Deconvoluting cell types through scRNA-Seq

Erica Chio

May 2020

Abstract

This paper details how to determine what cell types are present in Rattus norvegicus superior cervical ganglion (SCG) cultures and how these cell types respond to different drug treatments (LY, mirin) in terms of gene expression. The SCG culture was analyzed to determine it consisted of neurons, glial, endothelial, and Schwann cells. The cultures were then separated into its respective cell type, and gene expression was analyzed to determine what genes were up and down regulated with treatment. The change in gene expression from the drug treatment affected certain pathways, that would in turn affect cell survival rate.

1 Introduction

The superior cervical ganglion is a part of the autonomic nervous system, and is the largest ganglion of the cervical sympathetic trunk. The superior cervical ganglion is composed of neurons and receive presynaptic signals from the sympathetic trunk.[1;2] The superior cervical ganglia is the mid point between the sympathethic trunk and targets in the head and neck for the presynaptic signals.[2]

The Mre11-Rad50-Nbs1 (MRN) complex is vital in sensing DNA double-strand breaks, promoting repair, maintaining genome stability during replication, activating ATM, a protein kinase critical in maintaining genome integrity.[3;4] MRN recruits ATM to the damaged DNA, and ATM phosphorylates substrates to coordinate cell cycle arrest and DNA repair.[5] Mirin is a inhibitor of MRN, and prevents ATM activation, inhibiting the MRN-ATM pathway. Without ATM activated, there is no response to double-stranded breaks and no homology-directed repair.[3]

Another pathway, PI3K/AKT is associated with numerous cellular functions. These functions include cell metabolism, growth, proliferation, and survival.[6] LY294002 (LY) is an inhibitor on the PI3K/AKT pathway, by specifically inhibiting PI3K. LY was found to successfully enhance sensitivity of cancer cells to drug-related apoptosis, because of its ability to inhibit the PI3K/AKT pathway.[7]

The data are single cell RNA-seq data of four Rattus norvegicus SCG cell cultures with two treated with DMSO, one treated with mirin, and one treated with LY. The single cell RNA-seq data came out in fastq files which had to be aligned against the Rattus norvegicus genome to generate counts. From there, these aligned counts could be analyzed to determine cell clusters, differentially expressed genes, and impacted pathways.

2 Results

To determine cell types within the cell culture, Seurat was employed for its graph-based clustering approach. The cultures was first integrated. Integrating the different cultures allowed identification of shared cell states across the different data sets. First, anchors had to be identified. Anchors are pairwise correspondence between cells in different data sets.[8] The clusters' cell types were determined through their top conserved markers.

For example in Figure 1, S100 and SOX10 are known cell markers for Schwann cells. As, S100 and SOX10 were in the top 5 cell markers of cluster 9, cluster 9 can be defined as a Schwann cell cluster. Of the 17 different clusters, there was eight neuron, five glial, three endothelial, and two Schwann clusters. Gilal cells are support cells for neurons, and Schwann cells are a type of glial cells. The neuron and glial cells are clustered closely together, while the endothelial cells seem to be clustered on its own.

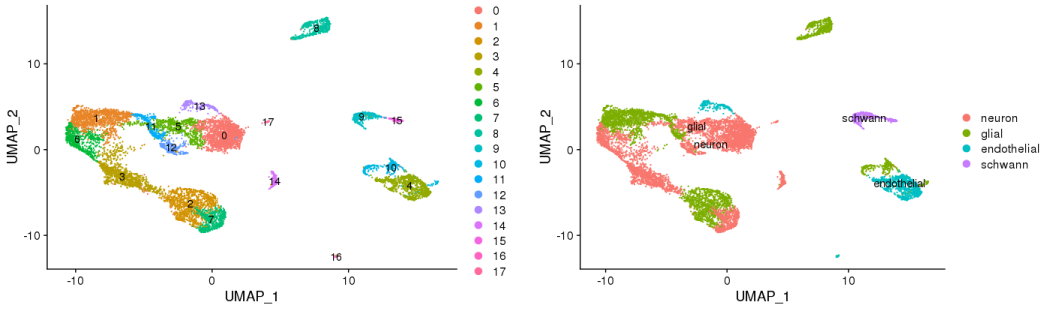
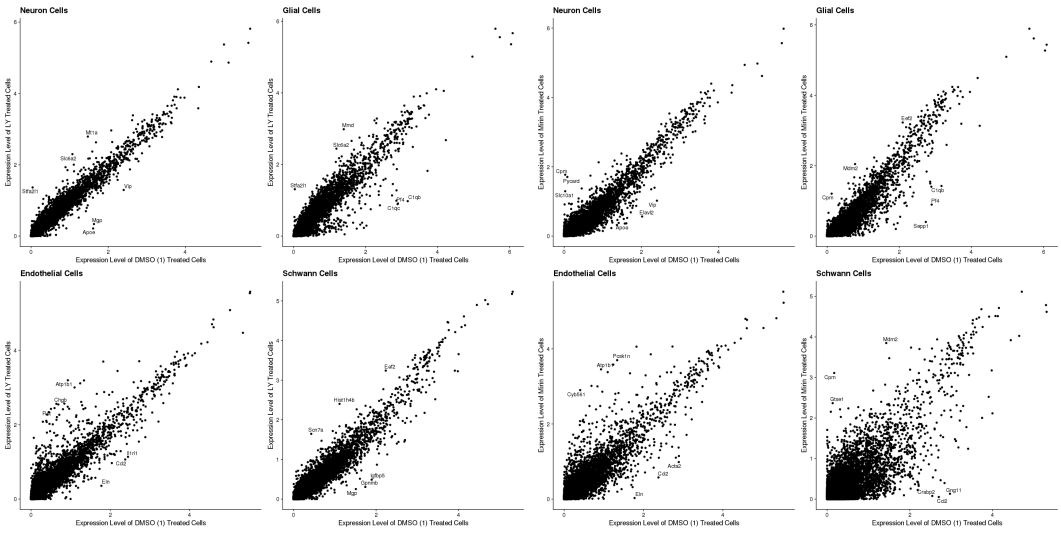


Figure 1: Clustering before and after identifying cell types through cell type markers

Experiment	Neuron	Glial	Endothelial	Schwann
DMSO Biological Replicate 1	1444	1075	437	147
DMSO Biological Replicate 2	1487	927	203	128
Treated with LY	717	638	207	87
Treated with mirin	1073	696	227	81

Table 1: Cell Numbers in Each Experiment

As shown in Table 1, compared to the first DMSO treated biological replicate, LY and mirin treated cultures had significantly less cell counts for every cell type. The second DMSO treated biological replicate also had more cells than the two drug treated cell cultures. This indicates an effect on cell counts from the drug treatment. The proportions of neuron, glial, endothelial, and Schwann cells in the control dmso treated culture are 46%, 34%, 14%, 5%, respectively. The proportions of cells within the LY treated culture are 43%, 38%, 12%, 5% within the neuron, glial, endothelial, and Schwann cells respectively. Within the mirin treated culture the neuron cell represented 51%, the glial cells represented 33%, the endothelial cells represented 10%, and the Schwann cells represented 3% of the entire culture. The glial cells with the LY treatment were affected more harshly than those treated with mirin. However, even with the treatment of drugs, the proportions of the cell types within the cell culture generally stayed within the same range, meaning the drugs did not severely impact one cell type specifically.



LY Average Expression Mirin Average Expression

Figure 2: Average Expression of Treated vs Control Cell Cultures, split by cell type

Seurat was also used to determine which genes were up and down regulated through the different drug treatments. Since the two cell cultures treated with DMSO are biological replicates, the one with more total cells was used as a control to determine differential expressed genes.

Experiment	Cell Type	Gene
Control (DMSO)	Neuron	Tuba1a
		Tuba1b
	Glial	Apoe
		Prdx1
	Endothelial	Gapdh
		Ifitm3
	Schwann	Npc2
		Prdx1
LY treated	Neuron	Stfa2l1
		Hist1h4b
	Glial	Th
		Rsrp1
	Endothelial	Ret
		Atp1b1
	Schwann	Eef2
		Eef1a1

Experiment	Cell Type	Gene
Control (DMSO)	Neuron	Elavl2
		Bri3
	Glial	Rps27l
		Bri3
	Endothelial	Fam111a
		Sod2
	Schwann	Gng11
		Nudt4
mirin treated	Neuron	Slc10a1
		Pycard
	Glial	Rpl9
		Eef2
	Endothelial	Atp1b1
		Ret
	Schwann	Cpm
		Gtse1

Table 2: Top Genes Expressed in Control vs Experimental Condition

The genes are then plotted for to visualize their average expression. As shown in Figure 2, there are genes that are visually outliers in average expression. This demonstrates that these genes had a stronger reaction to the drug than others. Those genes plotted above the trend would be considered up regulated in the treated group, as their expression is now higher than its control counterpart. The genes plotted below the trend would be considered down regulated within the treated culture. In the neuron cells treated with LY, Stfa2l1 was more up regulated than when it was only treated with DMSO. From the general shape of the plots, it would seem that the Schwann cells treated with mirin generally had a bigger response to the drug, as the dots are more wide spread. This is because the expression of a gene in the control is not expressed the same amount, deviating from the center trend.

Experiment	Cell Type	Pathway
LY treated	Neuron	negative regulation of neuron apoptotic process
		cellular response to manganese ion
	Glial	aging
		positive regulation of neuron projection development
	Endothelial	response to oxidative stress
		aging
	Schwann	translation
		protein folding
mirin Treated	Neuron	translation
		negative regulation of neuron apoptotic process
	Glial	translation
		aging
	Endothelial	aging
		response to drug
	Schwann	translation
		cell-cell adhesion

Table 3: Pathways Impacted by Ly and mirin Treatment

In Table 2, the top two genes expressed in each condition and cell cluster is shown. In the cultures treated with LY, in neuron cells, Stfa2l1 was more expressed than control, aligning with Figure 2. Genes Ret, Atp1b1, and Eef2 were also shown to be up regulated as compared to the control. There are also genes that were down regulated in the treated culture, such as Apoe. In the mirin treated cultures, genes like Slc10a1, Pycard, Cpm, Eef2, Atp1b1 were up regulated. Interestingly, Ret and Atp1b1 was the two two expressed genes within treated endothelial cells.

The pathways affected from the change in gene expression was determined through the use of Database for Annotation, Visualization and Integrated Discovery (DAVID). From Table 3, "aging"

pathway seemed to be most commonly impacted by the LY treatment through the different cell types. Other interesting pathways to note is "negative regulation of neuron apoptotic process" and "positive regulation of neuron projection development." These pathways would be integral in a cell's ability to grow and survive. Mirin impacted "aging", and "translation" through multiple cell types. Mirin also impacted "negative regulation of neuron apoptotic process," which is interesting as both drugs impacted that pathway. Since several pathways were impacted by both LY and mirin, it could suggest that that these pathways are more susceptible to drug treatment than other pathways.

