UMAP on murine fetal brain 1k cells

Nur-Taz Rahman, PhD

Feb 12, 2020

####The clustering and annotation of clusters shown here are only to illustrate how to use the Seurat package as a tool to analyze and visualize single-cell RNA-seq datasets. The analyses presented here are not meant to be used/considered as accurate representations of clustering and cluster annotations of any kind neurological/biological data. The results shown in these analyses are purely for the purpose of exercise and may nor may not have biological relevance.

Loading necessary packages

```
library (dplyr, quietly = TRUE)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library (Seurat)
```

Direct R to the directory you want to work in

```
setwd ("/Users/nr267/Desktop/All Work/Classes/Spring 2020/Seurat_class_resources/")
```

Load data and carry out QC

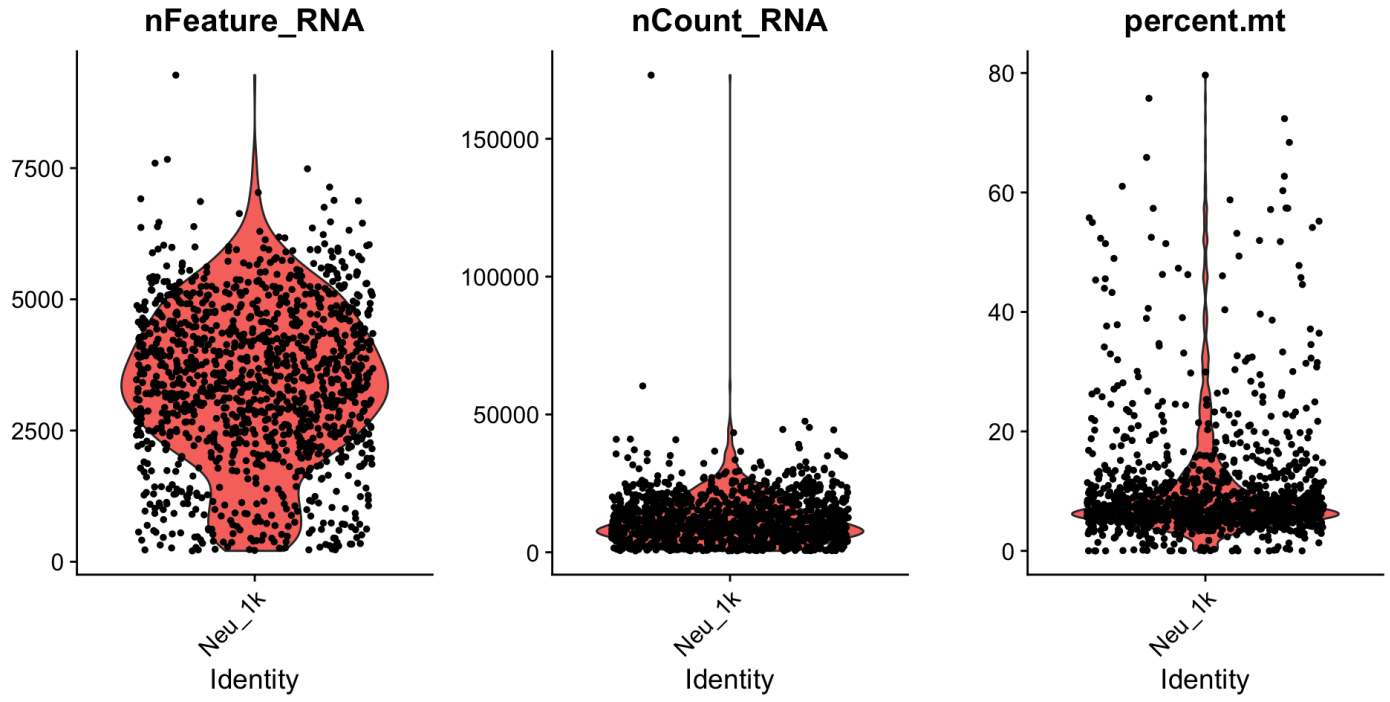
```
Neu_1k <- Read10X(data.dir = "../Seurat_class_resources/filtered_feature_bc_matrix_1K_ne
uron/1K_mouse_brain_cells/")
Neu_1k <- CreateSeuratObject(counts = Neu_1k, project = "Neu_1k", min.cells = 3, min.fea
tures = 200)

Neu_1k
```

```
## An object of class Seurat
## 15657 features across 1243 samples within 1 assay
## Active assay: RNA (15657 features)
```

