GSE1338852.RData

Adithya Ayanam

7/10/2020

```
library(Matrix)
library(dplyr)
```

```
##
## 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(stringi)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

```
library(Seurat)
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.0.2
```

```
 ls()
```

```
## character(0)
```

```
load("C:/Lab08-Data-Wrangling-scRNAseq/GSE138852/data_GSE138852.rdata")
```

#nCount_RNA: number of UMI per cell #nFeature_RNA: number of genes detected per cell.

```
head(GSE_seurat@meta.data)
```

```
##                         orig.ident nCount_RNA nFeature_RNA
## AAACCTGGTAGAAAGG_AD5_AD6        GSE        760           564
## AAACCTGGTAGCGATG_AD5_AD6        GSE        720           527
## AAACCTGTCAGTCAGT_AD5_AD6        GSE       1209           773
## AAACCTGTCCAAACAC_AD5_AD6        GSE        925           615
## AAACCTGTCCAGTATG_AD5_AD6        GSE        562           434
## AAAGCAACATGGGAAC_AD5_AD6        GSE        367           315
```

```
tail(GSE_seurat@meta.data)
```

```
##                         orig.ident nCount_RNA nFeature_RNA
## TTTGGTTTCCCAGGTG_AD1_AD2        GSE        435           359
## TTTGGTTTCCGTACAA_AD1_AD2        GSE        442           339
## TTTGTCACAAGCCATT_AD1_AD2        GSE        532           459
## TTTGTCAGTATAGGTA_AD1_AD2        GSE        383           330
## TTTGTCATCCACTGGG_AD1_AD2        GSE        868           603
## TTTGTCATCCGGGTGT_AD1_AD2        GSE        449           369
```

```
#Mitochondrial Ratio  gives the metric of cell reads from mitochondrial gene
GSE_seurat$mitoRatio <- PercentageFeatureSet(object=GSE_seurat, pattern="^MT-")
 GSE_seurat$mitoRatio <-GSE_seurat@meta.data$mitoRatio/100



#Returns the first  parts of a vector, matrix, table, data frame or function.
#Rename of Columns in the Seurat object
GSE_seurat$nUMI <-GSE_seurat@meta.data$nCount_RNA
GSE_seurat$nGenes <-GSE_seurat@meta.data$nFeature_RNA

# remove after rename
GSE_seurat$nCount_RNA <- NULL
GSE_seurat$nFeature_RNA <-NULL

# create sample column
GSE_seurat$sample<-NA
GSE_seurat$cells<-rownames(GSE_seurat@meta.data)

#calculate no of Genes detected per UMI - more genes detected per UMI  = more complex data)
GSE_seurat$log10GenesPerUMI <-log10(GSE_seurat$nGenes)/log10(GSE_seurat$nUMI)
library(stringr)
GSE_seurat$sample[which(str_detect(GSE_seurat$cells, "^Sample"))] <-"Sample"

#
# GSE_seurat@meta.data <-GSE_seurat
View(head(GSE_seurat@meta.data,1000))
save(GSE_seurat, file="C:/Lab08-Data-Wrangling-scRNAseq/GSE138852/data/GSE_seurat_filtered.RData")
```

```
#Filter out low quality breads using selected thresholds
 GSE_filtered <- subset(x = GSE_seurat,subset=(nUMI>=500)&(nGenes>=250)&(log10GenesPerUMI>0.80)&(mitoRatio<0
.20))


#Output logical vector for every gene on whether the more than 0 counts per cell.Remove genes that have 0 ex
pressions in all cells.
 counts <- GetAssayData(object = GSE_filtered,slot = "counts")
nonzero <- counts > 0
keep_genes <- Matrix::rowSums(nonzero)>=10
filtered_counts <- counts[keep_genes, ]
GSE_filtered <- CreateSeuratObject(filtered_counts, meta.data = GSE_filtered@meta.data)


#Save filtered subset to new metadata
GSE_clean <- GSE_filtered@meta.data
save(GSE_filtered, file="C:/Lab08-Data-Wrangling-scRNAseq/GSE138852/data/GSEfiltered.RData")
```

```
#Normalize the counts
GSE_seurat <- NormalizeData(GSE_filtered)
```

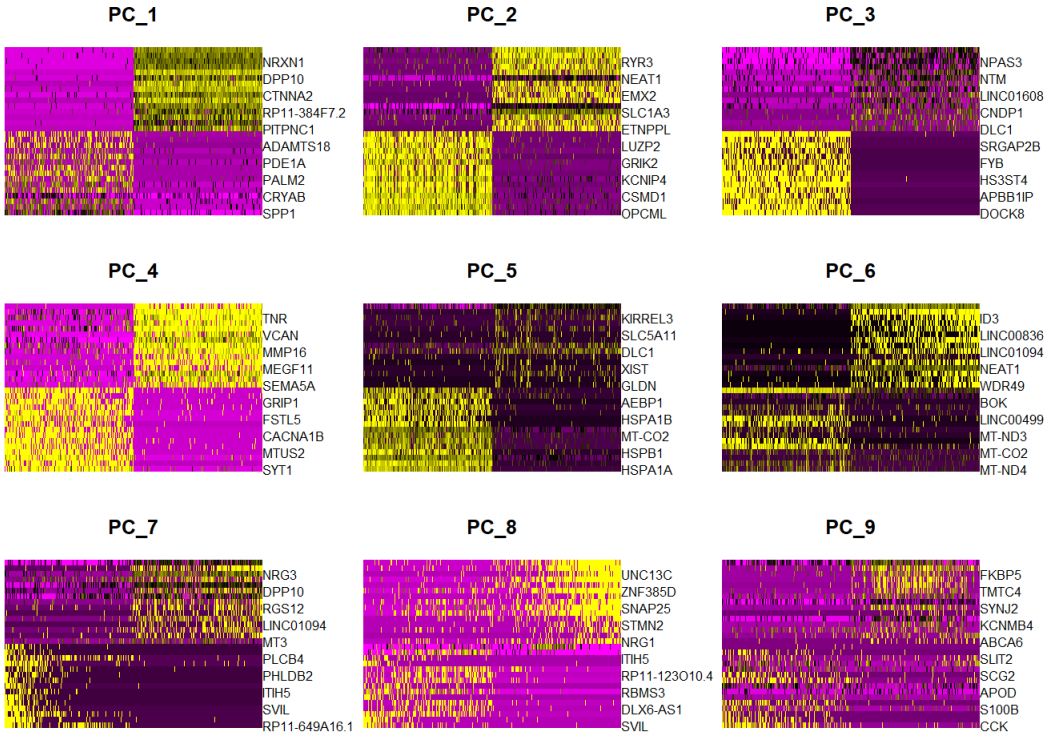
```
#Find Variable Genes and Scale Data