3 Discussion

Through the results of this analysis, it is clear that LY and mirin affect the SCG cell cultures in a number of ways. Firstly, the cell count of cells decreased with treatment. This can be contributed to the fact that that LY and mirin inhibit pathways that deal with DNA repair and survival. With these pathways inhibited, the cells are expected to have lower survival rates. Interestingly, mirin treated neuron cells did not drop decrease as much as LY treated neuron cells. This could be directly related back to the fact that the "positive regulation of neuron projection development" was impacted by LY, but not by mirin.

The different cell types found in the SCG cultures are were neuron cells, glial cells, endothelial, and Schwann cells. Since the SCG is apart of the nervous system, finding neuron cells and their support cells was to be expected. Endothelial cells are cells that line blood and lymphatic vessels, and could be found within or near the SCG. From cell clustering in Figure 1, it is clear that the neuron and glial cells are intertwined, which can be contributed to the fact that glial are support cells for neuron cells.

Pathways such as "negative regulation of neuron apoptotic process," "positive regulation of neuron projection development," and "aging" are all related to the cell's ability to grow, age, and die. With the LY treatment, these three pathways are affected, supporting the idea that the drug LY can potentially decrease cell numbers. Mirin is also shown to be affecting the same "negative regulation of neuron apoptotic process" and "aging" pathways. It is interesting to note that mirin had an effect on the "response to drug" pathway, but LY did not have that pathway in its top affected pathways.

4 Methods

Cell Ranger was first employed to align data sets to the Rattus norvegicus genome. The Rattus norvegicus genome had to first be downloaded. Cell Ranger then aligned the original fastq files and generated feature counts in a matrix.

The matrix is then imported into Seurat, which helps make sense of the raw matrix data. Seurat allowed the data sets to be integrated so that it could be comparable. Integration of the data sets required "anchors," so that the data could be "normalized" among all of the data. Seurat allowed clustering of the cells to be visualized, and the cell markers related to each of the clusters.

From the cell markers, each cluster was assigned a cell type. From here on, the analysis of the data would not only be separated by experiment but also by cell type. Through Seurat, the markers of each experiment and cell type was identified. These cell markers would be the up and down regulated genes of each cell type within each experiment.

These genes would then be inputted into DAVID, to find the pathways affected by the change in gene expression.

References

[1] Yokota H., Mukai H., Hattori S., Yamada K., Anzai Y., Uno T. (2017). MR Imaging of the Superior Cervical Ganglion and Inferior Ganglion of the Vagus Nerve: Structures That Can Mimic Pathologic Retropharyngeal Lymph Nodes. *AJNR Am J Neuroradiol* 39(1), 170-176

[2] Mitsuoka, K., Kikutani, T., Sato, I. (2016). Morphological relationship between the superior cervical ganglion and cervical nerves in Japanese cadaver donors. *Brain and behavior* 7(2)

[3] Dupré, A., Boyer-Chatenet, L., Sattler, R. M., Modi, A. P., Lee, J. H., Nicolette, M. L., Kopelovich, L., Jasin, M., Baer, R., Paull, T. T., Gautier, J. (2008). *A forward chemical genetic screen reveals an inhibitor of the Mre11-Rad50-Nbs1 complex.* Nature chemical biology, 4(2), 119–125.

[4] Lavin MF., Birrell G., Chen P., Kozlov S., Scott S., Gueven N. (2005). ATM signaling and genomic stability in response to DNA damage. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, 569(1-2), 123-132.

[5] Shiloh Y. (2003). ATM and related protein kinases: safeguarding genome integrity. *Nature Reviews Cancer*, (3), 155-168.

[6] Hemmings, B. A., Restuccia, D. F. (2012). PI3K-PKB/Akt pathway. *Cold Spring Harbor perspectives in biology*, 4(9)

[7] Nicholson K.M., Quinn D.M., Kellett G.L., Warr J.R. (2003). LY294002, an inhibitor of phosphatidylinositol-3-kinase, causes preferential induction of apoptosis in human multidrug resistant cells. *Cancer Letters*, 190(1), 31-36

[8] Stuart T., Butler A., Hoffman P., Hafemeister C., Papalexi E., Mauck W.M. 3rd, Hao Y., Stoeckius M., Smibert P., Satija R. (2019). Comprehensive Integration of Single-Cell Data. *Cell*, 177(7), 1888-1902

All Code Used:

Cell Ranger Code - Aligning and Generating Feature Counts

`/gpfs/data/courses/bminga3004/FinalProjects/Project1`
copy over to `/gpfs/scratch/ebc308/FinalProject/Project1`

`uninf-dmso-exp1-fastq`
`uninf-dmso-exp4-fastq`
`uninf-ly-exp4-fastq`
`uninf-mirin-exp4-fastq`

Step #1: align against latest version of /Rattus norvegicus/genome & generate counts

- Cell Ranger (powered by STAR) - genome level
  - https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/what-is-cell-ranger

Downloading Rat (Rattus norvegicus) Genome

```
wget http://igenomes.illumina.com.s3-website-us-east-1.amazonaws.com/
Rattus_norvegicus/UCSC/rn6/Rattus_norvegicus_UCSC_rn6.tar.gz

tar -zxvf Rattus_norvegicus_UCSC_rn6.tar.gz
```

Align & Generate Counts against Rattus norvegicus using Cell Ranger

Making Reference file

```
srun -c1 -t5:00:00 --mem=32gb --pty /bin/bash ## need larger mem than 8gb

module load cellranger/3.1.0

cellranger mkref --genome=rat_genome --fasta=/gpfs/scratch/ebc308/
Rattus_norvegicus/UCSC/rn6/Sequence/WholeGenomeFasta/genome.fa --genes=/gpfs/
scratch/ebc308/Rattus_norvegicus/UCSC/rn6/Annotation/Genes/genes.gtf
```

Cell Ranger

```
#!/bin/bash
#SBATCH --job-name=cellRanger # Job name
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=Erica.Chio@nyulangone.org # Where to send mail
#SBATCH --nodes=4 # Nodes
#SBATCH --cpus-per-task=8
#SBATCH --ntasks=8 # Run on a single CPU
#SBATCH --mem=64gb # Job memory request
#SBATCH --time=36:00:00 # Time limit hrs:min:sec
#SBATCH --output=cellRanger_%j.log # Standard output and error log
#SBATCH -p cpu_short # Specifies location to submit job

module purge

module load cellranger/3.1.0

echo $1

cellranger count --id=$1 \
                 --transcriptome=/gpfs/scratch/ebc308/FinalProject/scripts/
rat_genome \
                 --fastqs=/gpfs/scratch/ebc308/FinalProject/Project1/$1
```

```
sbatch cellRangerCounts.sh uninf-dmso-exp1-fastq #completed

# upped mem to 64, time limit 36hr
sbatch cellRangerCounts.sh uninf-dmso-exp4-fastq #completed
sbatch cellRangerCounts.sh uninf-ly-exp4-fastq #completed
sbatch cellRangerCounts.sh uninf-mirin-exp4-fastq #completed
```

Seurat Code, from Integration to Visualizing Differentially Expressed Genes

```
suppressMessages(library(dplyr))
suppressMessages(library(Seurat))
suppressMessages(library(cowplot))
suppressMessages(library(patchwork))
```

Loading Data into Seurat Objects

```
# Load the uninf_dmso_exp1_fastq dataset
uninf_dmso_exp1_fastq.data <- Read10X(data.dir = "/gpfs/scratch/ebc308/
FinalProject/scripts/uninf-dmso-exp1-fastq/outs/filtered_feature_bc_matrix")
# Initialize the Seurat object with the raw (non-normalized data).
uninf_dmso_exp1_fastq <- CreateSeuratObject(counts =
uninf_dmso_exp1_fastq.data, project = "uninf_dmso_exp1_fastq", min.cells = 3,
min.features = 200)

uninf_dmso_exp1_fastq$stim <- "uninf_dmso_exp1_fastq"
uninf_dmso_exp1_fastq[["percent.mt"]] <-
PercentageFeatureSet(uninf_dmso_exp1_fastq, pattern = "^MT-")
uninf_dmso_exp1_fastq <- subset(x = uninf_dmso_exp1_fastq, subset =
nFeature_RNA > 500 & percent.mt < 25)

uninf_dmso_exp1_fastq
```

An object of class Seurat
12691 features across 3103 samples within 1 assay
Active assay: RNA (12691 features, 0 variable features)

```
# Load the uninf_dmso_exp4_fastq dataset
uninf_dmso_exp4_fastq.data <- Read10X(data.dir = "/gpfs/scratch/ebc308/
```

```
FinalProject/scripts/uninf-dmso-exp4-fastq/outs/filtered_feature_bc_matrix")
# Initialize the Seurat object with the raw (non-normalized data).
uninf_dmso_exp4_fastq <- CreateSeuratObject(counts =
uninf_dmso_exp4_fastq.data, project = "uninf_dmso_exp4_fastq", min.cells = 3,
min.features = 200)

uninf_dmso_exp4_fastq$stim <- "uninf_dmso_exp4_fastq"
uninf_dmso_exp4_fastq[["percent.mt"]] <-
PercentageFeatureSet(uninf_dmso_exp4_fastq, pattern = "^MT-")
uninf_dmso_exp4_fastq <- subset(x = uninf_dmso_exp4_fastq, subset =
nFeature_RNA > 500 & percent.mt < 25)

uninf_dmso_exp4_fastq
```

An object of class Seurat
12703 features across 2745 samples within 1 assay
Active assay: RNA (12703 features, 0 variable features)

```
# Load the uninf_ly_exp4_fastq.data dataset
uninf_ly_exp4_fastq.data <- Read10X(data.dir = "/gpfs/scratch/ebc308/
FinalProject/scripts/uninf-ly-exp4-fastq/outs/filtered_feature_bc_matrix")
# Initialize the Seurat object with the raw (non-normalized data).
uninf_ly_exp4_fastq <- CreateSeuratObject(counts = uninf_ly_exp4_fastq.data,
project = "uninf_ly_exp4_fastq", min.cells = 3, min.features = 200)

uninf_ly_exp4_fastq$stim <- "uninf_ly_exp4_fastq"
uninf_ly_exp4_fastq[["percent.mt"]] <-
PercentageFeatureSet(uninf_ly_exp4_fastq, pattern = "^MT-")
uninf_ly_exp4_fastq <- subset(x = uninf_ly_exp4_fastq, subset = nFeature_RNA >
500 & percent.mt < 25)

uninf_ly_exp4_fastq
```