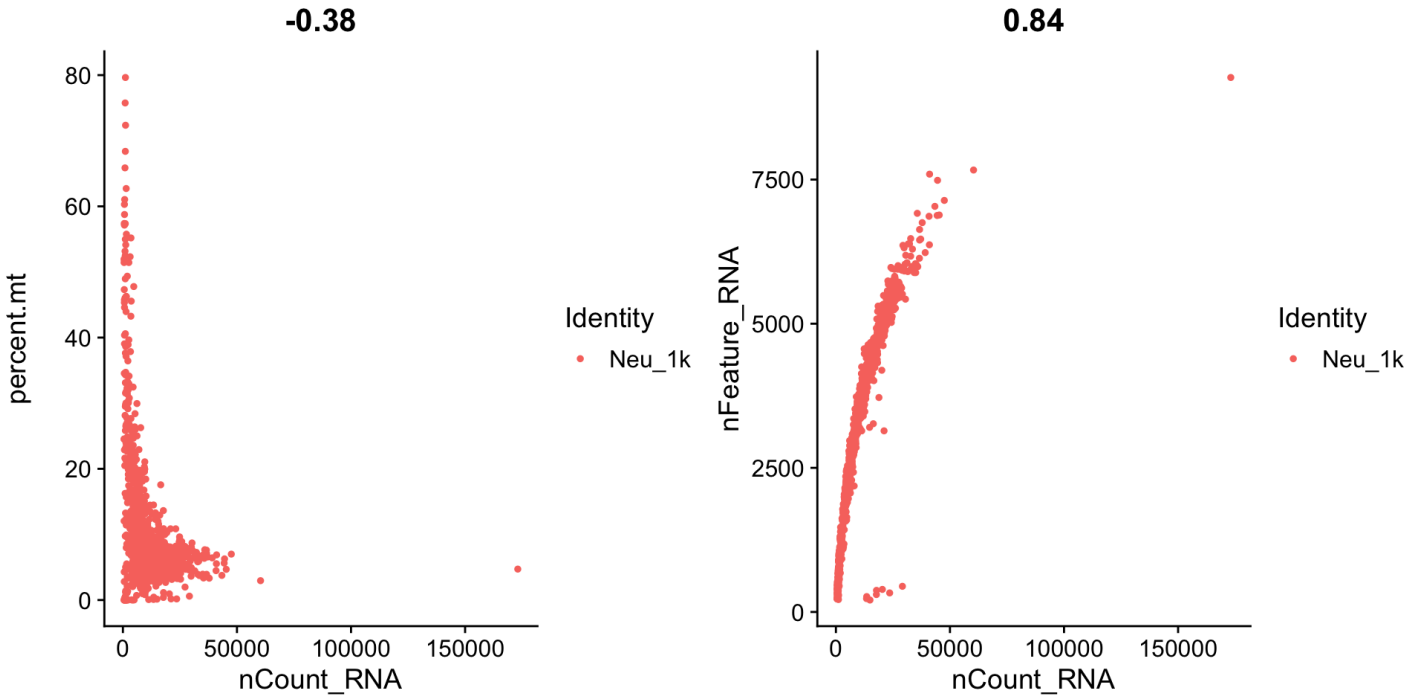
```
Neu_1k[["percent.mt"]] <- PercentageFeatureSet(Neu_1k, pattern = "^mt-")

head (Neu_1k@meta.data, 10)
```

```
##                  orig.ident nCount_RNA nFeature_RNA percent.mt
## AAACGAATCAAAGCCT     Neu_1k       1023          417  52.492669
## AAACGCTGTAATGTGA     Neu_1k       7209         2743  12.109863
## AAACGCTGTCCTGGGT     Neu_1k       9881         3601   7.600445
## AAAGAACCAGGACATG     Neu_1k       5289         2385   5.483078
## AAAGGTACACACGGTC     Neu_1k      18674         4746   5.515690
## AAAGTCCAGTCACTAC     Neu_1k      20189         5097   5.735797
## AAAGTCCTCCAGCCTT     Neu_1k       6447         2557  12.936249
## AAAGTGAGTTCCTAAG     Neu_1k      16836         4710   4.793300
## AAAGTGATCAGTGGGA     Neu_1k       2063         1007  32.670868
## AAATGGAGTAGCGTCC     Neu_1k       1946          924  31.397739
```

```
VlnPlot (Neu_1k, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
```




```
plot1 <- FeatureScatter(Neu_1k, feature1 = "nCount_RNA", feature2 = "percent.mt")
plot2 <- FeatureScatter(Neu_1k, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")

CombinePlots(plots = list(plot1, plot2))
```



Pre-process data

```
Neu_1k <- subset(Neu_1k, subset = nFeature_RNA > 200 & nFeature_RNA < 6000 & percent.mt
 < 10)
```