GSE_seurat <- FindVariableFeatures(object = GSE_seurat,selection.method = "vst", nfeatures = 2000)

all.genes <- rownames(GSE_seurat)
GSE_seurat <- ScaleData(GSE_seurat, features = all.genes)
```

```
## Centering and scaling data matrix
```

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

```
## PC_ 1
## Positive:  SPP1, LINC01608, CNDP1, CRYAB, KCNMB4, KIRREL3, PALM2, SYNJ2, CDH19, PDE1A
##     SLC5A11, CNTNAP4, ADAMTS18, RP11-81H3.2, TMTC4, GLDN, RP11-267C16.1, LRP2, APOD, VRK2
##     KLHL4, LINC01170, CHN2, LINGO1, BOK, HS3ST5, RP11-50D16.4, CD55, ST6GALNAC3, LAMA2
## Negative:  GPM6A, NRG3, NRXN1, RORA, SLC1A2, DPP10, NKAIN3, GPC5, CTNNA2, ADGRV1
##     DTNA, RP11-384F7.2, NEBL, FAM155A, PITPNC1, RNF219-AS1, LSAMP, GABRB1, NRCAM, NTRK2
##     SPARCL1, TRPM3, FMN2, SOX5, PTPRZ1, RYR3, TENM2, ADCY2, CPE, CSGALNACT1
## PC_ 2
## Positive:  OPCML, FGF14, SNTG1, CSMD1, DSCAM, PCDH15, KCNIP4, TNR, LHFPL3, GRIK2
##     NXPH1, FGF12, LUZP2, ATRNL1, MMP16, LRRTM4, GRIK1, GRM7, CA10, CSMD3
##     RBFOX1, MDGA2, KCND2, SGCZ, RIMS2, GRID2, RP4-668E10.4, XKR4, GRM5, MEG3
## Negative:  ADGRV1, RNF219-AS1, RYR3, TPD52L1, AQP4, NEAT1, GLIS3, LINC00499, EMX2, BMPR1B
##     PTGDS, SLC1A3, PRKG1, LRRC16A, ETNPPL, RANBP3L, TRPM3, PAMR1, MALAT1, STON2
##     SFXN5, SLC14A1, SLC4A4, GPC5, ZNRF3, COL5A3, GLI3, AC002429.5, MT2A, SLC7A11
## PC_ 3
## Positive:  DOCK8, RP11-624C23.1, ADAM28, APBB1IP, LPAR6, ST6GAL1, HS3ST4, CD74, ATP8B4, FYB
##     SYK, PTPRC, SRGAP2B, TBXAS1, SRGAP2, SRGN, CSF2RA, A2M, INPP5D, MEF2C
##     SAMSN1, C10orf11, ARHGAP15, CSF3R, ARHGAP24, SLCO2B1, P2RY12, HLA-DRA, BLNK, CD86
## Negative:  PTGDS, ERBB4, NPAS3, LRP1B, CRYAB, NTM, APOD, LSAMP, LINC01608, CTNND2
##     NRXN3, CNDP1, KIRREL3, NOVA1, DLC1, PALM2, SYNJ2, KCNMB4, CDH19, HS3ST5
##     CNTNAP4, PDE1A, LINC01170, MT3, GALNT13, BOK, HSP90AA1, DNAJB2, AC012593.1, RP11-81H3.2
## PC_ 4
## Positive:  SYT1, KCNC2, ROBO2, MTUS2, LINGO2, KCNQ5, CACNA1B, SYNPR, CELF4, FSTL5
##     FRMPD4, GABRB2, GRIP1, WBSCR17, RP11-123O10.4, GRIN1, HCN1, DLX6-AS1, GRIN2A, KCNJ3
##     KCNH7, CDH9, GALNTL6, ADGRL2, GABBR2, CHRM3, GRIA1, DCLK1, CNTN5, PCLO
## Negative:  RP4-668E10.4, LHFPL3, TNR, XYLT1, PCDH15, VCAN, PTPRZ1, LUZP2, MMP16, PDZRN4
##     CA10, MEGF11, SOX6, BRINP3, SEMA5A, DSCAM, MIR3681HG, CHST11, KCNMB2-AS1, SMOC1
##     LRRC4C, NOVA1-AS1, COL9A1, COL11A1, NOVA1, DCC, STK32A, GPC6, CALCRL, PDGFRA
## PC_ 5
## Positive:  HSPA1A, GFAP, LINGO1, HSPB1, MT-ND4, MT-CO3, MT-CO2, CRYAB, DNAJB1, HSPA1B
##     BOK, MT-ATP6, AEBP1, DNAJB2, MT-CYB, MT3, HSP90AA1, MT-ND2, MT-ND3, MT-ND1
##     FKBP4, SLC26A3, FOS, RHPN1, UBC, DHCR24, MT-CO1, HSPH1, IFITM3, ITGB4
## Negative:  MALAT1, PDE1A, KIRREL3, LINC01608, CNTNAP4, SLC5A11, PALM2, NRXN3, DLC1, LINC01170
##     RP11-81H3.2, XIST, HS3ST5, RP11-267C16.1, GLDN, SYNJ2, ADAMTS18, WIF1, SLC6A1-AS1, CNTNAP2
##     NPAS3, TMTC4, AC012593.1, ESRRG, KCNQ1OT1, PIK3C2A, CNDP1, ROBO1, SLC4A8, GNA14-AS1
```

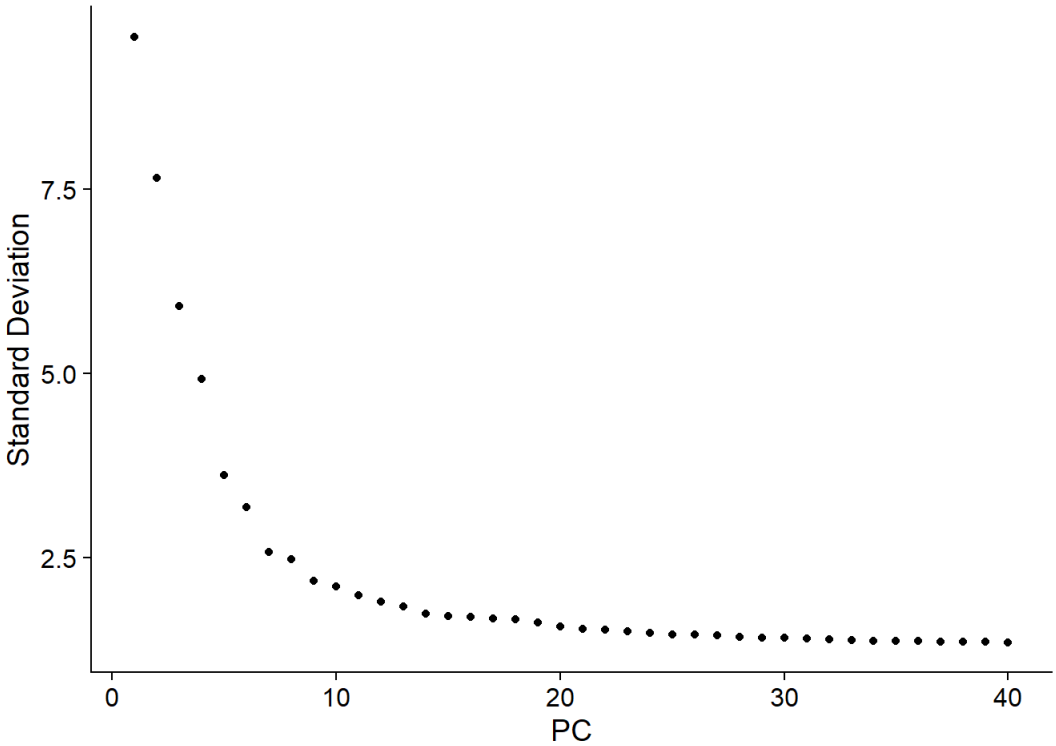
```
DimHeatmap(object = GSE_seurat,dims = 1:9,cells = 500, balanced = TRUE)
```