An object of class Seurat
12294 features across 1649 samples within 1 assay
Active assay: RNA (12294 features, 0 variable features)

```

```

```
# Load the uninf_mirin_exp4_fastq dataset
uninf_mirin_exp4_fastq.data <- Read10X(data.dir = "/gpfs/scratch/ebc308/
FinalProject/scripts/uninf-mirin-exp4-fastq/outs/filtered_feature_bc_matrix")
# Initialize the Seurat object with the raw (non-normalized data).
uninf_mirin_exp4_fastq <- CreateSeuratObject(counts =
uninf_mirin_exp4_fastq.data, project = "uninf_mirin_exp4_fastq", min.cells = 3,
min.features = 200)

uninf_mirin_exp4_fastq$stim <- "uninf_mirin_exp4_fastq"
uninf_mirin_exp4_fastq[["percent.mt"]] <-
PercentageFeatureSet(uninf_mirin_exp4_fastq, pattern = "^MT-")
uninf_mirin_exp4_fastq <- subset(x = uninf_mirin_exp4_fastq, subset =
nFeature_RNA > 500 & percent.mt < 25)

uninf_mirin_exp4_fastq
```

An object of class Seurat
12518 features across 2077 samples within 1 assay
Active assay: RNA (12518 features, 0 variable features)

Normalizing Data and Preprocessing

```
# normalize the data
uninf_dmso_exp1_fastq <- NormalizeData(uninf_dmso_exp1_fastq, verbose=FALSE)
uninf_dmso_exp4_fastq <- NormalizeData(uninf_dmso_exp4_fastq, verbose=FALSE)
uninf_ly_exp4_fastq <- NormalizeData(uninf_ly_exp4_fastq, verbose=FALSE)
uninf_mirin_exp4_fastq <- NormalizeData(uninf_mirin_exp4_fastq,verbose=FALSE)
```

```
uninf_dmso_exp1_fastq <- FindVariableFeatures(uninf_dmso_exp1_fastq,
selection.method = "vst", nfeatures = 2000)
uninf_dmso_exp4_fastq <- FindVariableFeatures(uninf_dmso_exp4_fastq,
selection.method = "vst", nfeatures = 2000)
uninf_ly_exp4_fastq <- FindVariableFeatures(uninf_ly_exp4_fastq,
selection.method = "vst", nfeatures = 2000)
uninf_mirin_exp4_fastq <- FindVariableFeatures(uninf_mirin_exp4_fastq,
selection.method = "vst", nfeatures = 2000)
```

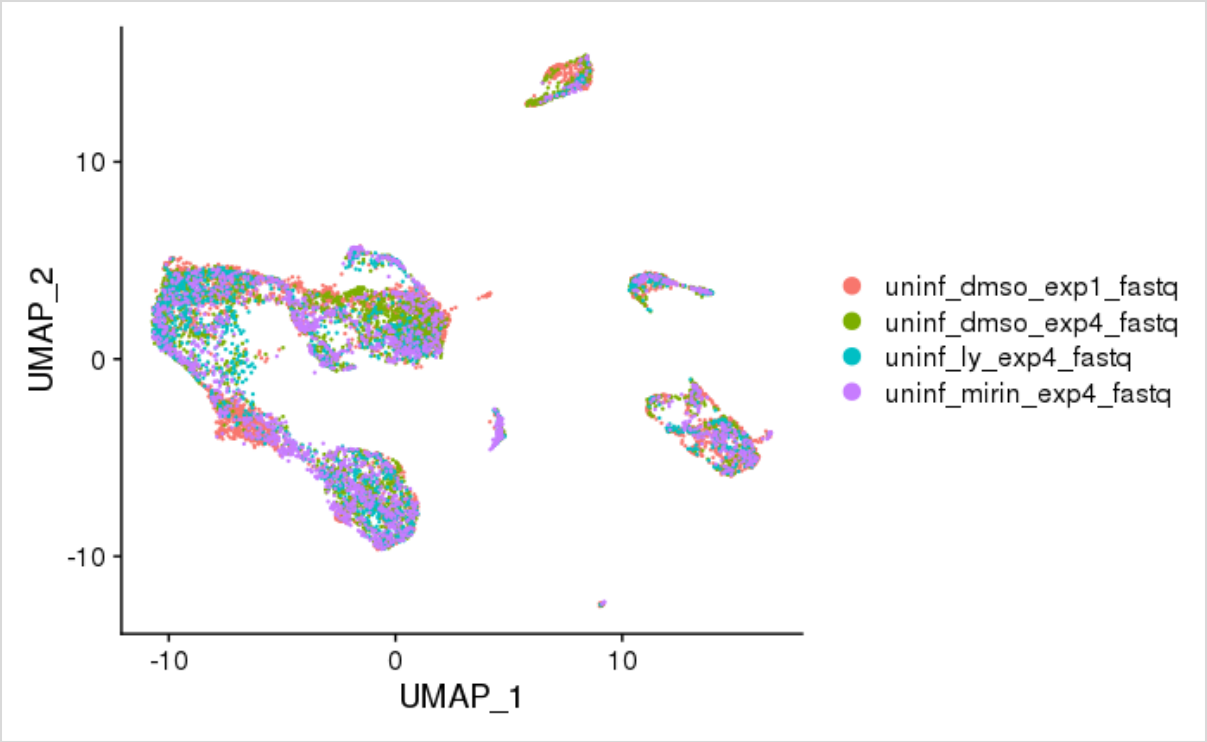
Integration of the 4 Datasets, and finding anchors

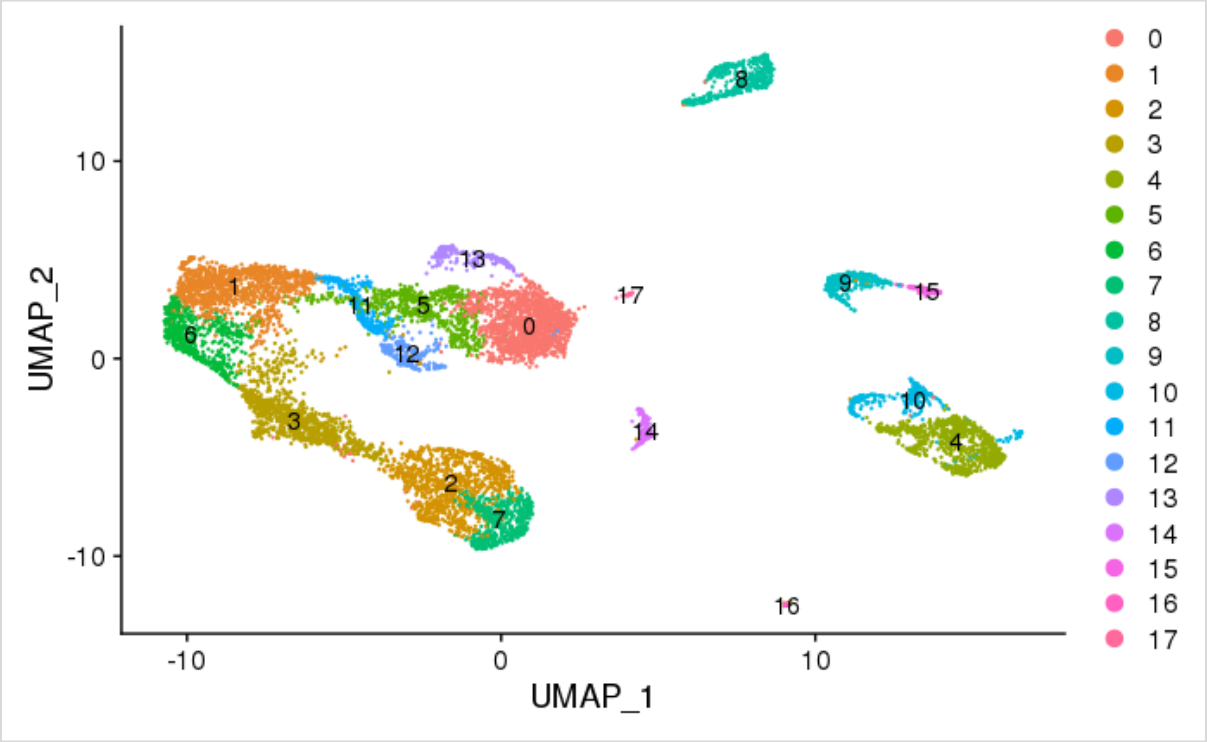
```
#Integration of 4 cell datasets
integrated_list <-c(uninf_dmso_exp1_fastq, uninf_dmso_exp4_fastq,
uninf_ly_exp4_fastq, uninf_mirin_exp4_fastq)
integrated_anchors <- FindIntegrationAnchors(object.list = integrated_list,
dims = 1:20)
integrated_data <- IntegrateData(anchorset = integrated_anchors, dims = 1:20)
```

Integrated Analysis

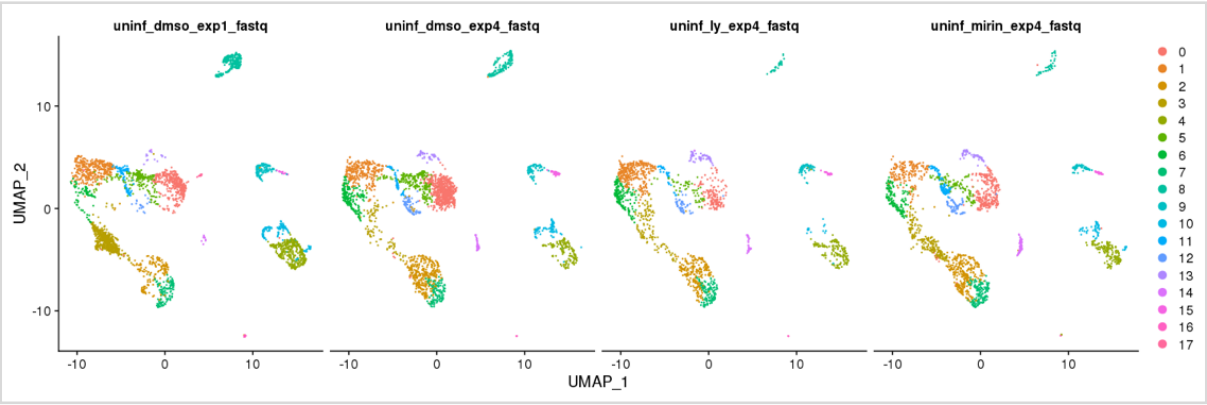
```
DefaultAssay(integrated_data) <- "integrated"

# Run the standard workflow for visualization and clustering
integrated_data <- ScaleData(integrated_data, verbose = FALSE)
integrated_data <- RunPCA(integrated_data, npcs = 30, verbose = FALSE)
# t-SNE and Clustering
integrated_data <- RunUMAP(integrated_data, reduction = "pca", dims = 1:20)
integrated_data <- FindNeighbors(integrated_data, reduction = "pca", dims =
1:20)
integrated_data <- FindClusters(integrated_data, resolution = 0.5)
```

```
# Visualization
p1 <- DimPlot(integrated_data, reduction = "umap", group.by = "stim")
p2 <- DimPlot(integrated_data, reduction = "umap", label = TRUE)
# plot_grid(p1, p2)
p1
p2
```