Check data dimensions after subsetting

```
Neu_1k
```

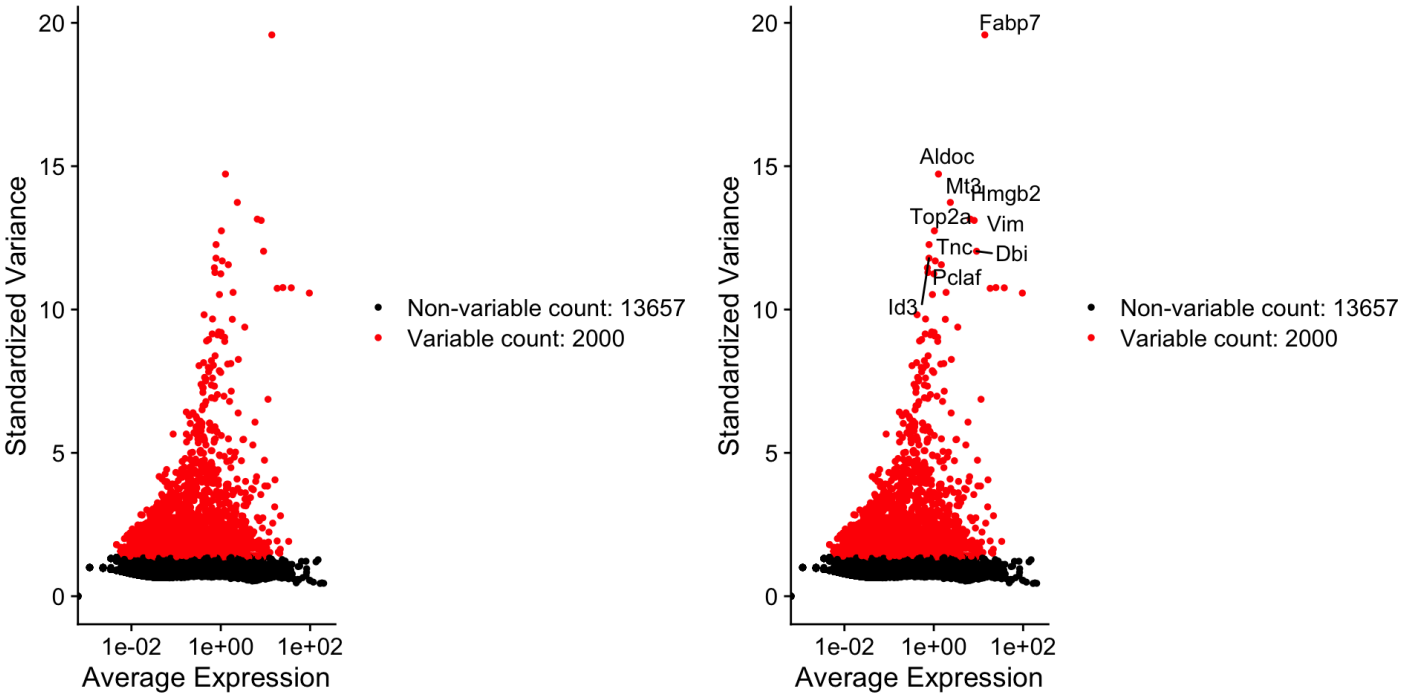
```
## An object of class Seurat
## 15657 features across 862 samples within 1 assay
## Active assay: RNA (15657 features)
```

LogNormalize Data (normalize count data per cell and transform to log scale)

```
Neu_1k <- NormalizeData(Neu_1k, verbose = TRUE)
```

Pre-process data for PCA

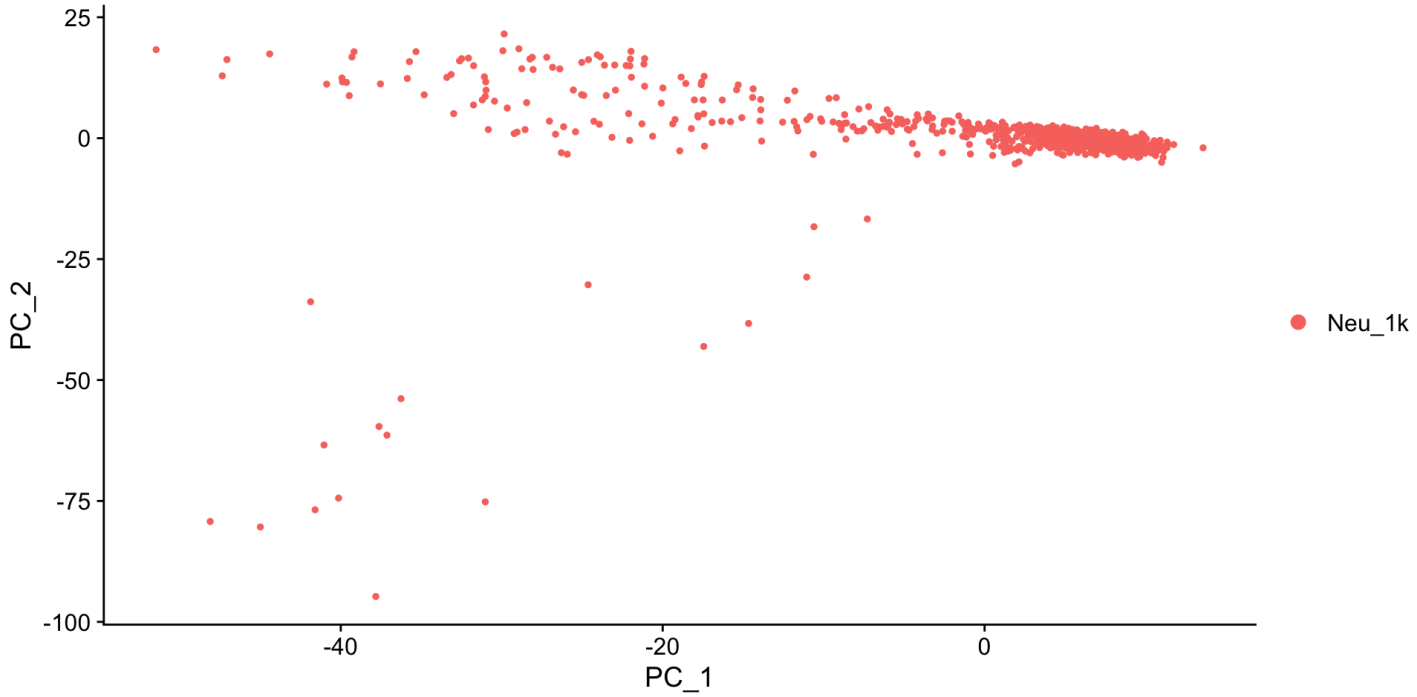
```
Neu_1k <- FindVariableFeatures(Neu_1k, selection.method = "vst", nfeatures = 2000)

top10 <- head(VariableFeatures(Neu_1k), 10)

plot3 <- VariableFeaturePlot(Neu_1k)

plot4 <- LabelPoints(plot = plot3, points = top10, repel = TRUE)

CombinePlots(plots = list(plot3, plot4))
```