```
print(x = GSE_seurat[["pca"]],
      dims= 1:10,
      nfeatures = 5)
```

```
## PC_ 1
## Positive:  SPP1, LINC01608, CNDP1, CRYAB, KCNMB4
## Negative:  GPM6A, NRG3, NRXN1, RORA, SLC1A2
## PC_ 2
## Positive:  OPCML, FGF14, SNTG1, CSMD1, DSCAM
## Negative:  ADGRV1, RNF219-AS1, RYR3, TPD52L1, AQP4
## PC_ 3
## Positive:  DOCK8, RP11-624C23.1, ADAM28, APBB1IP, LPAR6
## Negative:  PTGDS, ERBB4, NPAS3, LRP1B, CRYAB
## PC_ 4
## Positive:  SYT1, KCNC2, ROBO2, MTUS2, LINGO2
## Negative:  RP4-668E10.4, LHFPL3, TNR, XYLT1, PCDH15
## PC_ 5
## Positive:  HSPA1A, GFAP, LINGO1, HSPB1, MT-ND4
## Negative:  MALAT1, PDE1A, KIRREL3, LINC01608, CNTNAP4
## PC_ 6
## Positive:  MT-ND4, MT-CO3, HSPA1A, MT-CO2, COL5A3
## Negative:  MALAT1, DCLK1, ID3, TNC, CD44
## PC_ 7
## Positive:  RP11-649A16.1, CEMIP, ARHGAP29, SVIL, SLC6A13
## Negative:  NPAS3, DCLK1, NRG3, GFAP, LSAMP
## PC_ 8
## Positive:  SVIL, CEMIP, KCNT2, DLX6-AS1, KCNC2
## Negative:  CADPS2, CDH18, UNC13C, RALYL, SH3GL2
## PC_ 9
## Positive:  CCK, CALB2, VIP, S100B, DLX6-AS1
## Negative:  NEAT1, XIST, FKBP5, SLC5A11, ADAMTS18
## PC_ 10
## Positive:  CALB2, VWC2L, VIP, HTR2C, ASIC2
## Negative:  PTCHD4, ST6GALNAC5, UNC5D, FGF13, ADGRL2
```

```
ElbowPlot(object = GSE_seurat, ndims=40)
```




```
#Find clusters, Differentially expressed genes and cluster biomarkers
GSE_seurat <- FindNeighbors(object = GSE_seurat, dims = 1:10)
```

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

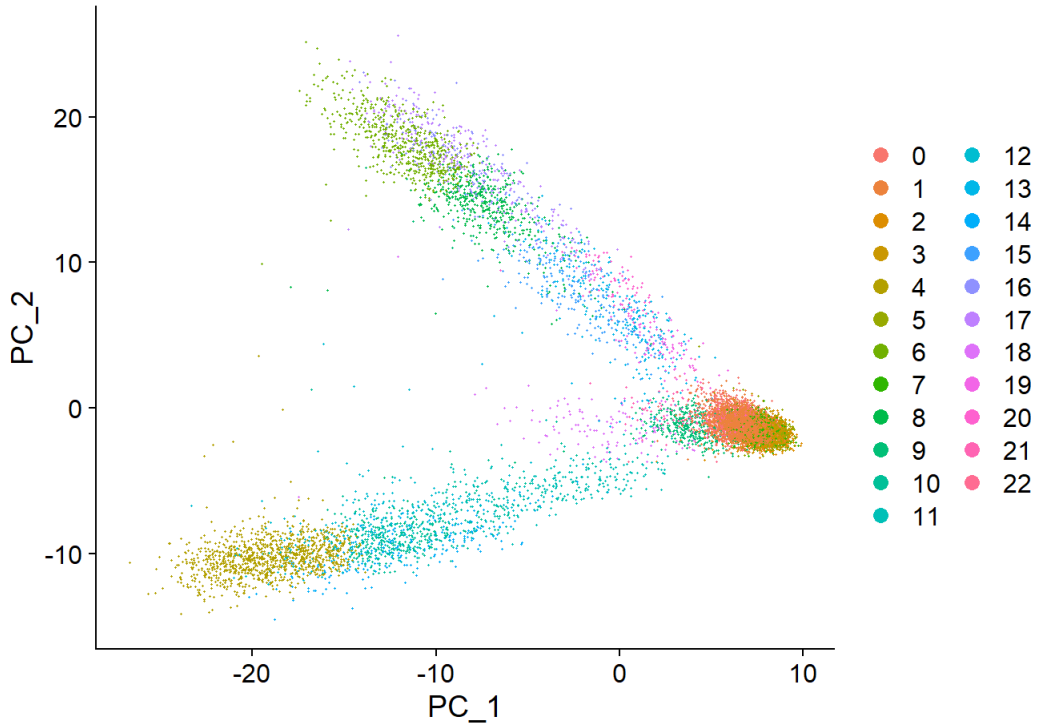
```
## Computing SNN
```