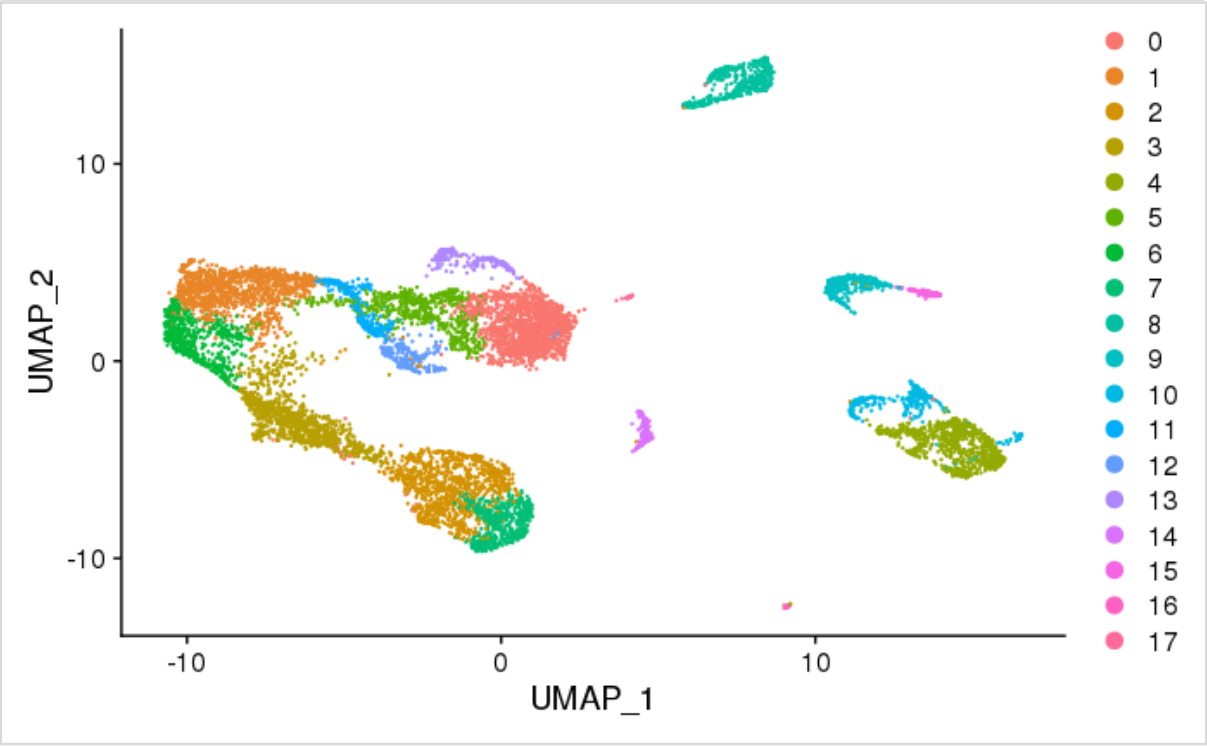





```
DimPlot(integrated_data, reduction = "umap", split.by = "stim")
```




```
DimPlot(integrated_data, reduction = "umap")
```



Identify conserved cell type markers in each cluster

```
DefaultAssay(integrated_data) <- "RNA"

cluster_0 <- FindConservedMarkers(integrated_data, ident.1 = 0, grouping.var =
"stim", verbose = FALSE)
cluster_1 <- FindConservedMarkers(integrated_data, ident.1 = 1, grouping.var =
"stim", verbose = FALSE)
cluster_2 <- FindConservedMarkers(integrated_data, ident.1 = 2, grouping.var =
"stim", verbose = FALSE)
```

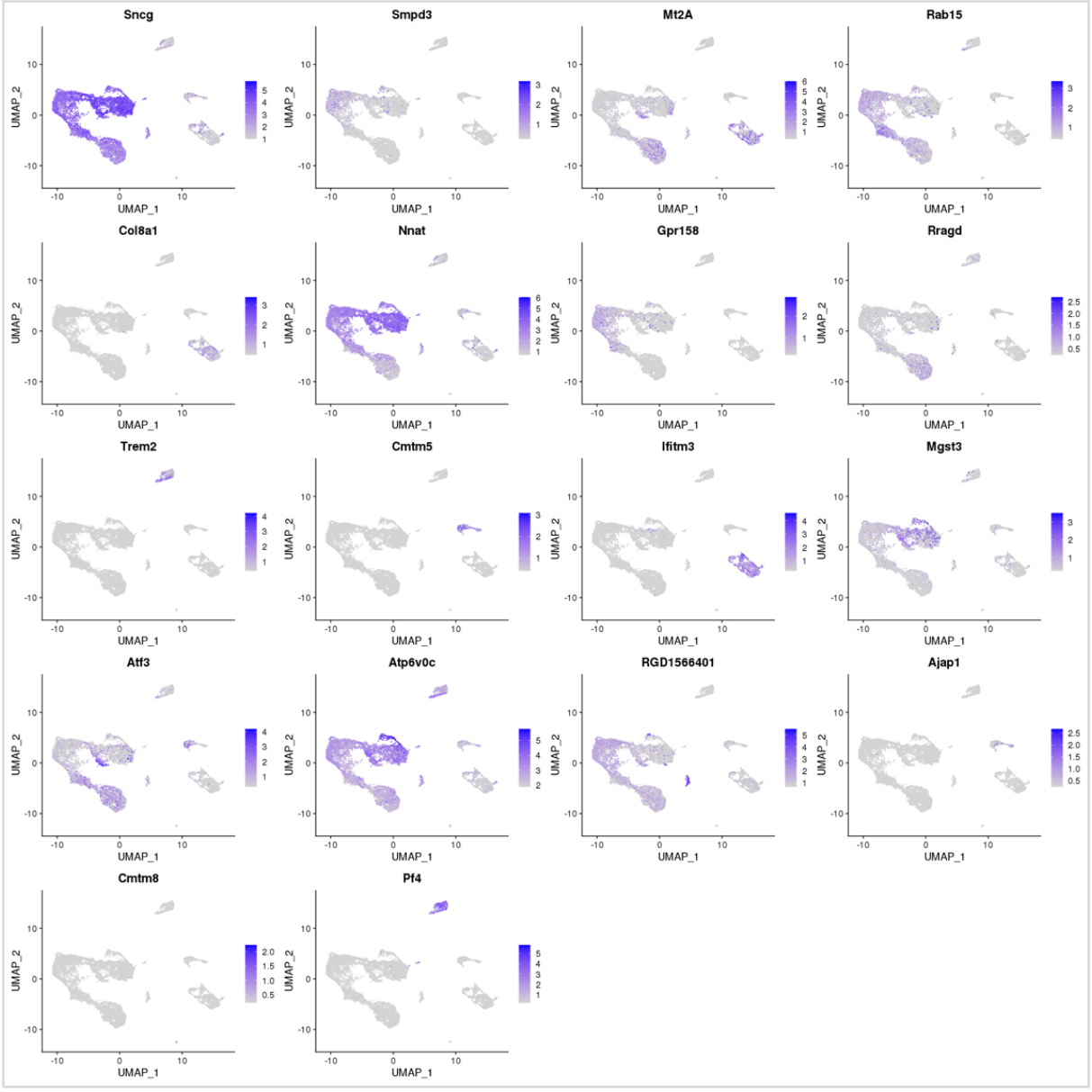
```
cluster_3 <- FindConservedMarkers(integrated_data, ident.1 = 3, grouping.var =
"stim", verbose = FALSE)
cluster_4 <- FindConservedMarkers(integrated_data, ident.1 = 4, grouping.var =
"stim", verbose = FALSE)
cluster_5 <- FindConservedMarkers(integrated_data, ident.1 = 5, grouping.var =
"stim", verbose = FALSE)
cluster_6 <- FindConservedMarkers(integrated_data, ident.1 = 6, grouping.var =
"stim", verbose = FALSE)
cluster_7 <- FindConservedMarkers(integrated_data, ident.1 = 7, grouping.var =
"stim", verbose = FALSE)
cluster_8 <- FindConservedMarkers(integrated_data, ident.1 = 8, grouping.var =
"stim", verbose = FALSE)
cluster_9 <- FindConservedMarkers(integrated_data, ident.1 = 9, grouping.var =
"stim", verbose = FALSE)
cluster_10 <- FindConservedMarkers(integrated_data, ident.1 = 10, grouping.var
= "stim", verbose = FALSE)
cluster_11 <- FindConservedMarkers(integrated_data, ident.1 = 11, grouping.var
= "stim", verbose = FALSE)
cluster_12 <- FindConservedMarkers(integrated_data, ident.1 = 12, grouping.var
= "stim", verbose = FALSE)
cluster_13 <- FindConservedMarkers(integrated_data, ident.1 = 13, grouping.var
= "stim", verbose = FALSE)
cluster_14 <- FindConservedMarkers(integrated_data, ident.1 = 14, grouping.var
= "stim", verbose = FALSE)
cluster_15 <- FindConservedMarkers(integrated_data, ident.1 = 15, grouping.var
= "stim", verbose = FALSE)
```

```
cluster_16 <- FindConservedMarkers(integrated_data, ident.1 = 16, grouping.var
= "stim", verbose = FALSE, min.cells.group = 2)
```

```
cluster_17 <- FindConservedMarkers(integrated_data, ident.1 = 17, grouping.var
= "stim", verbose = FALSE)
```

Visualize top Cell Marker

```
top_gene_marker <- c(rownames(cluster_0)[1], rownames(cluster_1)[1],
rownames(cluster_2)[1], rownames(cluster_3)[1], rownames(cluster_4)[1],
rownames(cluster_5)[1], rownames(cluster_6)[1], rownames(cluster_7)[1],
rownames(cluster_8)[1], rownames(cluster_9)[1], rownames(cluster_10)[1],
rownames(cluster_11)[1], rownames(cluster_12)[1], rownames(cluster_13)[1],
rownames(cluster_14)[1], rownames(cluster_15)[1],  rownames(cluster_16)[1],
rownames(cluster_17)[1])

FeaturePlot(integrated_data, features = top_gene_marker, min.cutoff = "q9")
```