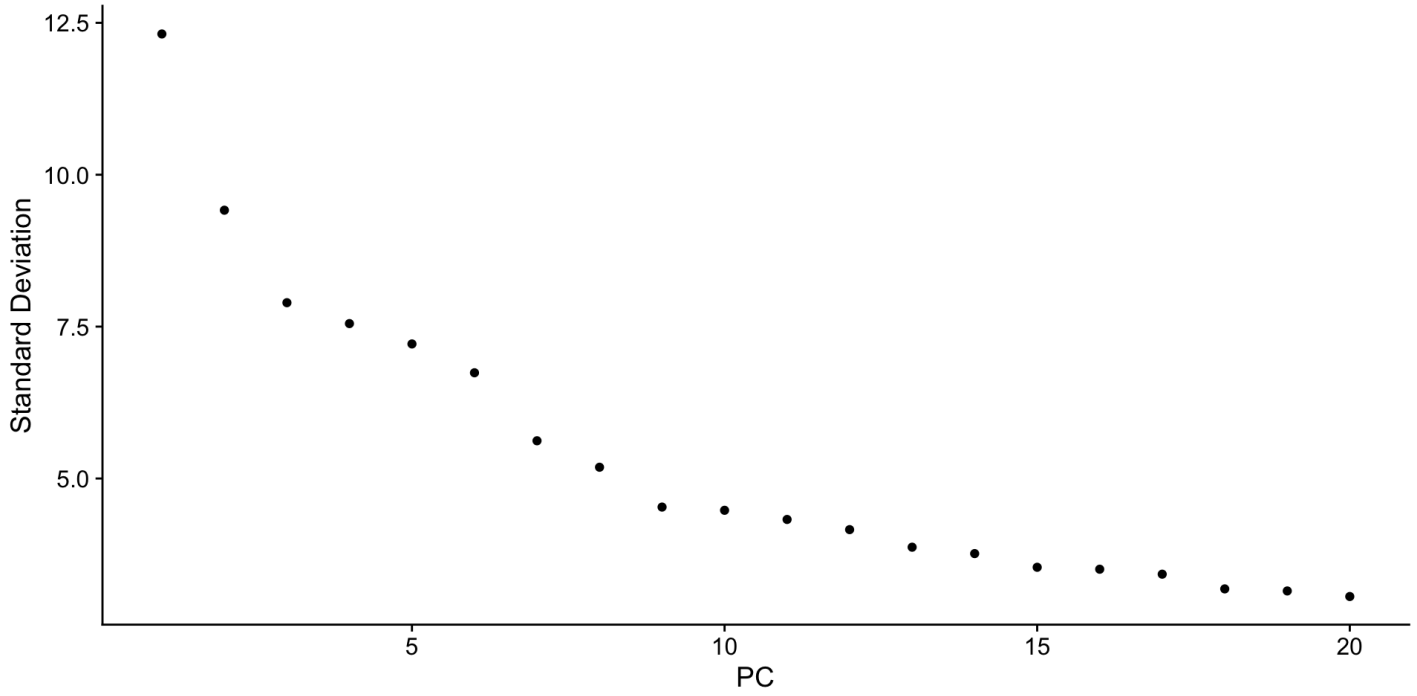

```
all.genes <- rownames(Neu_1k)

Neu_1k <- ScaleData(Neu_1k, features = all.genes)

Neu_1k
```

```
## An object of class Seurat
## 15657 features across 862 samples within 1 assay
## Active assay: RNA (15657 features)
```

```
Neu_1k<- RunPCA(Neu_1k, features = VariableFeatures(object = Neu_1k))

DimPlot (Neu_1k, reduction = "pca")
```

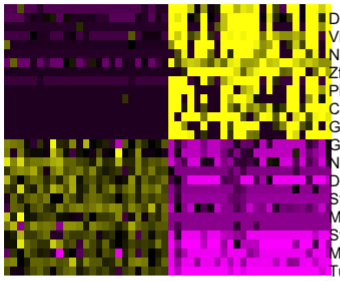



```
ElbowPlot(Neu_1k)
```

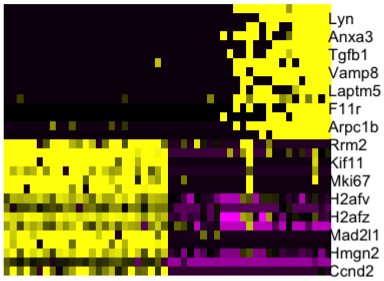



```
DimHeatmap (Neu_1k, dims = 1:15, cells = 50, balanced = TRUE)
```

