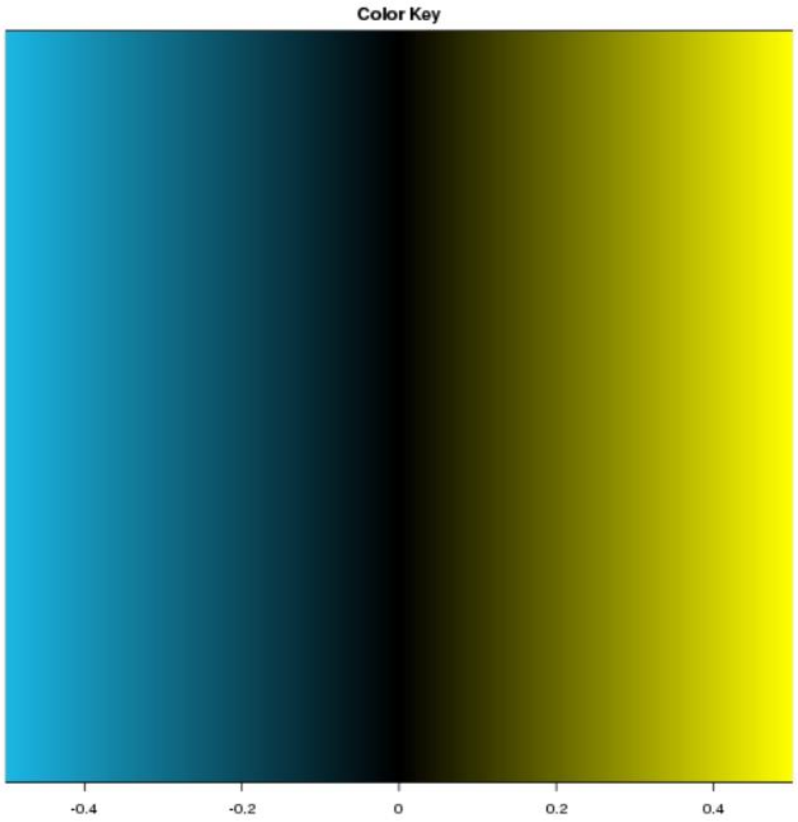


Figure 3.3.2 Color Legend for Figure 3.3.1. Values Given as Log2 of Fold Change

3.4 Differential Gene Expression and Pathway Analysis

To determine which genes were significantly differentially expressed between samples, we used DESeq2 to analyze the estimated gene expression counts in a DGE analysis. Six of the bulk mixture samples were analyzed in four contrasts which are shown in the first column of Table 3.4.1. The rest of Table 3.4.1 shows the number of DEGs found in each cell type for each contrast. Only neurons, oligodendrocytes, and microglia were found to have significantly DEGs among the estimated gene counts.

Contrast	Neuron	Oligodendrocyte	Microglia
Treatment1 vs Control at 3DPI	679	70	14

Treatment2 vs Control at 3DPI	544	69	68
Treatment1 vs Control at 35DPI	694	337	43
Treatment2 vs Control at 35DPI	682	113	33

Table 3.4.1 Number of DEGs by Cell Type

Next, each set of genes was analyzed using Qiagen's Ingenuity Pathway Analysis (IPA) software. This software matches a list of genes to related biological pathways and functions using a proprietary knowledge base. When also given the corresponding gene expression fold changes, IPA can report whether the pathway or function was upregulated or downregulated in the contrast. Selected canonical pathways and biological functions found by IPA are shown for neurons, oligodendrocytes, and microglia in Tables 3.4.2, 3.4.3, and 3.4.4 respectively. In these tables, a (+) indicates the pathway or function was upregulated or expressed more by the first sample in the contrast, while a (-) indicates the function was downregulated or expressed more by the second sample in the contrast. If genes related to the pathway or function appear in both samples of a contrast, that pathway was considered neutrally related and is not labelled with either a (+) or (-).

Contrast	**Canonical Pathways**	**Top Biological Functions**
Treatment1 vs Control 3 DPI	Sirtuin Signaling Oxidative Phosphorylation Sumolyation Pathway	Viral Infection (+) Astrocytosis (-) Gliosis (-)
Treatment2 vs Control 3 DPI	Synaptogenesis Signaling (-) Neuregulin Signaling (+) Neuroinflammation (+)	Nervous System Development (-) Degeneration of Neurons (+)
Treatment1 vs Control 35 DPI	Synaptogenesis Signaling (+) GABA Receptor Signaling Nestin Signaling (+) Glutamate Signaling (+)	Nervous System Development (+) Degeneration of Neurons (-)
Treatment2 vs Control 35 DPI	Synaptogenesis Signaling (+) Endocannabinoid Synapse (+)	Motor Dysfunction (-) Degeneration of Neurons (-)

	GABA Receptor Signaling Melatonin Signaling (-)	Nervous System Development (-)

Table 3.4.2 Selected Neuron Pathways and Biological Functions

Comparison	**Canonical Pathways**	**Top Biological Functions**
Treatment1 vs Control 3 DPI	P70s6k Signaling (-) Neuroinflammation (-) Neuropathic Pain Signaling (-)	Damage of Nervous System (-) Molecular Transport (+)
Treatment2 vs Control 3 DPI	nNos Signaling Endocytosis Signaling	Nervous System Development (-)
Treatment1 vs Control 35 DPI	Gai Signaling (+) Melatonin Signaling (+) Neuropathic Pain Signaling (+) Endocannabinoid Synapse (-)	Nervous System Development (+) Cell-To-Cell Signaling (+)
Treatment2 vs Control 35 DPI	Neuropathic Pain Signaling (-) Endocytosis Signaling Synaptogenesis Signaling (+)	Nervous System Development (+) Neurodegeneration (+)