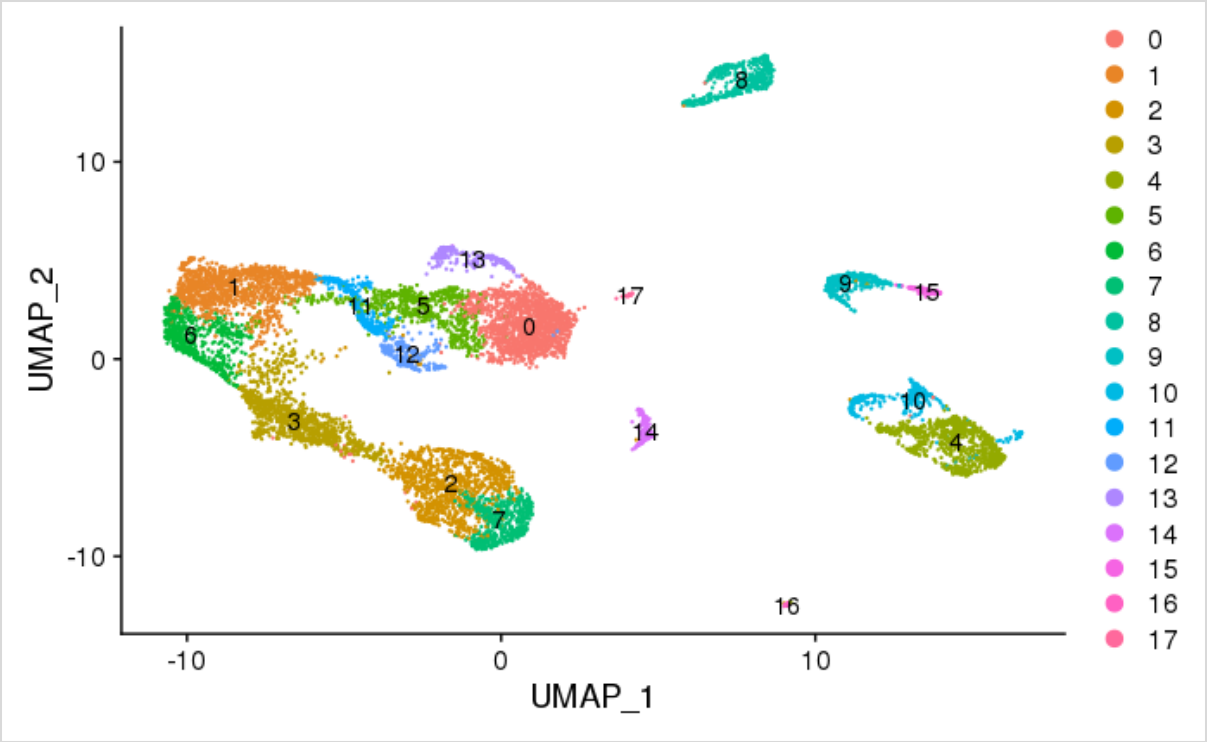


Create CSV of top 5 Cell Markers

```
#print out csv
top5_gene_marker <- c(rownames(cluster_0)[1:5], rownames(cluster_1)[1:5],
rownames(cluster_2)[1:5], rownames(cluster_3)[1:5], rownames(cluster_4)[1:5],
rownames(cluster_5)[1:5], rownames(cluster_6)[1:5], rownames(cluster_7)[1:5],
rownames(cluster_8)[1:5], rownames(cluster_9)[1:5], rownames(cluster_10)[1:5],
rownames(cluster_11)[1:5], rownames(cluster_12)[1:5], rownames(cluster_13)
[1:5], rownames(cluster_14)[1:5], rownames(cluster_15)[1:5],
rownames(cluster_16)[1:5], rownames(cluster_17)[1:5])

cluster <- c(rep("0", 5), rep("1", 5), rep("2", 5), rep("3", 5), rep("4", 5),
rep("5", 5), rep("6", 5), rep("7", 5), rep("8", 5), rep("9", 5), rep("10", 5),
rep("11", 5), rep("12", 5), rep("13", 5), rep("14", 5), rep("15", 5), rep("16",
5), rep("17", 5))

d <- data.frame("cluster" <- cluster, "gene_markers" = top5_gene_marker)

write.csv(d,"/gpfs/home/ebc308/finalproject/cluster_gene_markers.csv",
row.names = FALSE)
```

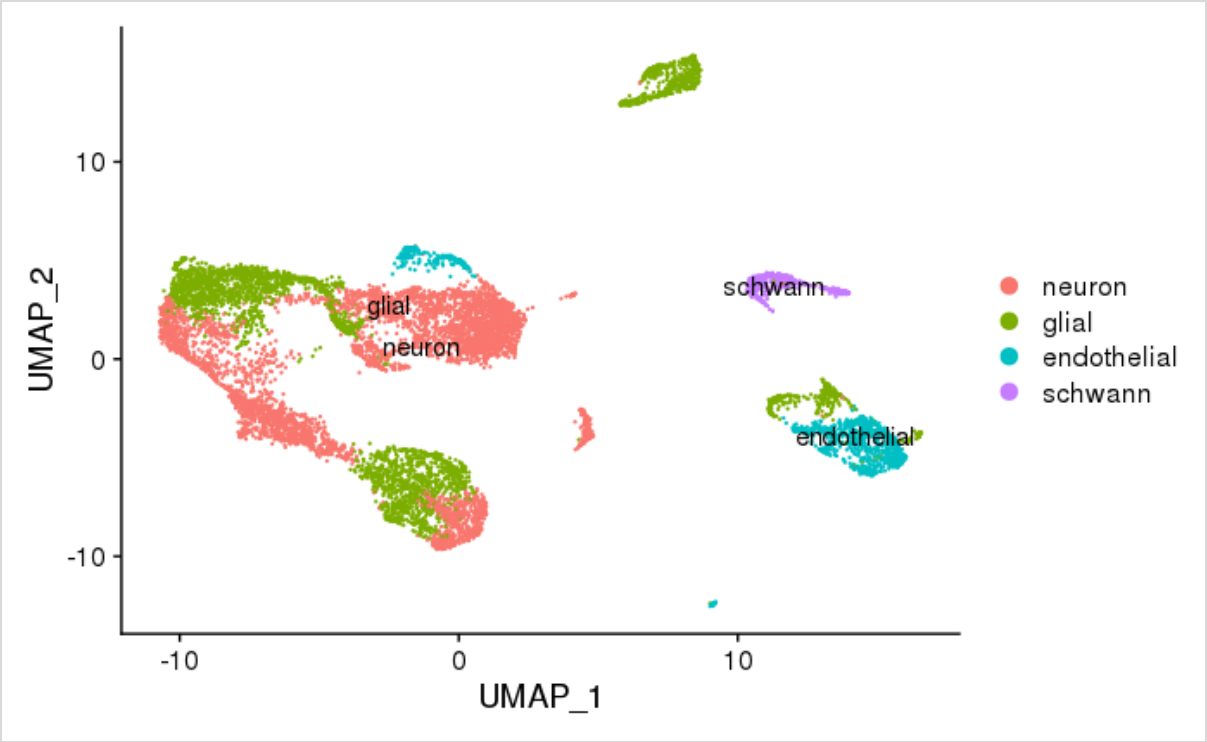
Cluster Visualization and Naming

```
DimPlot(integrated_data, label = TRUE)
```

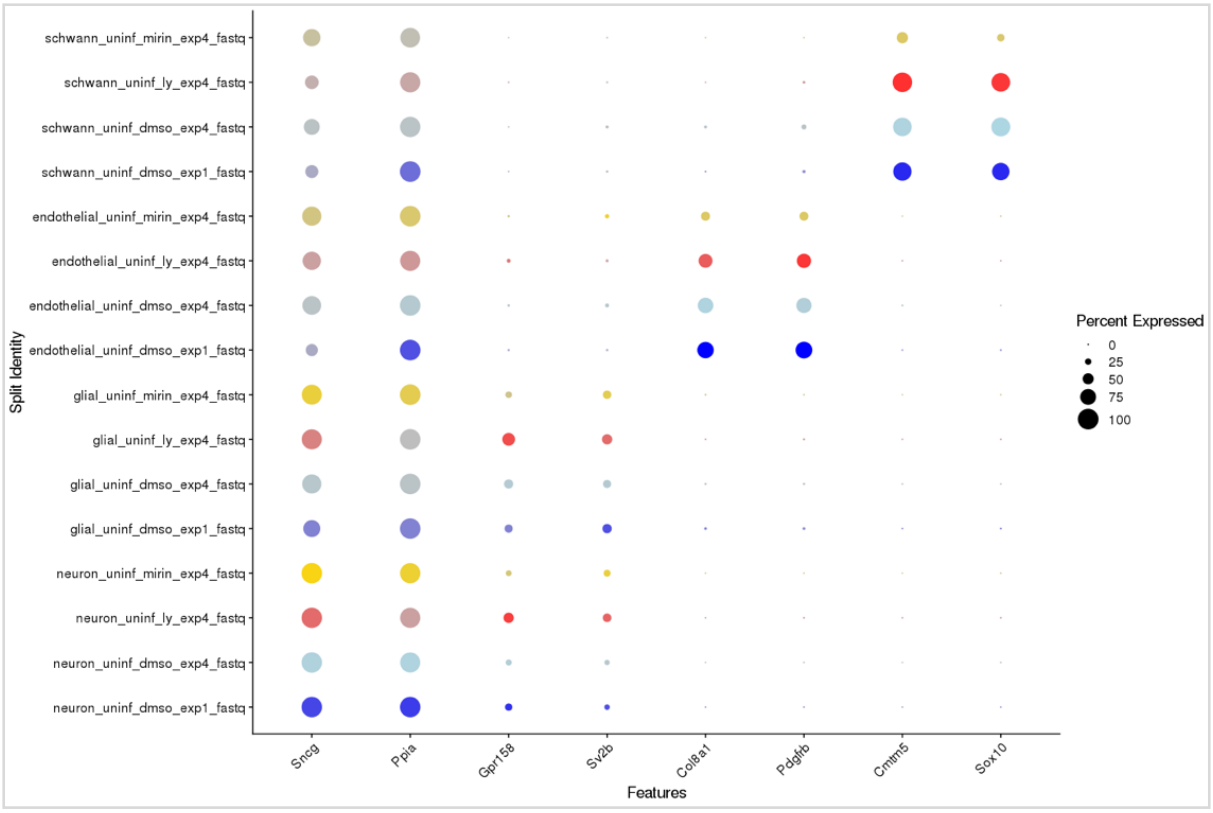


Assign Cell Type to Cluster

```
integrated_data <- RenameIdents(integrated_data, `0` = "neuron", `1` = "glial", `2` = "glial", `3` = "neuron", `4` = "endothelial", `5` = "neuron", `6` = "neuron", `7` = "neuron", `8` = "glial", `9` = "schwann", `10` = "glial", `11` = "glial", `12` = "neuron", `13` = "endothelial", `14` = "neuron", `15` = "schwann", `16` = "endothelial", `17` = "neuron")

DimPlot(integrated_data, label = TRUE)
```




```
Idents(integrated_data) <- factor(Idents(integrated_data), levels = c("neuron",
"glial", "endothelial", "schwann"))

top2_gene_marker <- c(rownames(cluster_0)[1:2], rownames(cluster_6)[1:2],
rownames(cluster_4)[1:2],  rownames(cluster_9)[1:2])

markers.to.plot <- top2_gene_marker[!duplicated(top2_gene_marker)]

DotPlot(integrated_data, features = rev(markers.to.plot), cols = c("blue",
"lightblue", "firebrick1", "gold"), dot.scale = 8, split.by = "stim") +
RotatedAxis()
```