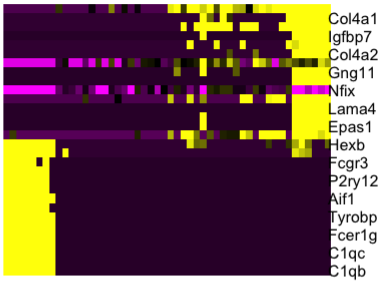
PC_1



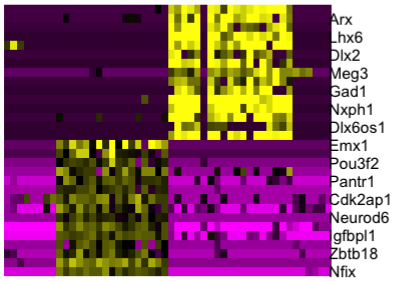
PC_2



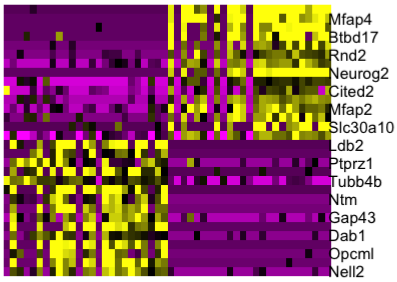
PC_3



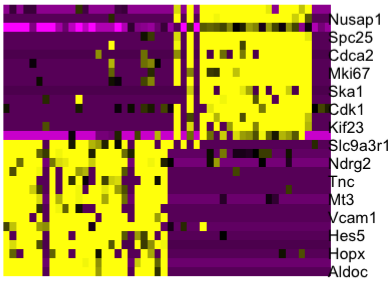
PC_4



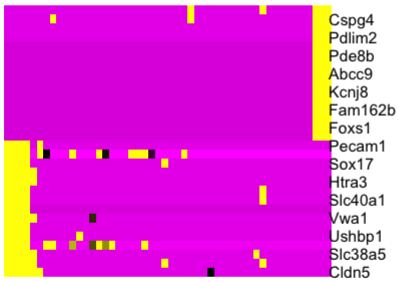
PC_5



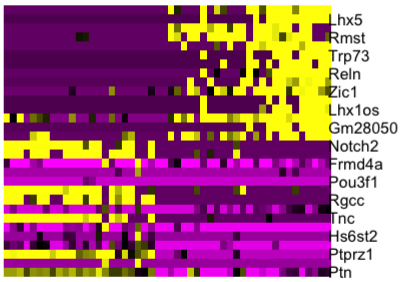
PC_6



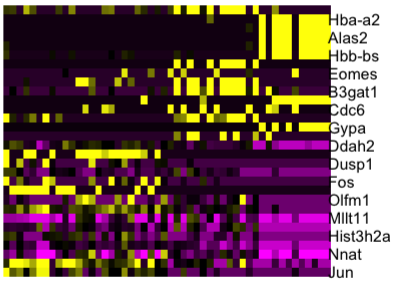
PC_7



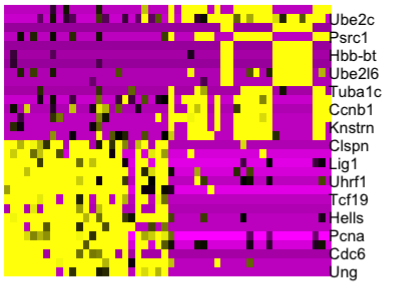
PC_8



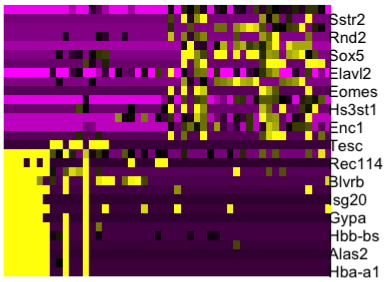
PC_9



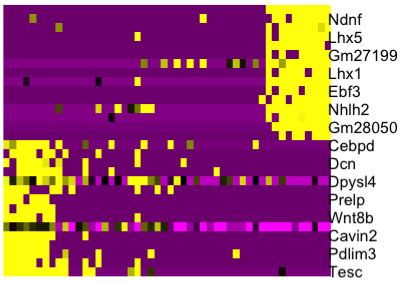
PC_10



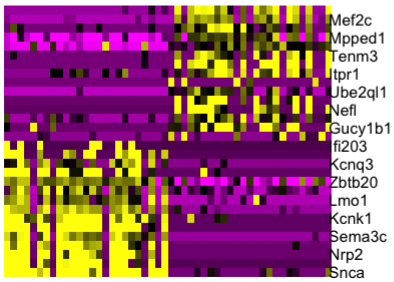
PC_11



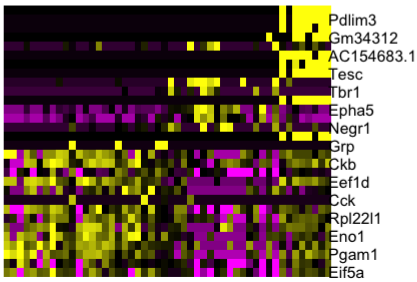
PC_12



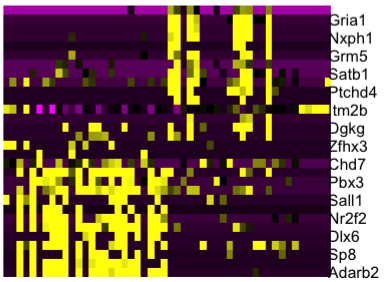
PC_13



PC_14



PC_15



Run UMAP on dataset

```
Neu_1k <- FindNeighbors(Neu_1k, dims = 1:15)
```

```
## Computing nearest neighbor graph
```

```
## Computing SNN
```

```
Neu_1k <- FindClusters (Neu_1k, resolution = 0.5)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 862
## Number of edges: 23522
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8617
## Number of communities: 7
## Elapsed time: 0 seconds
```

```
Neu_1k <- RunUMAP(Neu_1k, dims = 1:15)
```

```
## Warning: The default method for RunUMAP has changed from calling Python UMAP via reti
culate to the R-native UWOT using the cosine metric
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'cor
relation'
## This message will be shown once per session
```

```
## 00:28:08 UMAP embedding parameters a = 0.9922 b = 1.112
```

```
## 00:28:08 Read 862 rows and found 15 numeric columns
```

```
## 00:28:08 Using Annoy for neighbor search, n_neighbors = 30
```