Table 3.4.3 Selected Oligodendrocyte Pathways and Biological Functions

Comparison	**Canonical Pathways**	**Top Biological Functions**
Treatment1 vs Control 3 DPI	Adenine and Adenosine Salvage P53 Signaling	Inflammation (-)
Treatment2 vs Control 3 DPI	Leukocyte Signaling (-) Phagosome Maturation IL-8 Signaling (-)	Inflammation (-) Immune Response (-) Infection of Cells (+/-)
Treatment1 vs Control 35 DPI	Lymphocyte Apoptosis Th17 Activation	Immune Cell Migation (+) Inflammation (-)
Treatment2 vs Control 35 DPI	IL-8 Signaling	Immune Cell Proliferation (-) Inflammation (+/-)

Table 3.4.4 Selected Microglia Pathways and Biological Functions

Chapter 4. Discussion

The statistical power available using the CIBERSORTx high resolution tool is related to both the number of samples and cell types being analyzed. A higher number of samples increases the accuracy and precision of the tool, while trying to deconvolve the bulk mixtures into more cell types lowers the statistical power. Since this analysis was limited by the maximum number of samples available for analysis, we grouped the finely differentiated clusters into more generalized cell-types during clustering in order to analyze less groups. This step allowed us to analyze the main spinal cord cell types (neurons, oligodendrocytes, etc.) with the available statistical power provided by our dataset. Analyzing more clusters, such as including cell sub-populations, is possible with this analysis pipeline, but is limited by the available number of bulk RNA-Seq samples.

When estimating the cell type proportions in the bulk mixtures, only the Sham sample results were analyzed for quality control. The injury and treatment conditions were expected to cause differences in gene expression among the cell-type populations, leading to changes in the respective GEPs. After these changes, the injury condition GEPs would be incompatible with the reference GEPs generated by single-cell clustering, which could lead to unreliable cell-type proportion estimates, hence the need for the high-resolution tool. Conversely, since the Sham condition samples and single-cell dataset were both taken from healthy adult mice, CIBERSORT should be able to estimate the cell-type proportions reliably.

The Sham sample proportion estimates shown in Table 3.3.1 are similar to the known cell type proportions found in the single-cell dataset for neurons, vascular cells and microglia shown in Table 3.2.3. The precursor cell group was found to not be significantly present in the estimated cell-type proportions. Since this generalized group is composed of a wide mixture of cell-type sub-populations, the resulting GEP may not accurately represent a uniform cell phenotype. This could explain one potential reason for the reported absence of precursor cells in the estimated cell type proportions.

Both Schwann and meningeal cells were also underreported in the cell type proportions estimates compared to the known proportions of cell types from clustering. Schwann cells have important functions in the peripheral nervous system and are not usually found in the adult spinal cord outside of injury.[22] Therefore, one explanation for the presence of Schwann cells in the single-cell dataset could be differences in harvesting procedures between the single-cell sequencing and SCI studies. Since meningeal cells make up the outer layer of the spinal cord, this same explanation could also account for the differences in meningeal cell proportions.

Estimated proportions of oligodendrocyte and astrocytes populations were larger than the known single-cell clustering proportions. This indicates either that differences in cell harvesting procedures resulted in more glial cells in the SCI dataset, or that the gene expression from precursor, Schwann, and meningeal cells was attributed to glial cells instead during the proportion estimation step.

After GEPs were estimated for each cell-type in each bulk mixture sample, only neurons, oligodendrocytes, Schwann cells, meningeal cells, and microglia were found to have differentially expressed genes between conditions. Among these cell populations, only neurons, oligodendrocytes, and microglia were considered relevant to the SCI treatment study and analyzed further. The lack of differentially expressed genes among cell types such as astrocytes implies that the CIBERSORTx algorithm was not able to attribute the changes in gene expression between conditions to that cell type. This evidence primarily supports these cells did not have significant changes in gene expression between conditions. However, another explanation for the lack of differentially expressed genes for some cell types could be that the “true” cell-type GEP in the injury condition is different enough from the reference GEP that CIBERSORTx was not able to attribute gene expression changes across the bulk samples to these cells. This case would cause some cell types such as astrocytes to have been reported with no differentially expressed genes when there may in fact have been large changes in gene expression after injury.

When analyzing the identified canonical pathways and functions, it should be noted that although the IPA knowledge base contains many relevant pathways and gene functions, it is not an exhaustive source of information on genes. In addition, much of the information available through IPA is based on user studies and submitted datasets. This could potentially create a bias where more of the results are linked to areas where RNA-Seq is commonly employed and reported, such as cancer research. Only the pathways and functions found that are relevant to SCI are analyzed in this report. The full IPA results are available upon request from the Cai Lab.

Among the biological pathways and functions associated with the neuron DEGs, nervous system development and signaling were the primary results. As shown by Table 3.4.2, Treatment 1 did not show much effect on neural signaling at day 3 but produced upregulation of several types of signaling and nervous system development at day 35. Treatment 2 caused an upregulation of inflammation signaling and downregulation of synapse formation at day 3 implying that the treatment may have undesirable effects early on. However, by day 35 Treatment 2 also showed evidence of increased nervous system development and signaling. This implies that both treatments have beneficial effects on nervous system rehabilitation after a few weeks.

Many of the oligodendrocyte DEGs were linked to the same pathways as the neuron DEGs, as shown in Table 3.4.3. The downregulation of neuroinflammation and pain signaling by Treatment 1 at 3 days implies that the treatment may being to have beneficial effects early on. Treatment 2 shows a downregulation of nervous system development at day 3 which is consistent with the results from the neuron DEGs. Both treatments have mixed signaling results among oligodendrocytes at day 35 but produce an upregulation of nervous system development.