Visualize Differentially Expressed Genes

```
library(ggplot2)
library(ggrepel)
library(cowplot)
theme_set(theme_cowplot())
```

Visualizing Outliers of Average Gene Expression

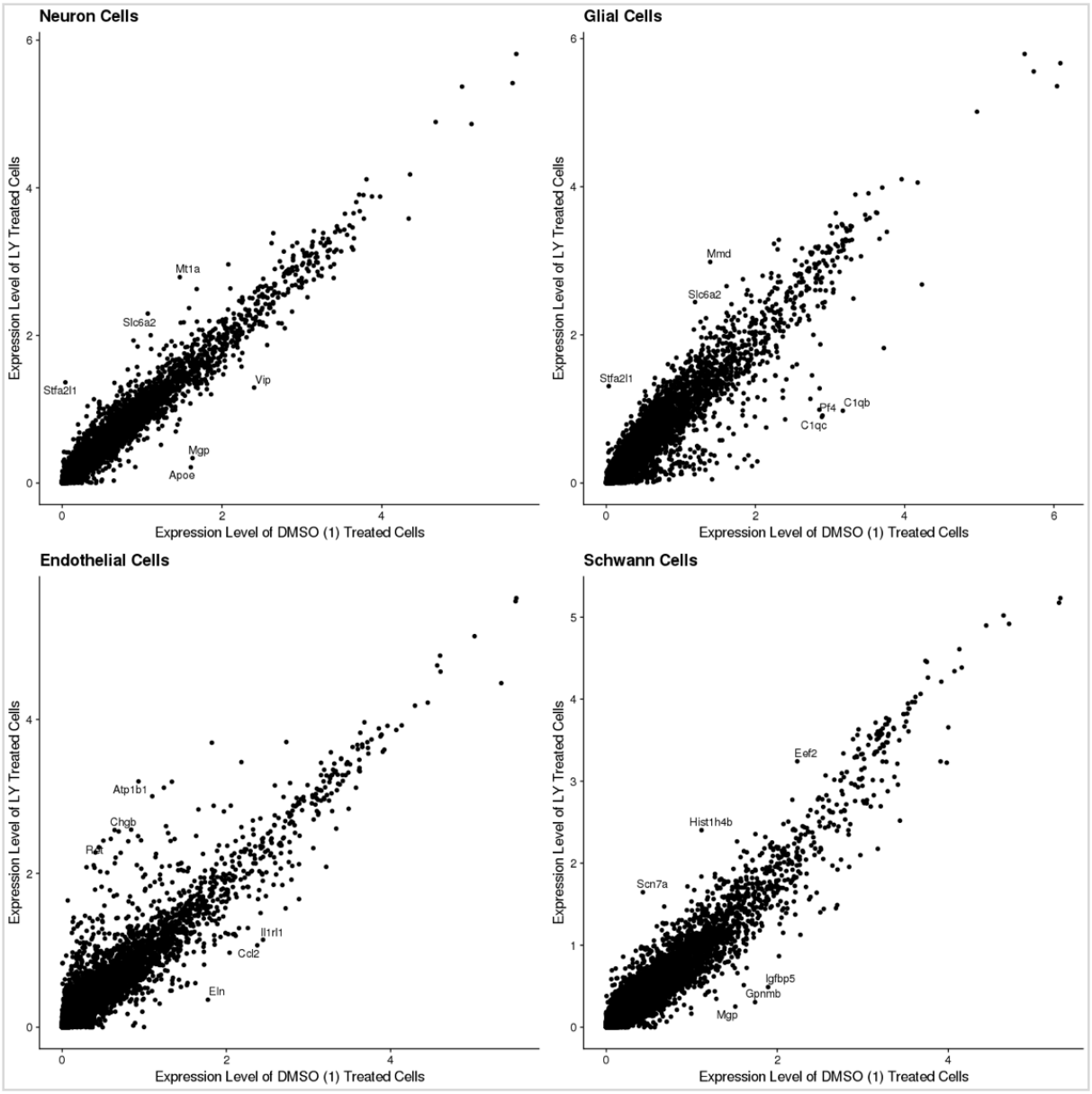
```
neuron.cells <- subset(integrated_data, idents = "neuron")
Idents(neuron.cells) <- "stim"
avg.neuron.cells <- log1p(AverageExpression(neuron.cells, verbose = FALSE)$RNA)
avg.neuron.cells$gene <- rownames(avg.neuron.cells)

glial.cells <- subset(integrated_data, idents = "glial")
Idents(glial.cells) <- "stim"
avg.glial.cells <- log1p(AverageExpression(glial.cells, verbose = FALSE)$RNA)
avg.glial.cells$gene <- rownames(avg.glial.cells)
```

```
endothelial.cells <- subset(integrated_data, idents = "endothelial")
Idents(endothelial.cells) <- "stim"
avg.endothelial.cells <- log1p(AverageExpression(endothelial.cells, verbose =
FALSE)$RNA)
avg.endothelial.cells$gene <- rownames(avg.endothelial.cells)

schwann.cells <- subset(integrated_data, idents = "schwann")
Idents(schwann.cells) <- "stim"
avg.schwann.cells <- log1p(AverageExpression(schwann.cells, verbose = FALSE)
$RNA)
avg.schwann.cells$gene <- rownames(avg.schwann.cells)
```

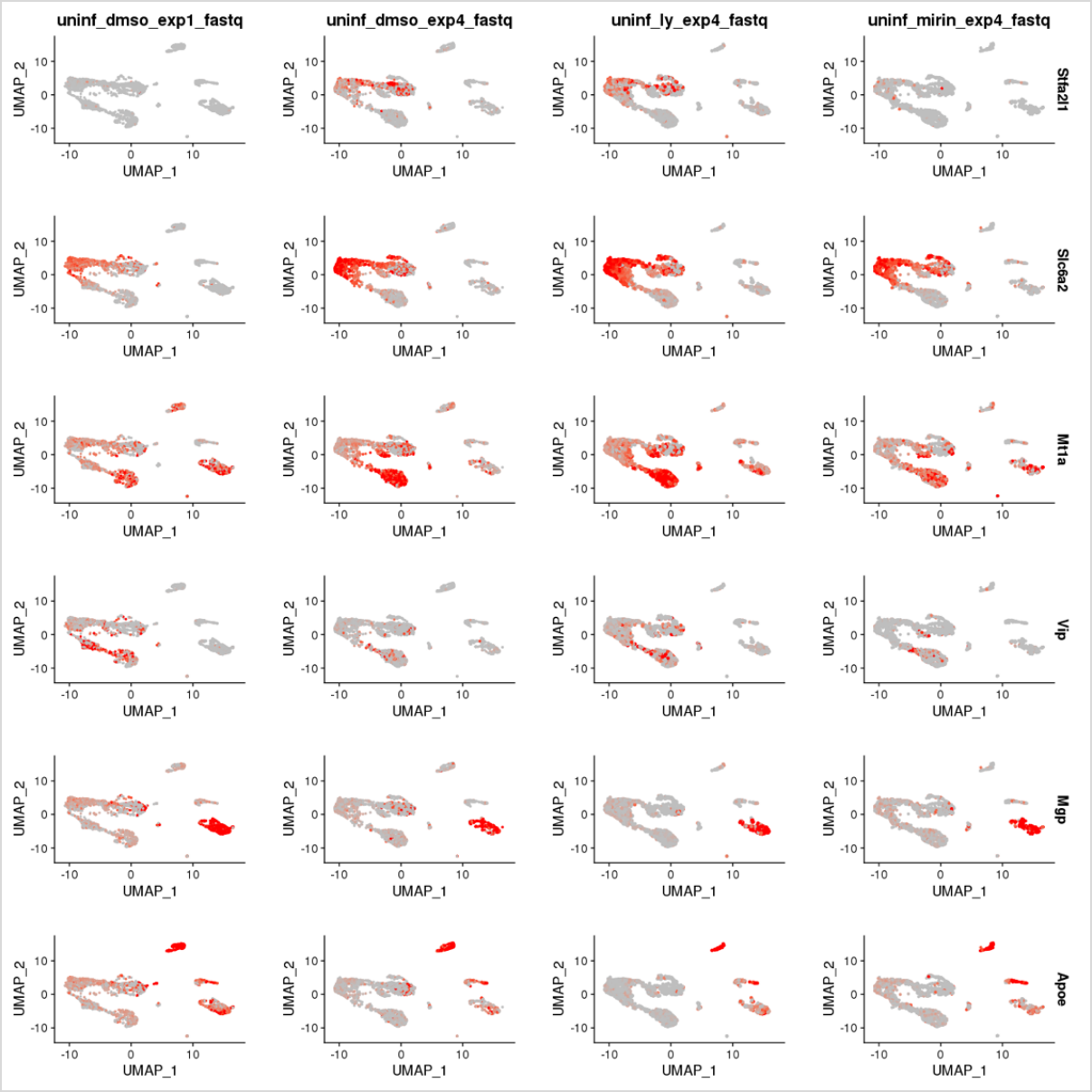
```
diff_neuron_ly <- avg.neuron.cells  %>% select(uninf_dmso_exp1_fastq,
uninf_ly_exp4_fastq)
diff_neuron_ly$diff <- (diff_neuron_ly$uninf_dmso_exp1_fastq -
diff_neuron_ly$uninf_ly_exp4_fastq)
diff_neuron_ly <- diff_neuron_ly[(order(diff_neuron_ly$diff)),]
# diff_neuron_ly$label <- ""
# ix_label <- c(1:5)
# diff_neuron_ly$label[ix_label] <- rownames(diff_neuron_ly)[ix_label]
# diff_neuron_ly

diff_glial_ly <- avg.glial.cells  %>% select(uninf_dmso_exp1_fastq,
uninf_ly_exp4_fastq)
diff_glial_ly$diff <- (diff_glial_ly$uninf_dmso_exp1_fastq -
diff_glial_ly$uninf_ly_exp4_fastq)
diff_glial_ly <- diff_glial_ly[(order(diff_glial_ly$diff)),]

diff_endothelial_ly <- avg.endothelial.cells  %>% select(uninf_dmso_exp1_fastq,
uninf_ly_exp4_fastq)
diff_endothelial_ly$diff <- (diff_endothelial_ly$uninf_dmso_exp1_fastq -
diff_endothelial_ly$uninf_ly_exp4_fastq)
diff_endothelial_ly <- diff_endothelial_ly[(order(diff_endothelial_ly$diff)),]

diff_schwann_ly <- avg.schwann.cells  %>% select(uninf_dmso_exp1_fastq,
uninf_ly_exp4_fastq)
diff_schwann_ly$diff <- (diff_schwann_ly$uninf_dmso_exp1_fastq -
```

```
diff_schwann_ly$uninf_ly_exp4_fastq)
diff_schwann_ly <- diff_schwann_ly[(order(diff_schwann_ly$diff)),]
```

```
p1 <- ggplot(avg.neuron.cells, aes(uninf_dmso_exp1_fastq, uninf_ly_exp4_fastq,
label=diff_neuron_ly$label)) + geom_point() + ggtitle("Neuron Cells") +
ylab("Expression Level of LY Treated Cells") + xlab("Expression Level of DMSO
(1) Treated Cells")
p1 <- LabelPoints(plot = p1, points = c(rownames(diff_neuron_ly)[1:3],
tail(rownames(diff_neuron_ly), 3)), repel = TRUE, xnudge = 0, ynudge= 0)
p2 <- ggplot(avg.glial.cells, aes(uninf_dmso_exp1_fastq, uninf_ly_exp4_fastq))
+ geom_point() + ggtitle("Glial Cells") + ylab("Expression Level of LY Treated
Cells") + xlab("Expression Level of DMSO (1) Treated Cells")
p2 <- LabelPoints(plot = p2, points = c(rownames(diff_glial_ly)[1:3],
tail(rownames(diff_glial_ly), 3)), repel = TRUE, xnudge = 0, ynudge= 0)
p3 <- ggplot(avg.endothelial.cells, aes(uninf_dmso_exp1_fastq,
uninf_ly_exp4_fastq)) + geom_point() + ggtitle("Endothelial Cells") +
ylab("Expression Level of LY Treated Cells") + xlab("Expression Level of DMSO
(1) Treated Cells")
p3 <- LabelPoints(plot = p3, points = c(rownames(diff_endothelial_ly)[1:3],
tail(rownames(diff_endothelial_ly), 3)), repel = TRUE, xnudge = 0, ynudge= 0)
p4 <- ggplot(avg.schwann.cells, aes(uninf_dmso_exp1_fastq,
uninf_ly_exp4_fastq)) + geom_point() + ggtitle("Schwann Cells") +
ylab("Expression Level of LY Treated Cells") + xlab("Expression Level of DMSO
(1) Treated Cells")
p4 <- LabelPoints(plot = p4, points = c(rownames(diff_schwann_ly)[1:3],
tail(rownames(diff_schwann_ly), 3) ), repel = TRUE, xnudge = 0, ynudge= 0)
plot_grid(p1, p2, p3, p4)
```




```
FeaturePlot(integrated_data, features = c("Stfa2l1", "Slc6a2", "Mt1a", "Vip",
"Mgp", "Apoe"), split.by = "stim", max.cutoff = 3,
    cols = c("grey", "red"))
```