```
GSE_seurat <- FindClusters(object = GSE_seurat,resolution = c(0.4,0.6, 0.8, 1.0,1.4))
```

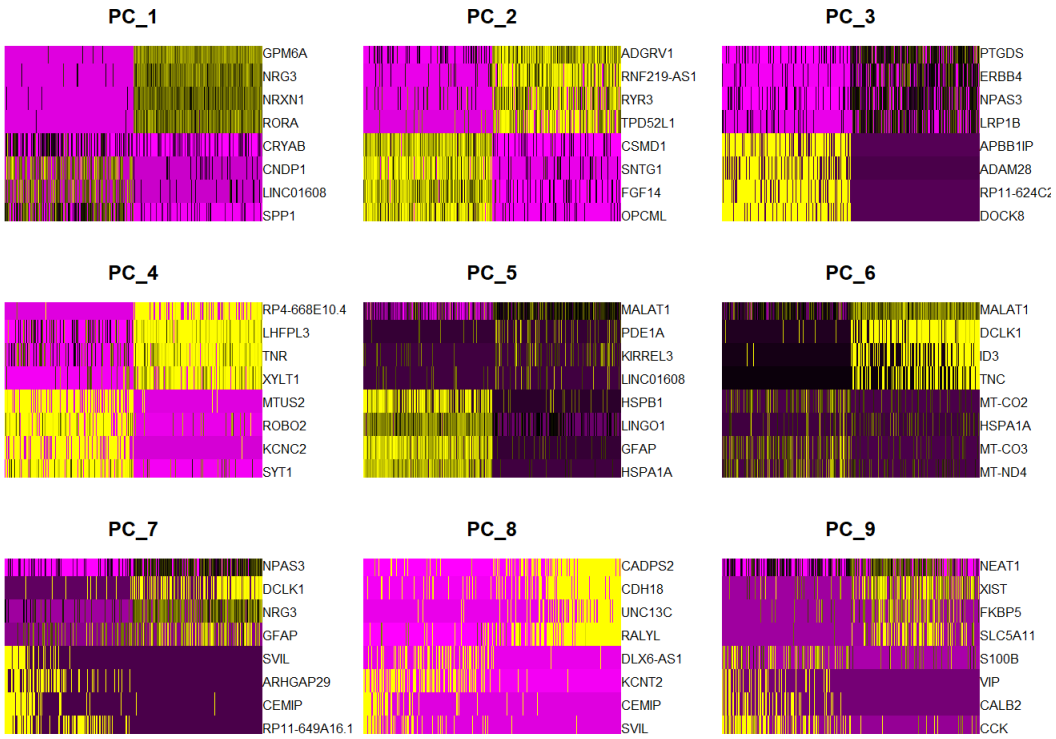
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 11503
## Number of edges: 381288
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9365
## Number of communities: 13
## Elapsed time: 1 seconds
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 11503
## Number of edges: 381288
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9136
## Number of communities: 13
## Elapsed time: 1 seconds
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 11503
## Number of edges: 381288
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8930
## Number of communities: 14
## Elapsed time: 1 seconds
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 11503
## Number of edges: 381288
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8771
## Number of communities: 22
## Elapsed time: 1 seconds
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 11503
## Number of edges: 381288
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8511
## Number of communities: 23
## Elapsed time: 1 seconds
```

```
#Explore resolutions
GSE_seurat@meta.data %>%
     View()
```

```
#Assign identity of clusters
DimPlot(object = GSE_seurat, reduction = "pca")
```




```
DimHeatmap(object = GSE_seurat,dims = 1:9,reduction = "pca", cells = 500,nfeatures = 8)
```




```
head(Idents(GSE_seurat), 5)
```

```
## AAACCTGGTAGAAAGG_AD5_AD6 AAACCTGGTAGCGATG_AD5_AD6 AAACCTGTCAGTCAGT_AD5_AD6
##                        7                        1                        7
## AAACCTGTCCAAACAC_AD5_AD6 AAACCTGTCCAGTATG_AD5_AD6
##                        3                        7
## Levels: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
```

```
GSE_seurat <- RunUMAP(GSE_seurat, dims = 1:10)
```

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

```
## 17:29:36 UMAP embedding parameters a = 0.9922 b = 1.112
```

```
## 17:29:36 Read 11503 rows and found 10 numeric columns
```

```
## 17:29:36 Using Annoy for neighbor search, n_neighbors = 30
```

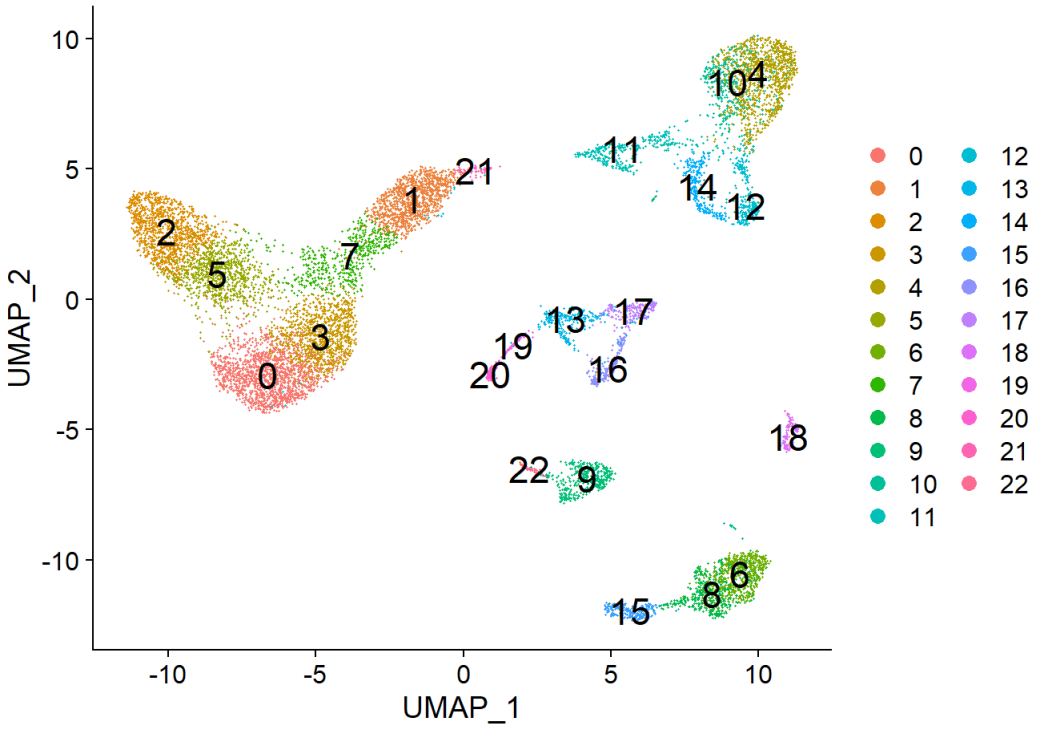
```
## 17:29:36 Building Annoy index with metric = cosine, n_trees = 50
```