The microglia DEGs in Table 3.4.4 show a downregulation of inflammation for both Treatments at days 3 and 35. Treatment 2 also produced a downregulation of immune response at day 3 and a downregulation of immune cell proliferation of day 35. The lowered immune response and inflammation provide evidence of beneficial effects of the treatments after SCI.

We have demonstrated an analysis pipeline for estimating cell-type specific gene expression in bulk RNA-Seq mixtures using scRNA-Seq data as a reference. This pipeline was then applied to our mouse SCI dataset to discern differentially expressed genes and their related biological pathways and functions for neurons, oligodendrocytes, and microglia. We believe the analysis presented in this paper has a broader application as a guide for estimating the cell-type DEGs and related pathways in any bulk RNA-Seq mixtures where scRNA-Seq references can be made available.

Appendices

Appendix A. R Scripts

Please note that these scripts contain hard coded file names and paths. To run the scripts successfully, each file path and/or name must be edited in the script to match the corresponding file location on the local machine.

Script 1: Single cell clustering and GEP formation

```
#This script is to run the Seurat algorithm to cluster the single cell RNA-Seq data from
Sathyamurthy et. Al.


#Libraries
library(dplyr)
library(Seurat)
library(ggplot2)
library(reshape2)

##Load raw scRNA-Seq counts
rawCounts<-
read.table("../rawData/GSE103892_Expression_Count_Matrix.txt",header=T,row.names=1,str
ingsAsFactors=F,sep="\t")

##Create Seurat Object
main<-CreateSeuratObject(counts=rawCounts, project = "sci", min.cells=3, min.features=
200)

##Quality Control
main[["percent.mt"]] <- PercentageFeatureSet(main, pattern = "^mt-")
#VlnPlot(main, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
main<-subset(main, subset = nFeature_RNA < 5000 & percent.mt <= 20)
#Keep only control nuclei
controlNames<-c('f1','F3','F4','m1','M4','M5','facx','fbcx','Macx')
main<-subset(main, subset = orig.ident %in% controlNames)

##Normalize using "LogNormalie"
main <- NormalizeData(main, normaliztion.method ="LogNormalize", scale.factor = 10000)

##Calculate Highly Variable Features
main <- FindVariableFeatures(main, selection.method = "vst", nfeatures = 2000)

##Scale Data
allGenes <- rownames(main)
main <-ScaleData(main, features = allGenes)

##Run PCA dimensional reduction
```

```
main <- RunPCA(main, features = VariableFeatures(object = main))
#Use the following for visualization of dimensions
#print(main[["pca"]], dims = 1:5, nfeatures = 5)
#VizDimLoadings(main, dims = 1:2, reduction = "pca")
#DimPlot(main, reduction = "pca")
#DimHeatmap(main, dims = 1, cells = 500, balanced = TRUE)

##Determine number of significant PCs
#Jaskstraw plot
main <-JackStraw(main, num.replicate = 100,dims=30)
main <- ScoreJackStraw(main, dims = 1:30)
JackStrawPlot(main, dims = 1:30)
ElbowPlot(main,ndims=30,reduction = "pca")

##Cluster cells (use 20 PCs) (resolution for 3k cells ~ 0.4:1.2)
main <- FindNeighbors(main, dims = 1:25)
main <- FindClusters(main, resolution = 1.2)

##Plot Clusters using UMAP
main <- RunUMAP(main, dims = 1:25)
DimPlot(main, reduction = "umap")

##Get Cluser Proportions
clusterProp <-
data.frame(matrix(0,nrow(unique(main[["seurat_clusters"]]))),stringsAsFactors=F)
rownames(clusterProp)<-seq(1:nrow(clusterProp))
for (i in seq(1:nrow(clusterProp))) {
clusterProp[i,1]<-100*length(which(main[["seurat_clusters"]] == (i-1)))/ncol(main)
rownames(clusterProp)[i]<-paste0("Cluster-",as.character(i-1))
}

##Marker Genes for each Cluster
markers<-list()
for (i in seq(1:nrow(clusterProp))) {
markers[[i]] <- FindMarkers(main, ident.1 = i-1, min.pct = 0.50, thresh.test = 0.5, only.pos=T)
}

#Label clusters (manual)
newClusterID<-
c("Neuron","Olig1","Neuron","Astro","Olig","Olig","Neuron","Neuron","Olig1","Menin","Neur
on","Vascular","Schwann","Neuron","Precursor","Menin","Neuron","Precursor","Vascular","N
A","Microglia","NA")
names(newClusterID)<-levels(main)
main<-RenameIdents(main,newClusterID)