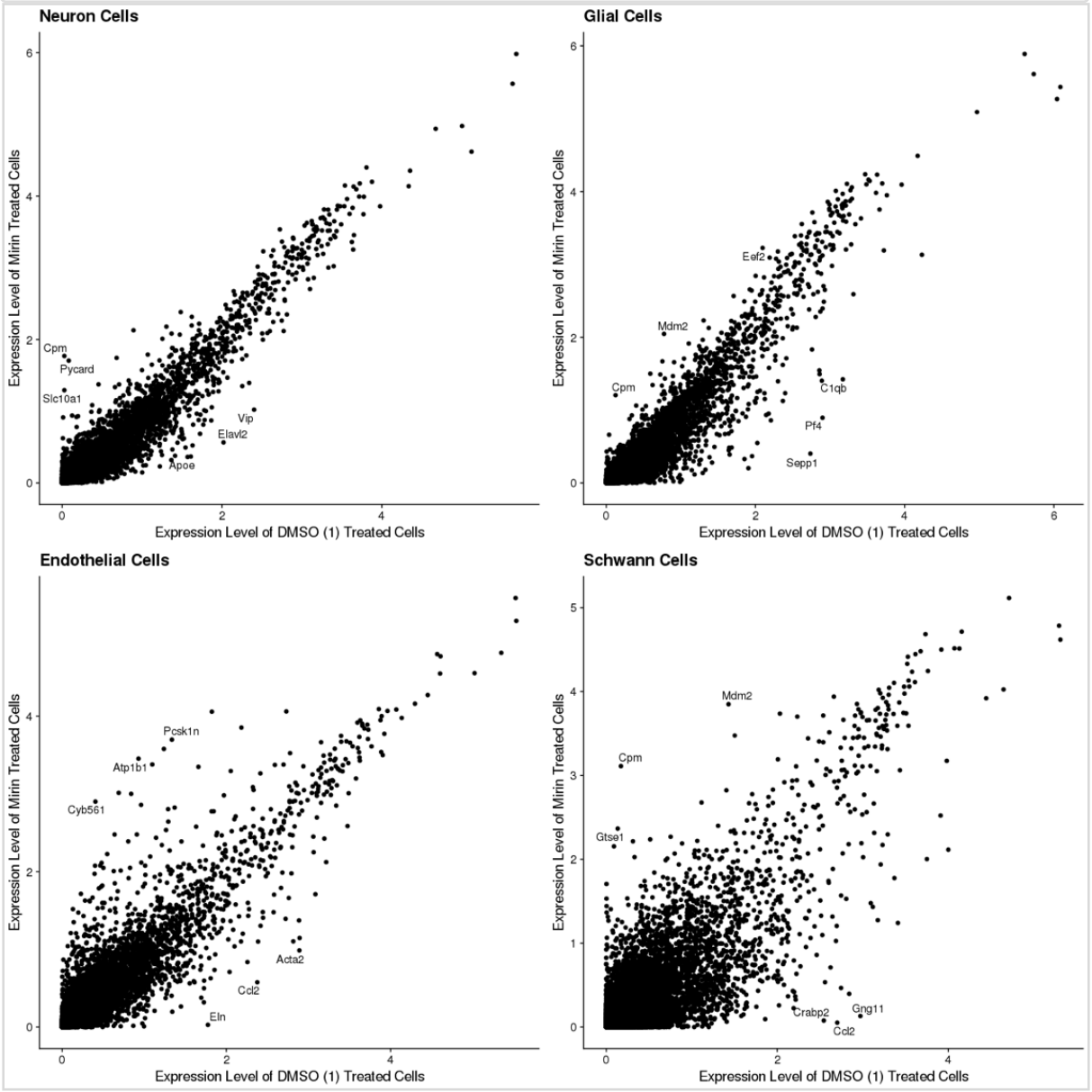



```
diff_neuron_mirin <- avg.neuron.cells  %>% select(uninf_dmso_exp1_fastq,
uninf_mirin_exp4_fastq)
diff_neuron_mirin$diff <- (diff_neuron_mirin$uninf_dmso_exp1_fastq -
diff_neuron_mirin$uninf_mirin_exp4_fastq)
diff_neuron_mirin <- diff_neuron_mirin[(order(diff_neuron_mirin$diff)),]
# diff_neuron_mirin$label <- ""
# ix_label <- c(1:5)
# diff_neuron_mirin$label[ix_label] <- rownames(diff_neuron_mirin)[ix_label]
# diff_neuron_mirin

diff_glial_mirin <- avg.glial.cells  %>% select(uninf_dmso_exp1_fastq,
uninf_mirin_exp4_fastq)
diff_glial_mirin$diff <- (diff_glial_mirin$uninf_dmso_exp1_fastq -
```

```
diff_glial_mirin$uninf_mirin_exp4_fastq)
diff_glial_mirin <- diff_glial_mirin[(order(diff_glial_mirin$diff)),]

diff_endothelial_mirin <- avg.endothelial.cells  %>%
select(uninf_dmso_exp1_fastq, uninf_mirin_exp4_fastq)
diff_endothelial_mirin$diff <- (diff_endothelial_mirin$uninf_dmso_exp1_fastq -
diff_endothelial_mirin$uninf_mirin_exp4_fastq)
diff_endothelial_mirin <-
diff_endothelial_mirin[(order(diff_endothelial_mirin$diff)),]

diff_schwann_mirin <- avg.schwann.cells  %>% select(uninf_dmso_exp1_fastq,
uninf_mirin_exp4_fastq)
diff_schwann_mirin$diff <- (diff_schwann_mirin$uninf_dmso_exp1_fastq -
diff_schwann_mirin$uninf_mirin_exp4_fastq)
diff_schwann_mirin <- diff_schwann_mirin[(order(diff_schwann_mirin$diff)),]
```

```
p5 <- ggplot(avg.neuron.cells, aes(uninf_dmso_exp1_fastq,
uninf_mirin_exp4_fastq)) + geom_point() + ggtitle("Neuron Cells") +
ylab("Expression Level of Mirin Treated Cells") + xlab("Expression Level of
DMSO (1) Treated Cells")
p5 <- LabelPoints(plot = p5, points = c(rownames(diff_neuron_mirin)[1:3] ,
tail(rownames(diff_neuron_mirin), 3)), repel = TRUE, xnudge = 0, ynudge= 0)
p6 <- ggplot(avg.glial.cells, aes(uninf_dmso_exp1_fastq,
uninf_mirin_exp4_fastq)) + geom_point() + ggtitle("Glial Cells") +
ylab("Expression Level of Mirin Treated Cells") + xlab("Expression Level of
DMSO (1) Treated Cells")
p6 <- LabelPoints(plot = p6, points = c(rownames(diff_glial_mirin)[1:3] ,
tail(rownames(diff_glial_mirin), 3)), repel = TRUE, xnudge = 0, ynudge= 0)
p7 <- ggplot(avg.endothelial.cells, aes(uninf_dmso_exp1_fastq,
uninf_mirin_exp4_fastq)) + geom_point() + ggtitle("Endothelial Cells") +
ylab("Expression Level of Mirin Treated Cells") + xlab("Expression Level of
DMSO (1) Treated Cells")
p7 <- LabelPoints(plot = p7, points = c(rownames(diff_endothelial_mirin)[1:3],
tail(rownames(diff_endothelial_mirin), 3) ), repel = TRUE, xnudge = 0, ynudge=
0)
p8 <- ggplot(avg.schwann.cells, aes(uninf_dmso_exp1_fastq,
uninf_mirin_exp4_fastq)) + geom_point() + ggtitle("Schwann Cells") +
```

```
ylab("Expression Level of Mirin Treated Cells") + xlab("Expression Level of
DMSO (1) Treated Cells")
p8 <- LabelPoints(plot = p8, points = c(rownames(diff_schwann_mirin)[1:3],
tail(rownames(diff_schwann_mirin), 3)), repel = TRUE, xnudge = 0, ynudge= 0)
plot_grid(p5, p6, p7, p8)
```