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

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

```
## **************************************************|
## 17:29:38 Writing NN index file to temp file C:\Users\adiay\AppData\Local\Temp\Rtmp2vR0im\file3b9855622a58
## 17:29:38 Searching Annoy index using 1 thread, search_k = 3000
## 17:29:42 Annoy recall = 100%
## 17:29:42 Commencing smooth kNN distance calibration using 1 thread
## 17:29:43 Initializing from normalized Laplacian + noise
## 17:29:44 Commencing optimization for 200 epochs, with 483926 positive edges
## 17:29:57 Optimization finished
```

```
DimPlot(GSE_seurat, reduction = "umap",label="TRUE",label.size=6)
```




```
#Find All Markers
GSE.markers <- FindAllMarkers(GSE_seurat, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)
```

```
## Calculating cluster 0
```

```
## Calculating cluster 1
```

```
## Calculating cluster 2
```

```
## Calculating cluster 3
```

```
## Calculating cluster 4
```

```
## Calculating cluster 5
```

```
## Calculating cluster 6
```

```
## Calculating cluster 7
```

```
## Calculating cluster 8
```

```
## Calculating cluster 9
```

```
## Calculating cluster 10
```

```
## Calculating cluster 11
```

```
## Calculating cluster 12
```

```
## Calculating cluster 13
```

```
## Calculating cluster 14
```

```
## Calculating cluster 15
```

```
## Calculating cluster 16
```

```
## Calculating cluster 17
```

```
## Calculating cluster 18
```

```
## Calculating cluster 19
```

```
## Calculating cluster 20
```

```
## Calculating cluster 21
```

```
## Calculating cluster 22
```

```
GSE.markers %>% group_by(cluster) %>% top_n(10, avg_logFC)
```

```
## # A tibble: 230 x 7
## # Groups:   cluster [23]
##        p_val avg_logFC pct.1 pct.2 p_val_adj cluster gene
##        <dbl>     <dbl> <dbl> <dbl>     <dbl> <fct>   <chr>
##  1 0.            1.00  0.974 0.56  0.        0       CTNNA3
##  2 0.            0.981 0.997 0.755 0.        0       IL1RAPL1
##  3 0.            0.967 0.923 0.466 0.        0       ST18
##  4 0.            0.965 0.907 0.475 0.        0       SLC24A2
##  5 1.23e-284     1.03  0.846 0.455 1.33e-280 0       NKAIN2
##  6 2.33e-256     1.10  0.59  0.193 2.53e-252 0       PDE1A
##  7 5.35e-238     0.959 0.748 0.348 5.80e-234 0       MIR219A2
##  8 1.23e-231     0.963 0.594 0.206 1.33e-227 0       AK5
##  9 7.81e-213     1.06  0.502 0.163 8.47e-209 0       HS3ST5
## 10 8.81e-155     0.986 0.431 0.151 9.55e-151 0       LINC01170
## # ... with 220 more rows
```

```
View(GSE.markers)
# library(knitr)
# kable(GSE.markers)
```

```
cluster0.markers <- FindMarkers(GSE_seurat, ident.1 = 0, min.pct = 0.25)
View(head(cluster0.markers, n = 10))
```

```
#Find all markers of cluster 1
cluster1.markers <- FindMarkers(GSE_seurat, ident.1 = 1, min.pct = 0.25)
View(head(cluster1.markers, n = 10))
```

```
#Find all markers of cluster 2
cluster2.markers <- FindMarkers(GSE_seurat, ident.1 = 2, min.pct = 0.25)
View(head(cluster2.markers, n = 10))
```

```
#Find all markers of cluster 3
cluster3.markers <- FindMarkers(GSE_seurat, ident.1 = 3, min.pct = 0.25)
View(head(cluster3.markers, n = 10))
```

```
#Find all markers of cluster 4
cluster4.markers <- FindMarkers(GSE_seurat, ident.1 = 4, min.pct = 0.25)
View(head(cluster4.markers, n = 10))
```

```
#Find all markers of cluster 5
cluster5.markers <- FindMarkers(GSE_seurat, ident.1 = 5, min.pct = 0.25)
View(head(cluster5.markers, n = 10))
```

```
#Find all markers of cluster 6
cluster6.markers <- FindMarkers(GSE_seurat, ident.1 = 6, min.pct = 0.25)
View(head(cluster6.markers, n = 10))
```

```
#Find all markers of cluster 7
cluster7.markers <- FindMarkers(GSE_seurat, ident.1 = 7, min.pct = 0.25)
View(head(cluster7.markers, n = 10))
```

```
#Find all markers of cluster8
cluster8.markers <- FindMarkers(GSE_seurat, ident.1 = 8, min.pct = 0.25)
View(head(cluster8.markers, n = 10))
```

```
 #Find all markers of cluster 9
cluster9.markers <- FindMarkers(GSE_seurat, ident.1 = 9, min.pct = 0.25)
View(head(cluster9.markers, n = 10))
```

```
 #Find all markers of cluster 10
cluster10.markers <- FindMarkers(GSE_seurat, ident.1 = 10, min.pct = 0.25)
View(head(cluster10.markers, n = 10))
```

```
 #Find all markers of cluster 11
cluster11.markers <- FindMarkers(GSE_seurat, ident.1 = 11, min.pct = 0.25)
View(head(cluster11.markers, n = 10))
```

```
 #Find all markers of cluster 12
cluster12.markers <- FindMarkers(GSE_seurat, ident.1 = 12, min.pct = 0.25)
View(head(cluster12.markers, n = 10))
```

```
#Find all markers of cluster 13
cluster13.markers <- FindMarkers(GSE_seurat, ident.1 = 13, min.pct = 0.25)
View(head(cluster13.markers, n = 10))
```

```
#Find all markers of cluster 14
cluster14.markers <- FindMarkers(GSE_seurat, ident.1 = 14, min.pct = 0.25)
View(head(cluster14.markers, n = 10))
```

```
#Find all markers of cluster 15
cluster15.markers <- FindMarkers(GSE_seurat, ident.1 = 15, min.pct = 0.25)
View(head(cluster15.markers, n = 10))
```

```
#Find all markers of cluster 16
cluster16.markers <- FindMarkers(GSE_seurat, ident.1 = 16, min.pct = 0.25)
View(head(cluster16.markers, n = 10))
```

```
#Find all markers of cluster 17
cluster17.markers <- FindMarkers(GSE_seurat, ident.1 = 17, min.pct = 0.25)
View(head(cluster17.markers, n = 10))
```

```
#Find all markers of cluster 18
cluster18.markers <- FindMarkers(GSE_seurat, ident.1 = 18, min.pct = 0.25)
View(head(cluster18.markers, n = 10))
```