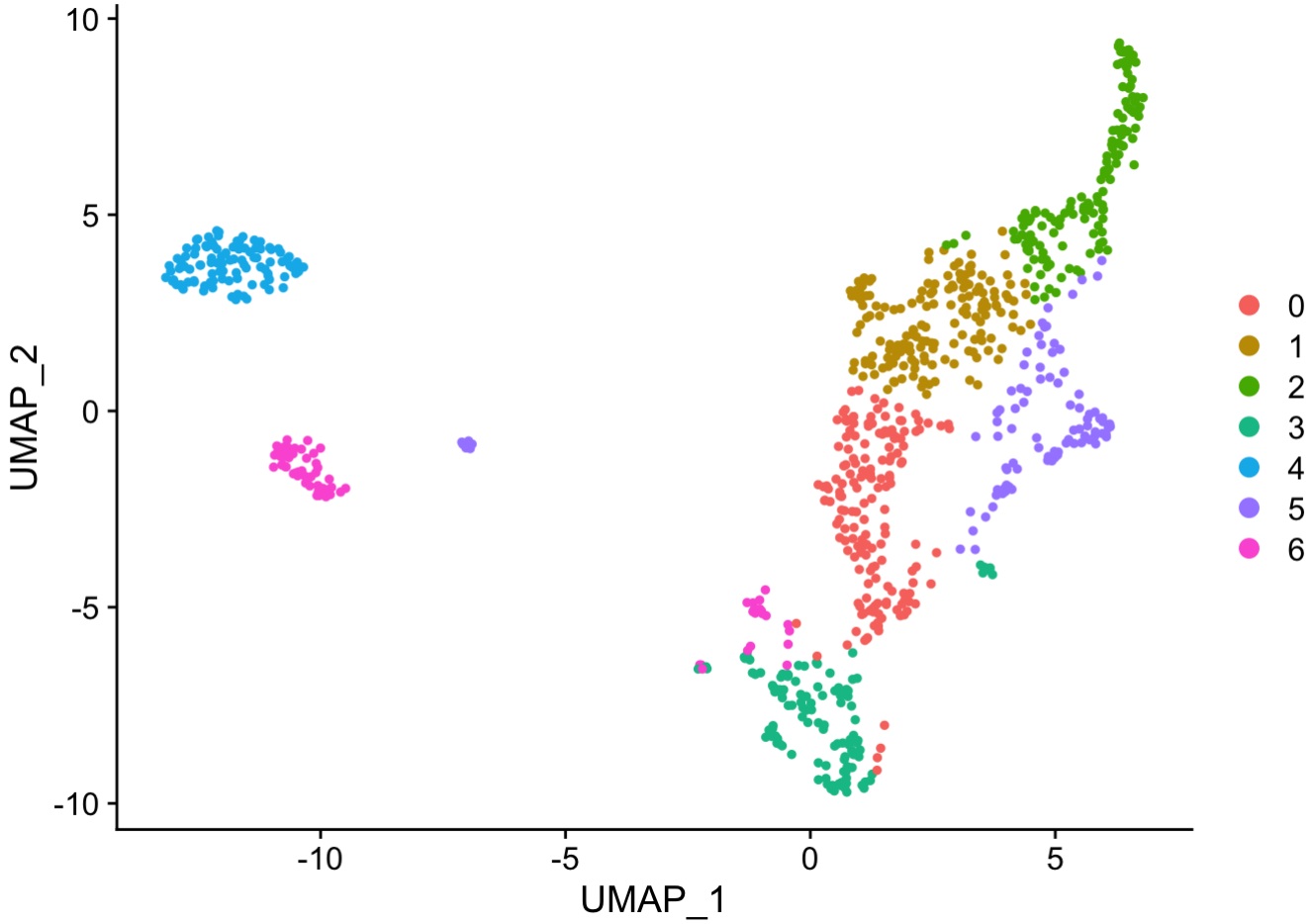
```
## 00:28:08 Building Annoy index with metric = cosine, n_trees = 50
```

```
## 0%   10   20   30   40   50   60   70   80   90   100%
```

```
## [----|----|----|----|----|----|----|----|----|----|
```

```
## ************************************************|
## 00:28:09 Writing NN index file to temp file /var/folders/s9/6k4hffrx25gb0_43dcj132p15
4t8k1/T//RtmpQzUT6j/file36825e754a8f
## 00:28:09 Searching Annoy index using 1 thread, search_k = 3000
## 00:28:09 Annoy recall = 100%
## 00:28:09 Commencing smooth kNN distance calibration using 1 thread
## 00:28:09 Initializing from normalized Laplacian + noise
## 00:28:09 Commencing optimization for 500 epochs, with 30606 positive edges
## 00:28:10 Optimization finished
```

```
DimPlot (Neu_1k, reduction = "umap")
```