Gene Expression in Different Conditions for Each Cell Type

```
suppressMessages(library(dplyr))
suppressMessages(library(tibble))
```

```
suppressMessages(library(Seurat))
suppressMessages(library(cowplot))
suppressMessages(library(patchwork))
suppressMessages(library(ggplot2))
theme_set(theme_cowplot())
```

```
# only run 1 time
integrated_data$celltype.stim <- paste(Idents(integrated_data),
integrated_data$stim, sep = "_")
integrated_data$celltype <- Idents(integrated_data)
Idents(integrated_data) <- "celltype.stim"
```

Finding Markers of each Condition & Cell Type

```
neuron_dmso1_ly <- FindMarkers(integrated_data, ident.1 =
"neuron_uninf_dmso_exp1_fastq", ident.2 = "neuron_uninf_ly_exp4_fastq", verbose
= FALSE, min.pct=0.4)

neuron_dmso2_ly <- FindMarkers(integrated_data, ident.1 =
"neuron_uninf_dmso_exp4_fastq", ident.2 = "neuron_uninf_ly_exp4_fastq", verbose
= FALSE, min.pct=0.4)

glial_dmso1_ly <- FindMarkers(integrated_data, ident.1 =
"glial_uninf_dmso_exp1_fastq", ident.2 = "glial_uninf_ly_exp4_fastq", verbose =
FALSE, min.pct=0.4)

glial_dmso2_ly <- FindMarkers(integrated_data, ident.1 =
"glial_uninf_dmso_exp4_fastq", ident.2 = "glial_uninf_ly_exp4_fastq", verbose =
FALSE, min.pct=0.4)

endothelial_dmso1_ly <- FindMarkers(integrated_data, ident.1 =
"endothelial_uninf_dmso_exp1_fastq", ident.2 =
"endothelial_uninf_ly_exp4_fastq", verbose = FALSE, min.pct=0.4)

endothelial_dmso2_ly <- FindMarkers(integrated_data, ident.1 =
```

```
"endothelial_uninf_dmso_exp4_fastq", ident.2 =
"endothelial_uninf_ly_exp4_fastq", verbose = FALSE, min.pct=0.4)

schwann_dmso1_ly <- FindMarkers(integrated_data, ident.1 =
"schwann_uninf_dmso_exp1_fastq", ident.2 = "schwann_uninf_ly_exp4_fastq",
verbose = FALSE, min.pct=0.4)

schwann_dmso2_ly <- FindMarkers(integrated_data, ident.1 =
"schwann_uninf_dmso_exp4_fastq", ident.2 = "schwann_uninf_ly_exp4_fastq",
verbose = FALSE, min.pct=0.4)
```

```
neuron_dmso1_mirin <- FindMarkers(integrated_data, ident.1 =
"neuron_uninf_dmso_exp1_fastq", ident.2 = "neuron_uninf_mirin_exp4_fastq",
verbose = FALSE, min.pct=0.4)

neuron_dmso2_mirin <- FindMarkers(integrated_data, ident.1 =
"neuron_uninf_dmso_exp4_fastq", ident.2 = "neuron_uninf_mirin_exp4_fastq",
verbose = FALSE, min.pct=0.4)

glial_dmso1_mirin <- FindMarkers(integrated_data, ident.1 =
"glial_uninf_dmso_exp1_fastq", ident.2 = "glial_uninf_mirin_exp4_fastq",
verbose = FALSE, min.pct=0.4)

glial_dmso2_mirin <- FindMarkers(integrated_data, ident.1 =
"glial_uninf_dmso_exp4_fastq", ident.2 = "glial_uninf_mirin_exp4_fastq",
verbose = FALSE, min.pct=0.4)

endothelial_dmso1_mirin <- FindMarkers(integrated_data, ident.1 =
"endothelial_uninf_dmso_exp1_fastq", ident.2 =
"endothelial_uninf_dmso_exp4_fastq", verbose = FALSE, min.pct=0.4)

endothelial_dmso2_mirin <- FindMarkers(integrated_data, ident.1 =
"endothelial_uninf_dmso_exp4_fastq", ident.2 =
"endothelial_uninf_mirin_exp4_fastq", verbose = FALSE, min.pct=0.4)

schwann_dmso1_mirin <- FindMarkers(integrated_data, ident.1 =
"schwann_uninf_dmso_exp1_fastq", ident.2 = "schwann_uninf_mirin_exp4_fastq",
```

```
verbose = FALSE, min.pct=0.4)

schwann_dmso2_mirin <- FindMarkers(integrated_data, ident.1 =
"schwann_uninf_dmso_exp4_fastq", ident.2 = "schwann_uninf_mirin_exp4_fastq",
verbose = FALSE, min.pct=0.4)
```

Getting Cell Count for Conditon and Cell Type

```
table(integrated_data@meta.data$stim, integrated_data@meta.data$celltype)
```

exp	neuron	glial	endothelial	schwann
uninf_dmso_exp1_fastq	1444	1075	437	147
uninf_dmso_exp4_fastq	1487	927	203	128
uninf_ly_exp4_fastq	717	638	207	87
uninf_mirin_exp4_fastq	1073	696	227	81

Getting Up / Down Regulated Genes of Each Cell Type of LY Treatment vs First Control

```
treated_neuron_ly <- neuron_dmso1_ly %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC < 0) %>%
  column_to_rownames('gene')
# treated_neuron_ly
control_neuron_ly <- neuron_dmso1_ly %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC > 0) %>%
  column_to_rownames('gene')
# control_neuron_ly

treated_glial_ly <- glial_dmso1_ly %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC < 0) %>%
  column_to_rownames('gene')
# treated_neuron_ly
control_glial_ly <- glial_dmso1_ly %>%
```

```
  rownames_to_column('gene') %>%
  filter(avg_logFC > 0) %>%
  column_to_rownames('gene')
# control_neuron_ly

treated_endothelial_ly <- endothelial_dmso1_ly %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC < 0) %>%
  column_to_rownames('gene')
# treated_neuron_ly
control_endothelial_ly <- endothelial_dmso1_ly %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC > 0) %>%
  column_to_rownames('gene')
# control_neuron_ly

treated_schwann_ly <- schwann_dmso1_ly %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC < 0) %>%
  column_to_rownames('gene')
# treated_neuron_ly
control_schwanna_ly <- schwann_dmso1_ly %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC > 0) %>%
  column_to_rownames('gene')
# control_neuron_ly
```

Getting Top Differentialy Expressed Genes

```
ly_control <- c(rownames(control_neuron_ly)[1:2], rownames(control_glial_ly)
[1:2], rownames(control_endothelial_ly)[1:2], rownames(control_schwanna_ly)
[1:2])
ly_control
```

“Tuba1a” “Tuba1b” “Apoe” “Prdx1” “Gapdh” “Ifitm3” “Npc2” “Prdx1”

```
ly_treated <- c(rownames(treated_neuron_ly)[1:2], rownames(treated_glial_ly)
[1:2], rownames(treated_endothelial_ly)[1:2], rownames(treated_schwann_ly)
[1:2])
ly_treated
```

“Stfa2l1” “Hist1h4b” “Th” “Rsrp1” “Ret” “Atp1b1” “Eef2” “Eef1a1”

Getting Up / Down Regulated Genes of Each Cell Type of Mirin Treatment vs First Control

```
treated_neuron_mirin <- neuron_dmso1_mirin %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC < 0) %>%
  column_to_rownames('gene')
# treated_neuron_mirin
control_neuron_mirin <- neuron_dmso1_mirin %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC > 0) %>%
  column_to_rownames('gene')
# control_neuron_mirin

treated_glial_mirin <- glial_dmso1_mirin %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC < 0) %>%
  column_to_rownames('gene')
# treated_neuron_mirin
control_glial_mirin <- glial_dmso1_mirin %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC > 0) %>%
  column_to_rownames('gene')
# control_neuron_mirin

treated_endothelial_mirin <- endothelial_dmso1_mirin %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC < 0) %>%
  column_to_rownames('gene')
# treated_neuron_mirin
control_endothelial_mirin <- endothelial_dmso1_mirin %>%
```

```
  rownames_to_column('gene') %>%
  filter(avg_logFC > 0) %>%
  column_to_rownames('gene')
# control_neuron_mirin

treated_schwann_mirin <- schwann_dmso1_mirin %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC < 0) %>%
  column_to_rownames('gene')
# treated_neuron_mirin
control_schwanna_mirin <- schwann_dmso1_mirin %>%
  rownames_to_column('gene') %>%
  filter(avg_logFC > 0) %>%
  column_to_rownames('gene')
# control_neuron_mirin
```

Getting Top Differentialy Expressed Genes

```
mirin_control <- c(rownames(control_neuron_mirin)[1:2],
rownames(control_glial_mirin)[1:2], rownames(control_endothelial_mirin)[1:2],
rownames(control_schwanna_mirin)[1:2])
mirin_control
```

"Elavl2" "Bri3" "Rps27l" "Bri3" "Fam111a" "Sod2" "Gng11" "Nudt4"

```
mirin_treated <- c(rownames(treated_neuron_mirin)[1:2],
rownames(treated_glial_mirin)[1:2], rownames(treated_endothelial_mirin)[1:2],
rownames(treated_schwann_mirin)[1:2])
mirin_treated
```

"Slc10a1" "Pycard" "Rpl9" "Eef2" "Atp1b1" "Ret" "Cpm" "Gtse1"

Getting List of Genes to search up Pathways

```
```

```
# neuron_dmso1_ly, glial_dmso1_ly, endothelial_dmso1_ly, schwann_dmso1_ly

# neuron_dmso1_mirin, glial_dmso1_mirin, endothelial_dmso1_mirin,
schwann_dmso1_mirin

list_genes <- c(rownames(schwann_dmso1_mirin))
paste(list_genes,collapse=", ")
```

#appliedsequencinginformatics