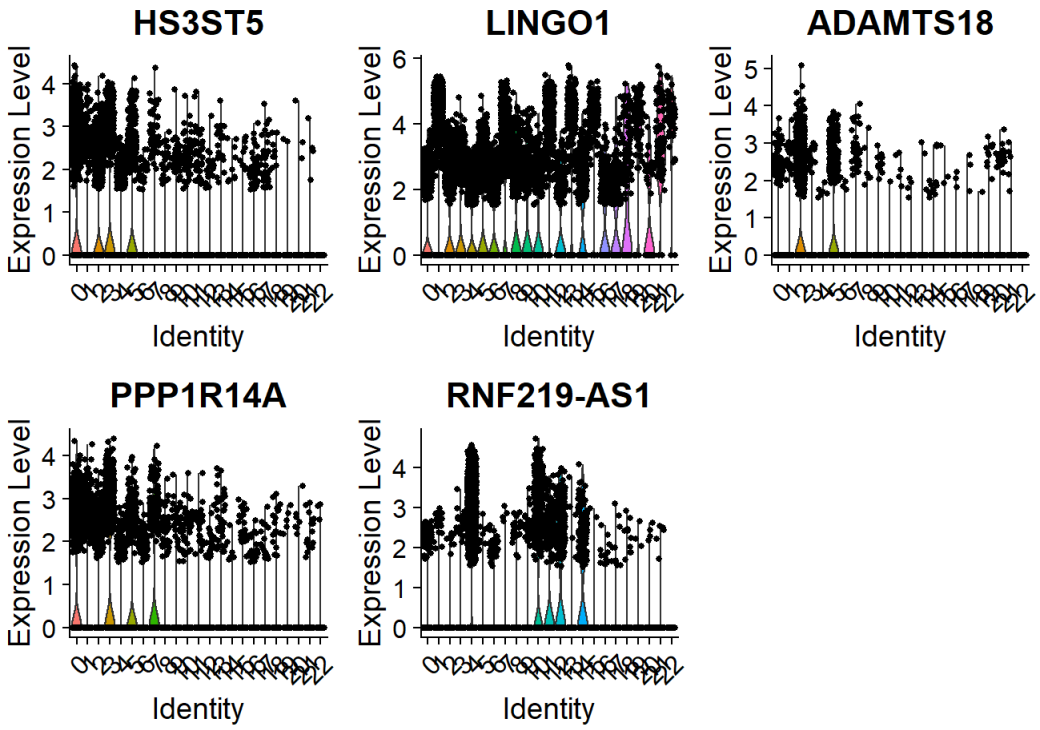
```
#Find all markers of cluster 19
cluster19.markers <- FindMarkers(GSE_seurat, ident.1 = 19, min.pct = 0.25)
View(head(cluster19.markers, n = 10))
```

```
#Find all markers of cluster 20
cluster20.markers <- FindMarkers(GSE_seurat, ident.1 = 20, min.pct = 0.25)
View(head(cluster20.markers, n = 10))
```

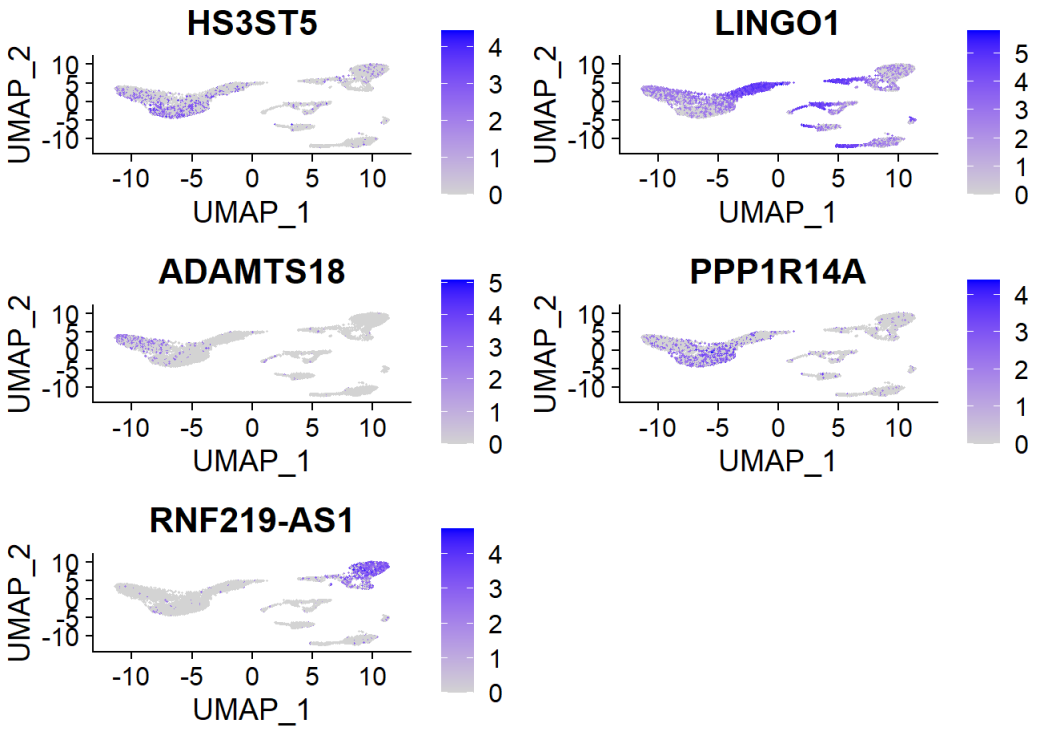
```
#Find all markers of cluster 21
cluster21.markers <- FindMarkers(GSE_seurat, ident.1 = 21, min.pct = 0.25)
View(head(cluster21.markers, n = 10))
```

```
#Find all markers of cluster 22
cluster22.markers <- FindMarkers(GSE_seurat, ident.1 = 22, min.pct = 0.25)
View(head(cluster22.markers, n = 10))
```