DimPlot(main, reduction = "umap", label = TRUE, pt.size = 0.5) + NoLegend()
clusterProp[,2]<-newClusterID
colnames(clusterProp)<-c("Percentage","Label")
```

```
#Read in cell-type markers
neuroMarks<-
read.table("../markers/neuronMarkers.csv",sep=",",stringsAsFactors=F,row.names=1,header=
T)
oligMarks<-
read.table("../markers/OligMarkers.csv",sep=",",stringsAsFactors=F,row.names=1,header=T)
astroMarks<-
read.table("../markers/astroMarkers.csv",sep=",",stringsAsFactors=F,row.names=1,header=T)
microMarks<-
read.table("../markers/microMarkers.csv",sep=",",stringsAsFactors=F,row.names=1,header=T
)
schwannMarks<-
read.table("../markers/schwannMarkers.csv",sep=",",stringsAsFactors=F,row.names=1,heade
r=T)
vascMarks<-
read.table("../markers/vascMarkers.csv",sep=",",stringsAsFactors=F,row.names=1,header=T)
meninMarks<-
read.table("../markers/meninMarkers.csv",sep=",",stringsAsFactors=F,row.names=1,header=
T)
opcMarks<-
read.table("../markers/opcMarkers.csv",sep=",",stringsAsFactors=F,row.names=1,header=T)
preMarks<-
read.table("../markers/preMarkers.csv",sep=",",stringsAsFactors=F,row.names=1,header=T)

#Marker Proportions
fProp<-data.frame()
for (i in 1:length(unique(main@meta.data[["seurat_clusters"]]))) {
fProp[i,1]<-100*length(which(rownames(neuroMarks) %in% rownames(markers[[i]])))
/nrow(neuroMarks)
fProp[i,2]<-100*length(which(rownames(oligMarks) %in%
rownames(markers[[i]])))/nrow(oligMarks)
fProp[i,3]<-100*length(which(rownames(astroMarks) %in%
rownames(markers[[i]])))/nrow(astroMarks)
fProp[i,4]<-100*length(which(rownames(microMarks) %in%
rownames(markers[[i]])))/nrow(microMarks)
fProp[i,5]<-100*length(which(rownames(schwannMarks) %in%
rownames(markers[[i]])))/nrow(schwannMarks)
fProp[i,6]<-100*length(which(rownames(vascMarks) %in%
rownames(markers[[i]])))/nrow(vascMarks)
fProp[i,7]<-100*length(which(rownames(meninMarks) %in%
rownames(markers[[i]])))/nrow(meninMarks)
fProp[i,8]<-100*length(which(rownames(preMarks) %in%
rownames(markers[[i]])))/nrow(preMarks)
}
colnames(fProp)<-
c("Neuron","Oligo","Astro","Micro","Schwann","Vascular","Meningeal","Precursor")
```

```
#Marker heatmap
all<-fProp
all$Cluster<-rownames(clusterProp)
allMelt<-melt(all,variable.name='Cell')
allMelt$Cluster<-factor(allMelt$Cluters,levels=rownames(clusterProp)
ggplot(data=allMelt,aes(x=Cell,y=Cluster,fill=value)) + geom_tile() +
geom_text(aes(label=round(value,digits=3)),color='white') +theme_bw() + labs(x=”Cell Type”)

h<-fProp[,1:7]
for (i in 1:nrow(h)) {
for (j in 1:7) {
if(fProp[i,j]==max(fProp[i,1:7])) {
h[i,j]=100
}
}}
image(t(as.matrix(h)),xlab=colnames(fProp),ylab=rownames(clusterProp))

#Cut bad clusters
trim<-subset(main, subset = seurat_clusters %in% c(0,2:7,9:18,20))
DimPlot(trim, reduction = "umap",label=TRUE, pt.size = 0.5) + NoLegend()

#Relevel cluster numbers
for (i in 0:(max(as.numeric(trim[["seurat_clusters"]][,1]))-1)) {
if (i %in% 2:7) {
trim[["seurat_clusters"]][which(trim[["seurat_clusters"]][,1]==i),1]<-(i-1)
} else if (i %in% 9:18) {
trim[["seurat_clusters"]][which(trim[["seurat_clusters"]][,1]==i),1]<-(i-2)
} else if (i==20) {
trim[["seurat_clusters"]][which(trim[["seurat_clusters"]][,1]==i),1]<-(i-3)
}
}
trimProp <- data.frame(matrix(0,nrow(unique(trim[["seurat_clusters"]]))),stringsAsFactors=F)
rownames(trimProp)<-seq(1:nrow(trimProp))
for (i in seq(1:nrow(trimProp))) {
trimProp[i,1]<-100*length(which(trim[["seurat_clusters"]] == (i-1)))/ncol(trim)
rownames(trimProp)[i]<-paste0("Cluster-",as.character(i-1))
}
trimClusterID<-
c("Neuron","Neuron","Astro","Olig","Olig","Neuron","Neuron","Menin","Neuron","Vascular","
Schwann","Neuron","Precursor","Menin","Neuron","Precursor","Vascular","Microglia")
trimProp[,2]<-trimClusterID
colnames(trimProp)<-c("Percentage","Cell Type")

#Extract single cell counts data
```

```
scData<-data.frame(rownames(trim),GetAssayData(object = trim, slot =
'counts'),stringsAsFactors=F)
trimNames<-data.frame(matrix(0,(nrow(trim@meta.data)+1)),stringsAsFactors=F)
trimNames[1,1]<-"GeneNames"
for (i in 1:nrow(trim@meta.data)) {
trimNames[(i+1),1]<-trimClusterID[as.numeric(trim[["seurat_clusters"]][i,1])]
}
```