Find all markers of each cluster to identify them

```
cluster0.markers <- FindMarkers (Neu_1k, ident.1 = 0, min.pct = 0.5)
cluster1.markers <- FindMarkers (Neu_1k, ident.1 = 1, min.pct = 0.5)
cluster2.markers <- FindMarkers (Neu_1k, ident.1 = 2, min.pct = 0.5)
cluster3.markers <- FindMarkers (Neu_1k, ident.1 = 3, min.pct = 0.5)
cluster4.markers <- FindMarkers (Neu_1k, ident.1 = 4, min.pct = 0.5)
cluster5.markers <- FindMarkers (Neu_1k, ident.1 = 5, min.pct = 0.5)
cluster6.markers <- FindMarkers (Neu_1k, ident.1 = 6, min.pct = 0.5)
```

To view the genes lists on your RStudio console

View(cluster0.markers)
View(cluster1.markers)
View(cluster2.markers)
View(cluster3.markers)
View(cluster4.markers)
View(cluster5.markers)
View(cluster6.markers)

To “export” the genes lists to your current directory

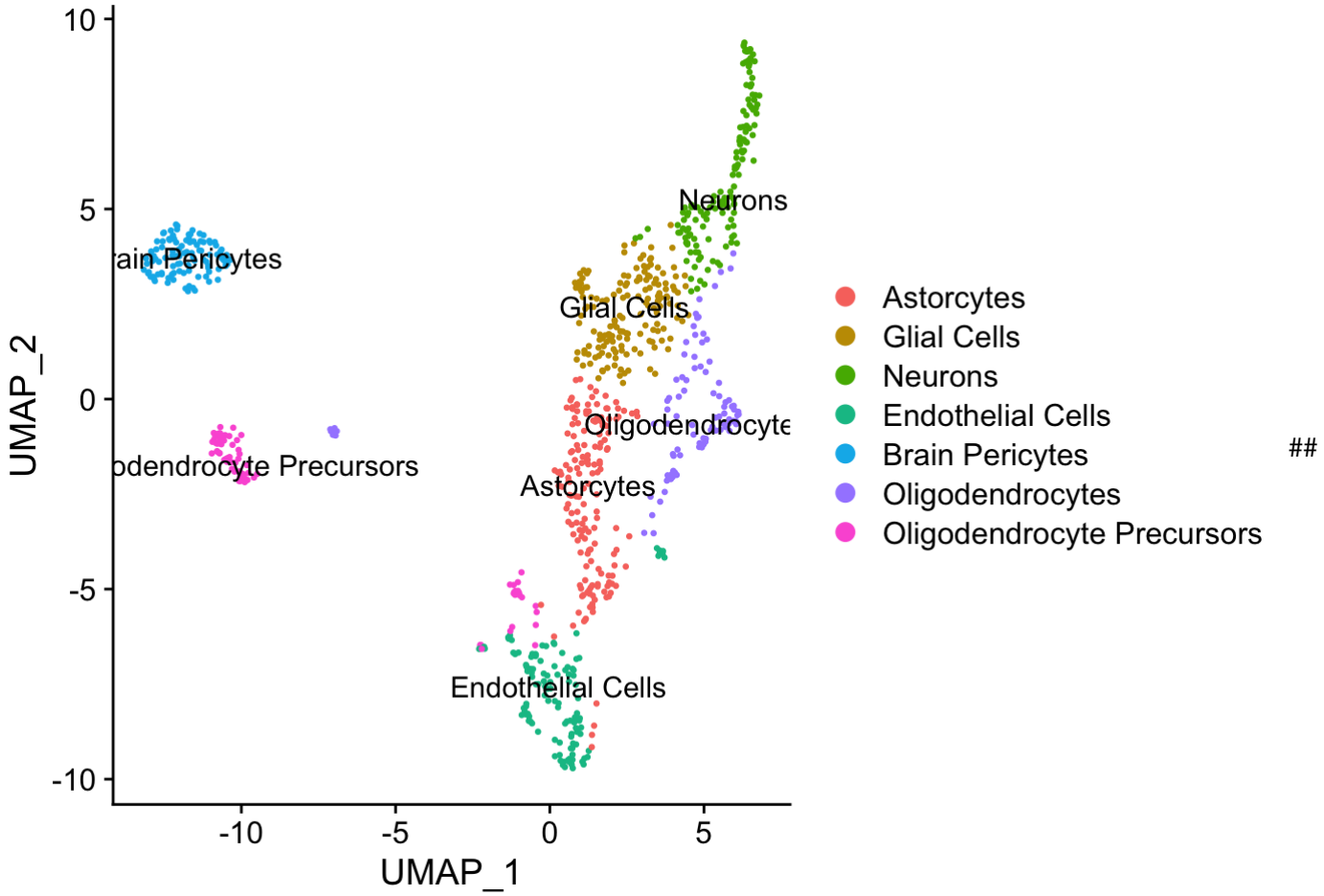
write.csv(cluster0.markers, “cluster0_markers_UMAP”)
write.csv(cluster1.markers, “cluster1_markers_UMAP”)
write.csv(cluster2.markers, “cluster2_markers_UMAP”)
write.csv(cluster3.markers, “cluster3_markers_UMAP”)
write.csv(cluster4.markers, “cluster4_markers_UMAP”)
write.csv(cluster5.markers, “cluster5_markers_UMAP”)
write.csv(cluster6.markers, “cluster6_markers_UMAP”)