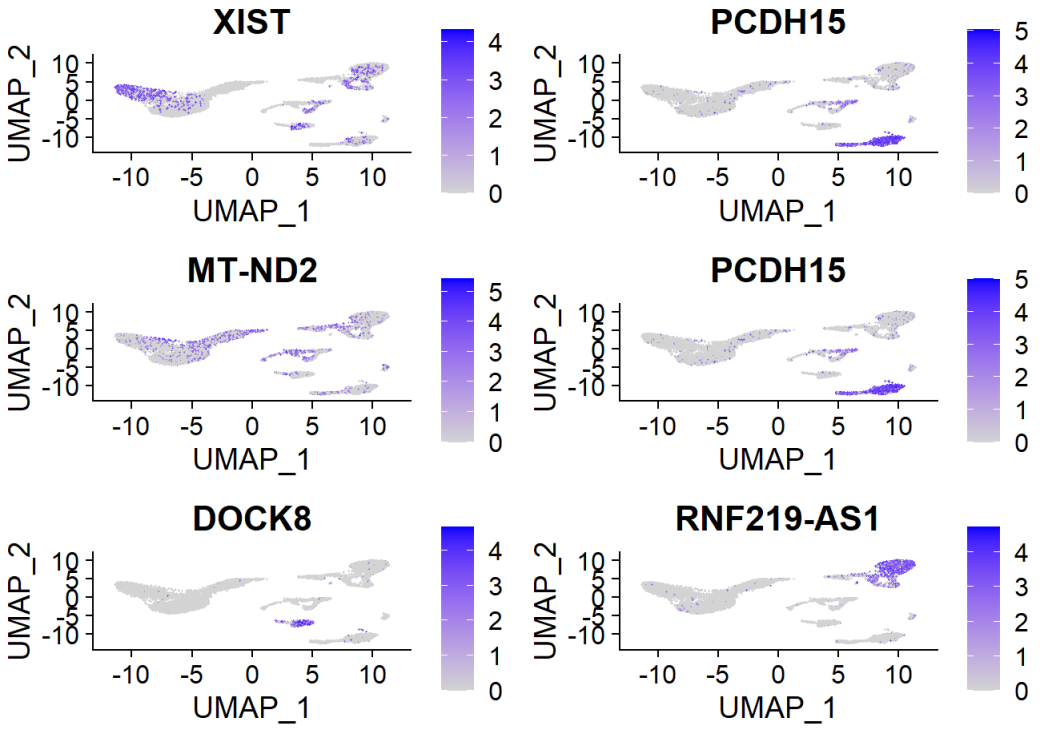
```
VlnPlot(GSE_seurat, features = c("HS3ST5", "LINGO1","ADAMTS18","PPP1R14A", "RNF219-AS1"))
```



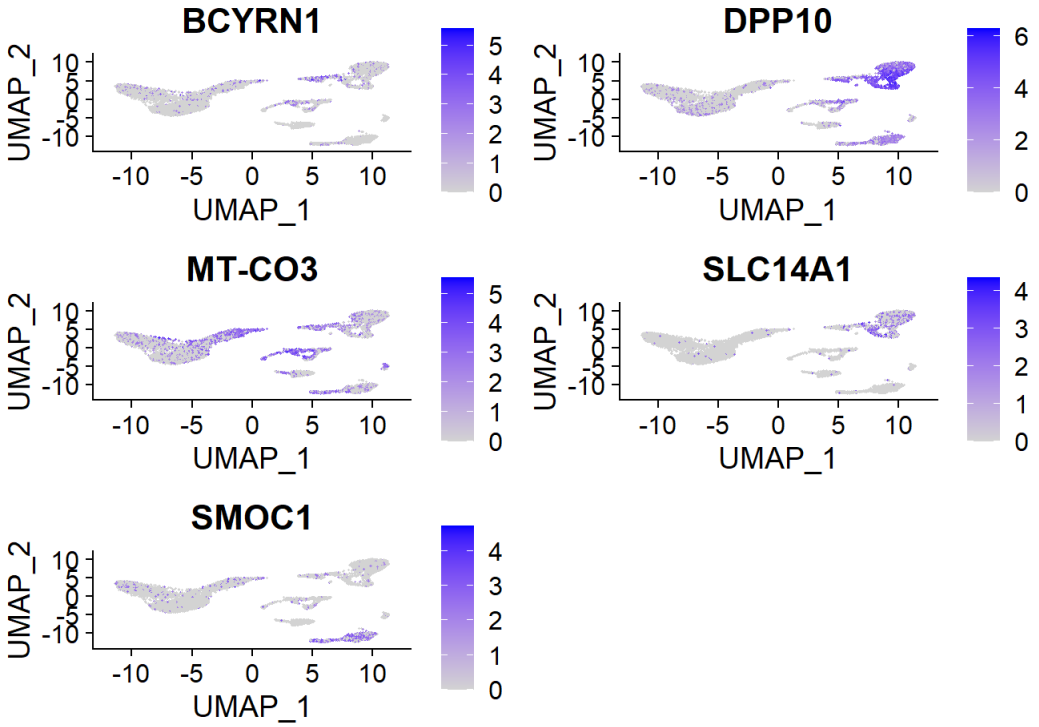
```
FeaturePlot(GSE_seurat, features = c("HS3ST5", "LINGO1","ADAMTS18","PPP1R14A", "RNF219-AS1"))
```



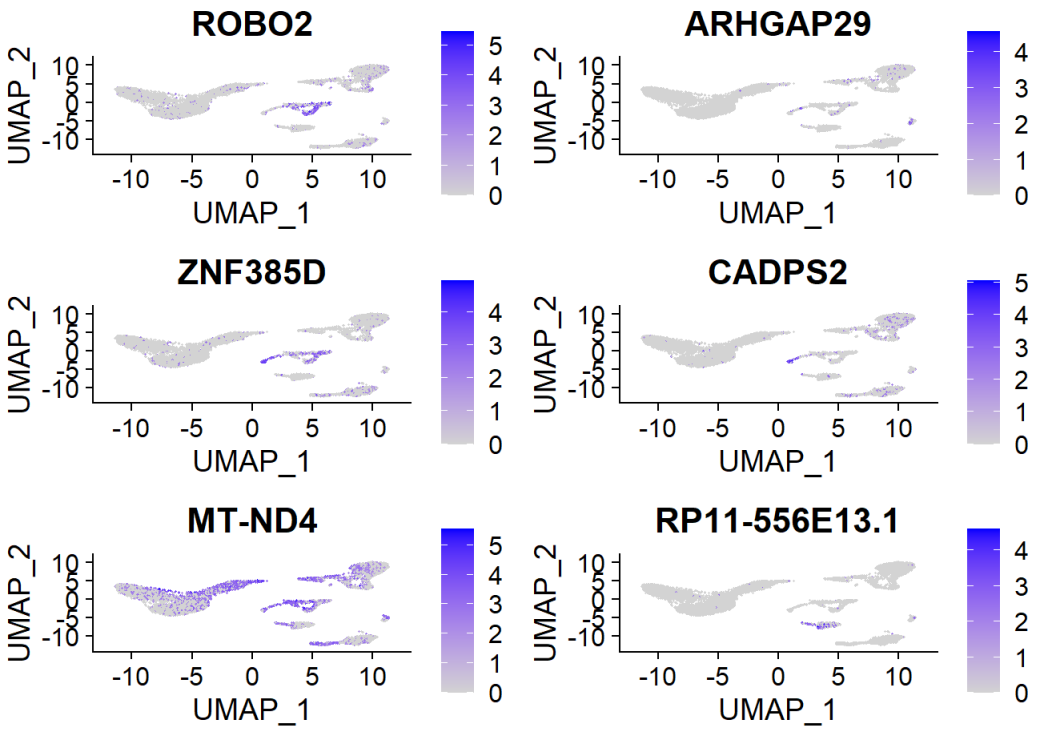

```
FeaturePlot(GSE_seurat, features = c(   "XIST", "PCDH15", "MT-ND2","PCDH15",    "DOCK8", "RNF219-AS1"))
```