Script 2: Cut bulk RNA-Seq samples to match the genes in the single cell dataset

```
#This script is to cut the bulk SCI RNA-Seq counts to match the genes in the finished single cell
matrix

#Libraries

#Load in single cell matrix and bulk counts
sc<-read.table("../singleCellMatrix.txt",sep="\t",header=T,stringsAsFactors=F)
bulk<-read.table("../rawData/bulkCounts.txt",sep="\t",header=T,stringsAsFactors=F)
colnames(bulk)[1]<-"GeneNames"

#Match the gene names and cut bulk Counts
bulkCut<-bulk[which(bulk[,1] %in% sc[,1]),]

#Write out
write.table(bulkCut,file="../results/bulkCounts.txt",row.names=F,quote=F,sep="\t")
```

Script 3: Separate the GEP genes into groups of 1000 or less for CIBERSORTx high resolution

```
#This script is for selecting up to 1000 genes to run the CIBERSORTx high resolution mode

#Libraries

#Load in single cell data
sc<-read.table("../singleCellMatrix.txt",row.names=1,header=T,stringsAsFactors=F,sep="\t")

#Cut genes that are counted less than 0.5% of number of nuclei (27 counts)
geneCount<-rowSums(sc)
scCut<-sc[which(geneCount>=27),]

#Split genes into groups of 1000
numGroups<-ceiling(nrow(scCut)/1000)
for (i in 1:(numGroups-1)) {
toDo<-rownames(scCut)[(1000*(i-1)):(1000*i)]
write.table(toDo,file=paste0("../results/geneGroups/group",as.character(1000*(i-
1)),".txt"),row.names=F,col.names=F,sep="\t",quote=F)
}
toDo<-rownames(scCut)[(1000*(numGroups-1)):nrow(scCut)]
```

```
write.table(toDo,file=paste0("../results/geneGroups/group",as.character(1000*(numGroups-
1)),".txt"),row.names=F,col.names=F,sep="\t",quote=F)
```

Script 4: Concatenate the CIBERSORTx output tables

```
#This script is to concatenate all of the cibersort cell type tables into one master table for
each cell type

#Master list
results<-list()

#Loop through each cell type
for (cellType in c("Neuron", "Olig", "Vascular", "Astro", "Precursor", "Schwann", "Menin",
"Microglia")) {
index<-length(results)+1

##Load each gene group table
#Initialize Dataframe
first<-
read.table(paste0("~/dtf32/seurat/results/cibersortx/group0/",list.files(path=paste0("~/dtf32
/seurat/results/cibersortx/group0/"),pattern=paste0(cellType,"_Window15.txt"))),header=T,s
ep="\t",stringsAsFactors=F)
results[[index]]<-first
names(results)[index]<-cellType
#Loop through other groups
for (group in as.character(seq(1000,15000,1000))) {
toDo<-
read.table(paste0("~/dtf32/seurat/results/cibersortx/group",group,"/",list.files(path=paste0("
~/dtf32/seurat/results/cibersortx/group",group,"/"),pattern=paste0(cellType,"_Window15.txt
"))),header=T,sep="\t",stringsAsFactors=F)
results[[index]]<-rbind(results[[index]],toDo)
}
}

#Remove NA rows
resultsCut<-list()
for (i in seq(1:length(results))) {
resultsCut[[i]]<-results[[i]][which(is.finite(results[[i]][,2]) & results[[i]][,2] != 1),]
resultsCut[[i]]<-resultsCut[[i]][!duplicated(resultsCut[[i]][,1]),]
}
names(resultsCut)<-names(results)

save.image("../varsR/concatenateCibersortTables")
geps<-resultsCut
save("geps",file="../varsR/geps")
```

Script 5: Differential Gene Expression Analysis

```
This script is to run DGE analysis on the GEP produced by cibersort

#Libraries
library(DESeq2)

#Load in GEPs
load("../varsR/geps") #Loaded in as: geps

#Isolate cell type data: Neuron
cellType<-"Neuron" #Change this to string of cell type name ex: cellType <-"Neuron"
index<-which(names(geps)==cellType)
cts<-ceiling(geps[[index]][,-1])
rownames(cts)<-geps[[index]][,1]

#DESeq dataset
Design<-data.frame(row.names = colnames(cts), condition = gsub(pattern="\\.[0-
9]","",colnames(cts)))
DESeq.ds<-DESeqDataSetFromMatrix(countData = cts, colData = Design, design = ~ condition,
tidy = FALSE)

dds<- DESeq(DESeq.ds)

#Compare Results
#Treatment 1
c3t1Results<-results(dds, independentFiltering = TRUE, alpha = 0.05, contrast = c("condition",
"Treatment1T3","ControlT3"))

filteredNA<-c3t1Results[complete.cases(c3t1Results[,5]),]
filtered<-filteredNA[filteredNA[,5]<=0.05,]
sortedFC<-filtered[order(filtered$log2FoldChange,decreasing=TRUE),]

write.table(sortedFC,
file=paste0("../results/DESeq/",cellType,"/c3t1.txt"),sep="\t",row.names=T,col.names=NA,quo
te=F)

c35t1Results<-results(dds, independentFiltering = TRUE, alpha = 0.05, contrast =
c("condition", "Treatment1T35","ControlT35"))

filteredNA<-c35t1Results[complete.cases(c35t1Results[,5]),]
filtered<-filteredNA[filteredNA[,5]<=0.05,]
sortedFC<-filtered[order(filtered$log2FoldChange,decreasing=TRUE),]

write.table(sortedFC,
file=paste0("../results/DESeq/",cellType,"/c35t1.txt"),sep="\t",row.names=T,col.names=NA,qu
ote=F)
```

```
#Treatment 2
c3t2Results<-results(dds, independentFiltering = TRUE, alpha = 0.05, contrast = c("condition",
"Treatment2T3","ControlT3"))

filteredNA<-c3t2Results[complete.cases(c3t2Results[,5]),]
filtered<-filteredNA[filteredNA[,5]<=0.05,]
sortedFC<-filtered[order(filtered$log2FoldChange,decreasing=TRUE),]

write.table(sortedFC,
file=paste0("../results/DESeq/",cellType,"/c3t2.txt"),sep="\t",row.names=T,col.names=NA,quo
te=F)

c35t2Results<-results(dds, independentFiltering = TRUE, alpha = 0.05, contrast =
c("condition", "Treatment2T35","ControlT35"))

filteredNA<-c35t2Results[complete.cases(c35t2Results[,5]),]
filtered<-filteredNA[filteredNA[,5]<=0.05,]
sortedFC<-filtered[order(filtered$log2FoldChange,decreasing=TRUE),]

write.table(sortedFC,
file=paste0("../results/DESeq/",cellType,"/c35t2.txt"),sep="\t",row.names=T,col.names=NA,qu
ote=F)
```

Appendix B. Top Differentially Expressed Genes

The following tables contain the differentially expressed genes found through DGE analysis. Where more than 20 genes were found to be differentially expressed in a contrast, only the top 20 most differentially expressed genes (in either direction) are shown. The full lists of genes are available upon request from the Cai Lab. In the tables, a positive fold change indicates higher expression in the preceding sample, and a negative fold change indicates higher expression among the latter sample.