Find genes distinguishing one cluster from other specific clusters

```
green_cluster.markers <- FindMarkers (Neu_1k, ident.1 = 2, ident.2 = c(0, 1), min.pct = 0.25)
```

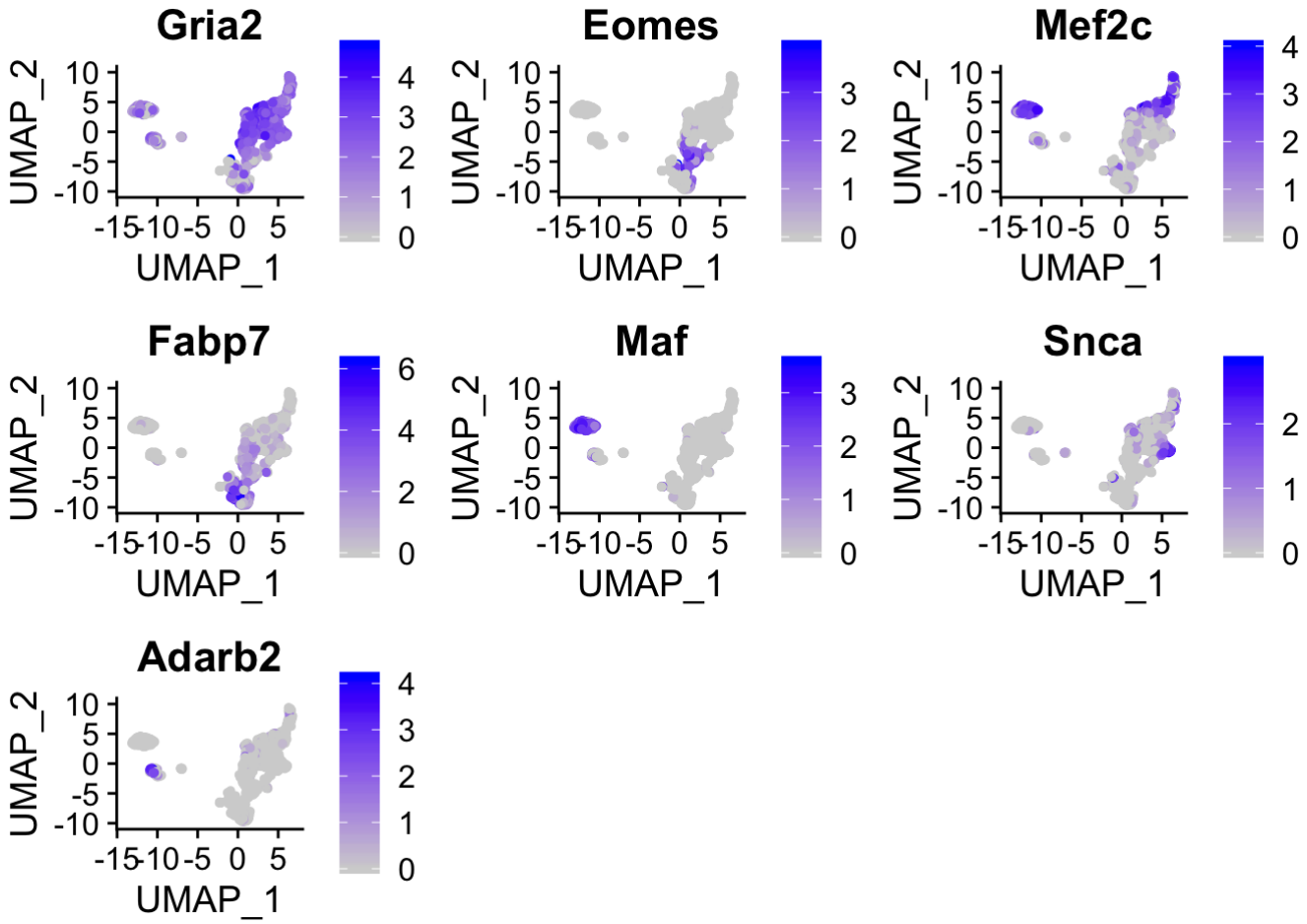
Label UMAP clusters with ids

```
new.cluster.ids <- c("Astorcytes", "Glial Cells", "Neurons", "Endothelial Cells", "Brain Pericytes", "Oligodendrocytes", "Oligodendrocyte Precursors")
names(new.cluster.ids) <- levels (Neu_1k)
Neu_1k <- RenameIdents (Neu_1k, new.cluster.ids)
DimPlot (Neu_1k, reduction = "umap", label = TRUE, pt.size = 0.5)
```



Visualize feature expression

```
FeaturePlot(Neu_1k, features = c("Gria2", "Eomes", "Mef2c", "Fabp7", "Maf", "Snca", "Adarb2"))
```


Gria2
Eomes
Mef2c
Fabp7
Maf
Snca
Adarb2
UMAP_2
UMAP_1
10
5
0
-5
-10
-15
-10
-5
0
5
4
3
2
1
0
6