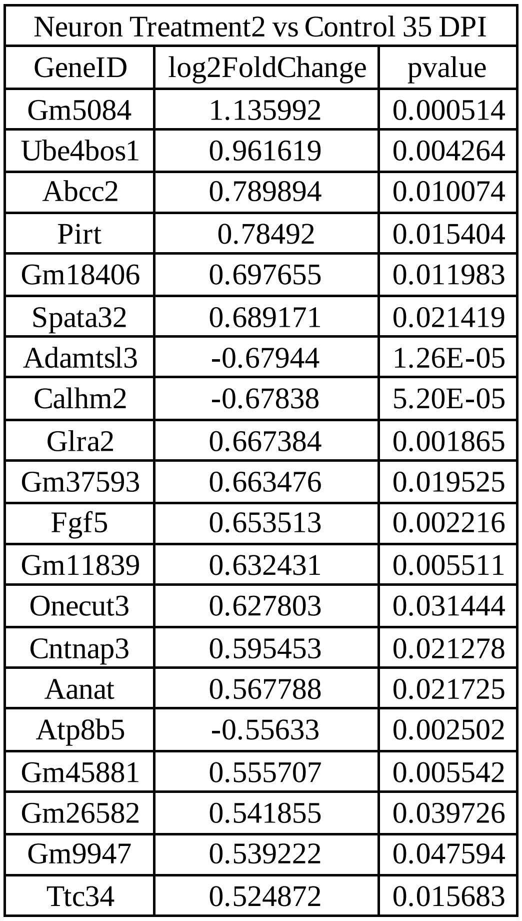
Neuron Treatment 1 vs Control 3DPI		
GeneID	log2FoldChange	pvalue
Nek11	0.894665	0.000474
Gm44618	0.7942	0.033498
Tmc1	0.7468	0.044726
Igsf9	0.720185	0.003211
Arrdc2	0.701915	0.017479
Scn7a	-0.70165	0.031447
Fgd2	0.662474	0.005648
Pipox	0.616289	0.001199
Mri1	0.604942	0.008821
Ccdc146	0.602459	0.036859

Rubcnl	0.588275	0.006133
Gm30340	0.586487	0.032727
Rab26os	0.583781	0.020101
Gm20751	0.573152	0.010425
Dusp12	0.554898	0.016009
Spa17	0.519206	0.029648
Atp8b5	0.508598	0.008874
Lrrc23	0.508293	0.009372
Gm29480	0.507135	0.013197
Gm20457	0.505578	0.002475

Neuron Treatment 2 vs Control 3DPI		
GeneID	log2FoldChange	pvalue
Scn7a	-1.0488	0.001356
Slc2a5	0.947609	0.005064
Pzp	-0.88677	0.02042
Lrrc43	-0.87633	0.022168
Arrdc2	0.860157	0.003575
Mri1	0.842108	0.00026
Mfsd7a	0.802249	0.00099
Nmnat3	0.799916	0.004501
Hpgds	0.72902	0.011916
Tigd2	0.719878	0.000121
Gm16201	0.705069	0.014125
Sema5a	-0.70337	0.000203
Fgd2	0.691645	0.003854
Tmco4	0.666988	0.00179
Cfap54	-0.65967	0.018335
Dcdc5	-0.64495	0.017252
Ephx1	0.634312	0.000599
Trim71	-0.61196	0.034123
Calhm2	0.600861	0.000692
Adamts2	-0.60058	0.009157

Neuron Treatment1 vs Control 35DPI		
GeneID	log2FoldChange	pvalue
Il1rapl2	1.265442	0.002502
AU023762	0.849022	0.006797
Ube4bos1	0.841985	0.007932
Slc26a4	0.834547	0.020324

Gm44618	0.754096	0.017681
Scn4b	0.696987	0.015588
Gm14342	0.667785	0.036371
Gm45470	0.656535	3.49E-05
Aanat	0.620659	0.006726
Gm30003	0.611846	0.046491
Adam21	0.591897	0.000219
A830073O21Rik	0.570991	0.001085
Acp7	0.56916	0.02261
Vsnl1	0.548759	0.002157
Psg16	0.546818	0.003296
Cdh12	0.537286	0.039777
Fgf5	0.530894	0.008166
Tacr3	0.527774	0.000412
Gm14164	0.524307	0.006722
Gm2824	0.518113	0.002759



Neuron Treatment2 vs Control 35 DPI		
GeneID	log2FoldChange	pvalue
Gm5084	1.135992	0.000514
Ube4bos1	0.961619	0.004264
Abcc2	0.789894	0.010074
Pirt	0.78492	0.015404
Gm18406	0.697655	0.011983
Spata32	0.689171	0.021419
Adamtsl3	-0.67944	1.26E-05
Calhm2	-0.67838	5.20E-05
Glra2	0.667384	0.001865
Gm37593	0.663476	0.019525
Fgf5	0.653513	0.002216
Gm11839	0.632431	0.005511
Onecut3	0.627803	0.031444
Cntnap3	0.595453	0.021278
Aanat	0.567788	0.021725
Atp8b5	-0.55633	0.002502
Gm45881	0.555707	0.005542
Gm26582	0.541855	0.039726
Gm9947	0.539222	0.047594
Ttc34	0.524872	0.015683

Oligodendroctye Treatment 1 vs Control 3DPI		
GeneID	log2FoldChange	pvalue
Dolk	-3.15764	0.000127
Gm34821	-2.26303	0.020099
P2rx3	1.757429	0.048984
Il4	-1.20051	0.000461
Spsb4	-1.18319	0.011789
Nbas	1.139813	0.045237
Pcsk6	-1.09406	0.041594
Map3k21	0.965991	0.025454
Hlcs	0.875986	0.040629
Gramd1b	0.844254	0.017872
Zfp994	-0.83137	0.002073
Pik3c3	-0.76605	0.023653
Enpp2	-0.70643	0.014049
Vgll3	-0.69583	0.019444
Mgat4c	0.654473	0.040948
Zswim1	0.617203	0.015098
Azin1	-0.5731	0.047093
Pomk	-0.57015	0.014575
Gnb5	0.527452	0.042454
Abce1	-0.5055	0.040811

Oligodendrocyte Treatment2 vs Control 3DPI		
GeneID	log2FoldChange	pvalue
Tcte2	-2.86027	0.037261
P2rx3	2.050625	0.021388
Rint1	-1.14499	0.037551
Hsph1	-0.99891	0.008046
Gm14403	0.869692	0.030952
Etnppl	0.758278	0.017169
Vgll3	-0.67855	0.022649
Alox8	0.642263	0.045043
Cntn4	-0.60837	0.011059
Cdh10	0.578933	0.027189
Npr3	-0.55336	0.016403
Zfp994	-0.54036	0.04372
Hrh1	-0.53153	0.035781
Gm15991	0.51929	0.010437
Gm816	-0.4622	0.018195

Dgke	-0.4516	0.019744
Cadps	-0.44118	0.032492
Sv2c	0.438789	0.027705
Prss12	0.40658	0.037003
Sacs	-0.39594	0.037848

Oligodendrocyte Treatment1 vs Control 35DPI		
GeneID	log2FoldChange	pvalue
Tmem245	1.331267	0.031131
Cntn6	-1.30252	0.007048
Il1rapl2	-1.16033	0.001774
Isca1	1.141669	0.005442
Disp3	1.138967	0.034337
Itfg1	1.101933	0.011085
Map3k21	1.089851	0.003246
Dync2h1	1.053654	0.014184
Pax8	-1.05009	0.018792
Tshz3	1.014598	0.016627
Tmem229b	-0.99297	0.008694
Nap1l5	-0.97177	0.006892
Xrcc2	-0.94864	0.015474
Mest	-0.92291	0.012592
Zfp354c	0.909058	0.010417
Dlg3	0.896972	0.00787
Cdh12	-0.89158	0.01466
Nkiras1	-0.89102	0.000253
Podxl2	-0.87789	0.029713
Pcmtd1	0.874745	0.025557

Oligodendrocyte Treatment2 vs Control 35DPI		
GeneID	log2FoldChange	pvalue
Glra2	-1.53782	0.004725
Susd2	-1.30809	0.038271
Pax8	-1.14069	0.018223
Isca1	0.890487	0.044729
Slc7a14	-0.85384	0.028521
Pafah1b2	0.840196	0.003834
Kcnip4	-0.77719	0.000403
Pdha1	0.76448	0.012864

Grm7	-0.72264	0.019488
Mum1l1	0.700355	0.001034
Grm3	-0.6883	0.039155
Prss12	-0.66947	0.000279
Nefh	-0.66862	0.006544
Usp22	0.655989	0.038979
Hgsnat	-0.65455	0.010751
Slc27a2	-0.6241	0.000192
Plekha6	0.606053	0.045269
Cygb	-0.59967	0.022427
Zc2hc1a	0.594027	0.018922
Carmil2	-0.5853	0.03789

Microglia Treatment1 vs Control 3DPI		
GeneID	log2FoldChange	pvalue
Hdac1	0.286477	8.78E-05
Ncapg2	0.15881	0.039637
Selplg	-0.85205	0.043079
Ikzf1	-0.43719	0.004605
Rack1	-0.40433	0.003914
Eef1a1	-0.37114	0.001807
Apbb1ip	-0.34366	0.02393
Fmnl3	-0.30741	0.011745
Coro1a	-0.29832	0.016813
Aprt	-0.27238	0.031718
Npc2	-0.26866	0.026254
Prdx6	-0.25843	0.021235
Gadd45g	-0.22192	0.036959
Plekha2	-0.17599	0.048038

Microglia Treatment2 vs Control 3DPI		
GeneID	log2FoldChange	pvalue
Selplg	-1.2575	0.002831
Tspan4	-0.95956	0.000708
Abca1	-0.93595	0.014636
Rnf128	0.893164	0.025694
Cald1	-0.86722	0.024698
Stom	-0.82764	0.00808
Lima1	-0.78801	0.007874
Nrp1	-0.70874	0.031455

Haus8	0.686692	0.028359
Nfatc1	-0.65546	0.009583
Antxr2	0.601351	0.038766
Pik3cg	-0.60011	0.043272
Cotl1	-0.53127	0.019836
Ccnd3	-0.53057	0.000181
Stab1	0.525443	0.045011
Rack1	-0.51697	0.000226
Heatr1	0.431744	0.022475
Ptpn1	-0.42891	0.02805
Sdcbp	0.427876	0.048353
Ikzf1	-0.42479	0.005902

Microglia Treatment1 vs Control 35DPI		
GeneID	log2FoldChange	pvalue
Kntc1	0.799499	0.048651
Rnf128	0.781142	0.024354
Abca1	0.772069	0.020053
Banf1	-0.72129	0.025613
Stab1	-0.66603	0.00335
Parpbp	0.655236	0.008306
Gpr171	-0.40353	0.008659
Sqor	0.399088	0.000533
Tbxas1	0.385499	0.040584
Tmem51	0.362405	0.006605
Mdfic	0.346339	0.017789
Hacd4	0.33431	0.027931
Ecm1	-0.30158	0.000201
Rbp1	0.297111	0.003164
Runx1	0.2903	0.022347
Ttc7	0.275763	0.000681
Rack1	0.275566	0.023181
Irf5	0.264094	0.030646
Cmtm3	0.257758	0.037728
Fmnl3	0.2452	0.020145

Microglia Treatment2 vs Control 35DPI		
GeneID	log2FoldChange	pvalue
Thbs1	-1.54824	0.000755

E2f7	-0.7805	0.001924
Cd93	-0.74905	0.011233
Adam8	-0.69812	0.011485
Stab1	-0.66647	0.006571
Parpbp	0.664121	0.013204
Spp1	-0.66062	0.020248
Nrp1	0.637514	0.038539
Tbxas1	0.510697	0.012012
Pus7l	0.388115	0.024911
Ccnd3	0.370619	0.005124
Sqor	0.362329	0.003596
Ptbp3	0.357058	0.042785
Ddx11	0.32057	0.012577
Hpgd	-0.28374	0.028135
Irak4	0.275311	0.013095
Npc2	0.271122	0.016499
Creg1	0.244437	0.020004
Il4ra	-0.23768	0.02761
Rfc4	0.231363	0.011055

References

1. Wang Z, Gerstein M, Snyder M. RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet*. 2009;10(1):57-63. doi:10.1038/nrg2484

2. Anders S, Huber W. Differential expression analysis for sequence count data. *Nat Preced*. April 2010:1-1. doi:10.1038/npre.2010.4282.2

3. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol*. 2014;15(12):550-550. doi:10.1186/s13059-014-0550-8

4. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*. 2010;26(1):139-140. doi:10.1093/bioinformatics/btp616

5. Macosko EZ, Basu A, Satija R, et al. Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell*. 2015;161(5):1202-1214. doi:10.1016/j.cell.2015.05.002

6. Wu H, Kirita Y, Donnelly EL, Humphreys BD. Advantages of Single-Nucleus over Single-Cell RNA Sequencing of Adult Kidney: Rare Cell Types and Novel Cell States Revealed in Fibrosis. *J Am Soc Nephrol JASN*. 2019;30(1):23-32. doi:10.1681/ASN.2018090912

7. Eldeiry M, Yamanaka K, Reece TB, Aftab M. Spinal Cord Neurons Isolation and Culture from Neonatal Mice. *J Vis Exp JoVE*. 2017;(125). doi:10.3791/55856

8. Liu S, Trapnell C. Single-cell transcriptome sequencing: recent advances and remaining challenges. *F1000Research*. 2016;5:182. doi:10.12688/f1000research.7223.1

9. Newman AM, Liu CL, Green MR, et al. Robust enumeration of cell subsets from tissue expression profiles. *Nat Methods*. 2015;12(5):453-457. doi:10.1038/nmeth.3337

10. Wang X, Park J, Susztak K, Zhang NR, Li M. Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nat Commun*. 2019;10(1):1-9. doi:10.1038/s41467-018-08023-x

11. Newman AM, Steen CB, Liu CL, et al. Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nat Biotechnol*. 2019;37(7):773-782. doi:10.1038/s41587-019-0114-2

12. R Core Team. R: A language and environment for statistical computing. *R Found Stat Comput*. http://www.R-project.org/.

13. Sathyamurthy A, Johnson KR, Matson KJE, et al. Massively Parallel Single Nucleus Transcriptional Profiling Defines Spinal Cord Neurons and Their Activity during Behavior. *Cell Rep*. 2018;22(8):2216-2225. doi:10.1016/j.celrep.2018.02.003

14. Edgar R, Domrachev M, Lash A. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res*. 2002;30(1):207-210.

15. Stuart T, Butler A, Hoffman P, et al. Comprehensive Integration of Single-Cell Data. *Cell*. 2019;177(7):1888-1902.e21. doi:10.1016/j.cell.2019.05.031

16. Blondel, Vincent D, Guillaume, Jean-Loup Blondel, Vincent D, Lambiotte, Renaud, Lefebvre, Etienne. Fast unfolding of communities in large networks. *J Stat Mech Theory Exp*. 2008;2008(10). doi:doi:10.1088/1742-5468/2008/10/P10008

17. Becht E, McInnes L, Healy J, et al. Dimensionality reduction for visualizing single-cell data using UMAP. *Nat Biotechnol*. 2019;37(1):38-44. doi:10.1038/nbt.4314

18. Wilcoxon F. Individual Comparisons by Ranking Methods. *Biom Bull*. 1945;1(6):80-83. doi:10.2307/3001968

19. Zhang X, Lan Y, Xu J, et al. CellMarker: a manually curated resource of cell markers in human and mouse. *Nucleic Acids Res*. 2019;47(D1):D721-D728. doi:10.1093/nar/gky900

20. Scholkopf, Bernhard, Smola, Alex J., Williamson, Robert C., Bartlett, Peter L. New Support Vector Algorithms. *Neural Comput*. 2000;12(5):1207-1245.

21. Bifari F, Berton V, Pino A, et al. Meninges harbor cells expressing neural precursor markers during development and adulthood. *Front Cell Neurosci*. 2015;9. doi:10.3389/fncel.2015.00383

22. Zhang S, Huang F, Gates M, Holmberg EG. Role of endogenous Schwann cells in tissue repair after spinal cord injury. *Neural Regen Res*. 2013;8(2):177-185. doi:10.3969/j.issn.1673-5374.2013.02.011