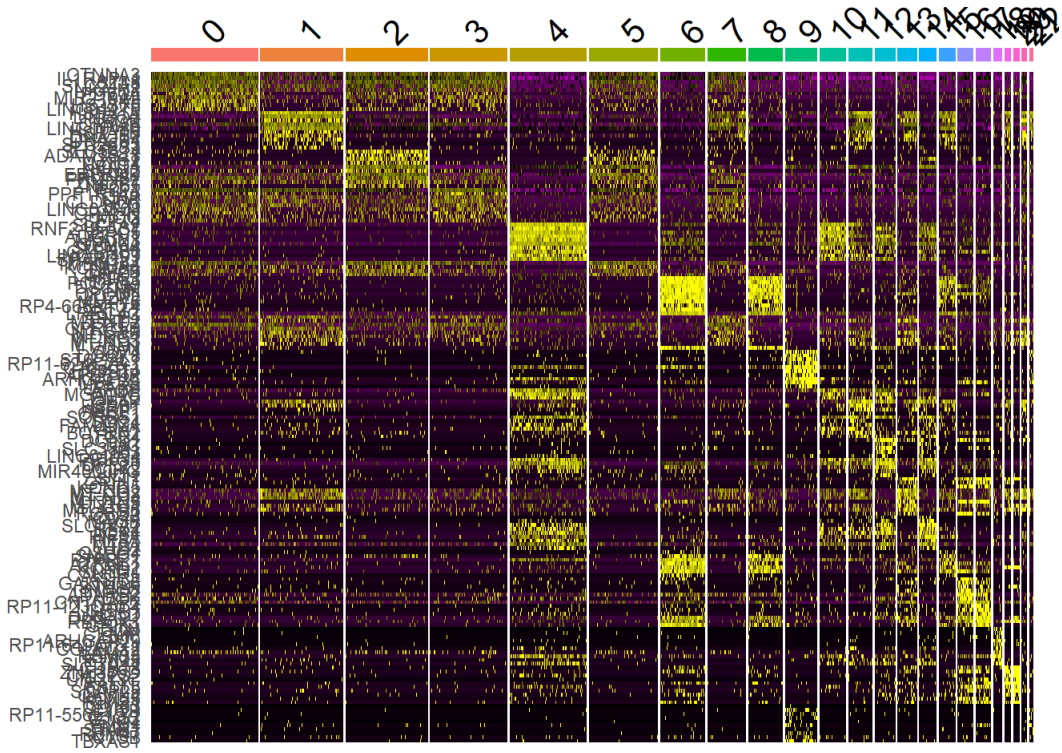

```
FeaturePlot(GSE_seurat, features = c("BCYRN1", "DPP10","MT-CO3", "SLC14A1","SMOC1"))
```




```
FeaturePlot(GSE_seurat, features = c("ROBO2","ARHGAP29","ZNF385D","CADPS2","MT-ND4","RP11-556E13.1"))
```




```
top10 <- GSE.markers %>% group_by(cluster) %>% top_n(n = 10, wt = avg_logFC)
DoHeatmap(GSE_seurat, features = top10$gene) + NoLegend()
```



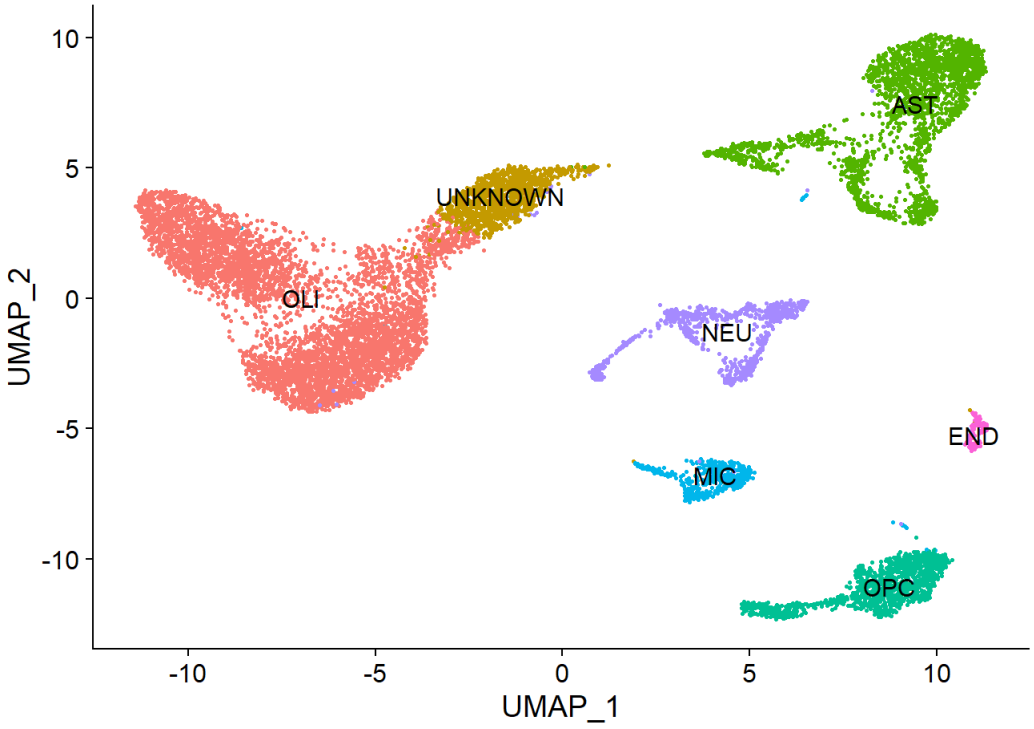

```
View(cell_labels)
```

```
new.cluster.ids <- c("0"="OLI" ,
                     "1"="UNKNOWN",
                     "2"="OLI",
                     "3"="OLI",
                     "4"="AST",
                     "5"="OLI",

                     "6"="OPC",
                     "7"="OLI",
                     "8"="OPC",
                     "9"="MIC",
                     "10"="AST",
                     "11"="AST",
                     "12"="AST",
                     "13"="NEU",
                     "14"="AST",
                     "15"="OPC",
                     "16"="NEU",
                     "17"="NEU",
                     "18"="END",
                     "19"="NEU",
                     "20"="NEU",
                     "21"="UNKNOWN",
                     "22"="MIC"
                      )

names(new.cluster.ids) <- levels(GSE_seurat)
GSE_seurat_Cell <- RenameIdents(GSE_seurat,new.cluster.ids)
```

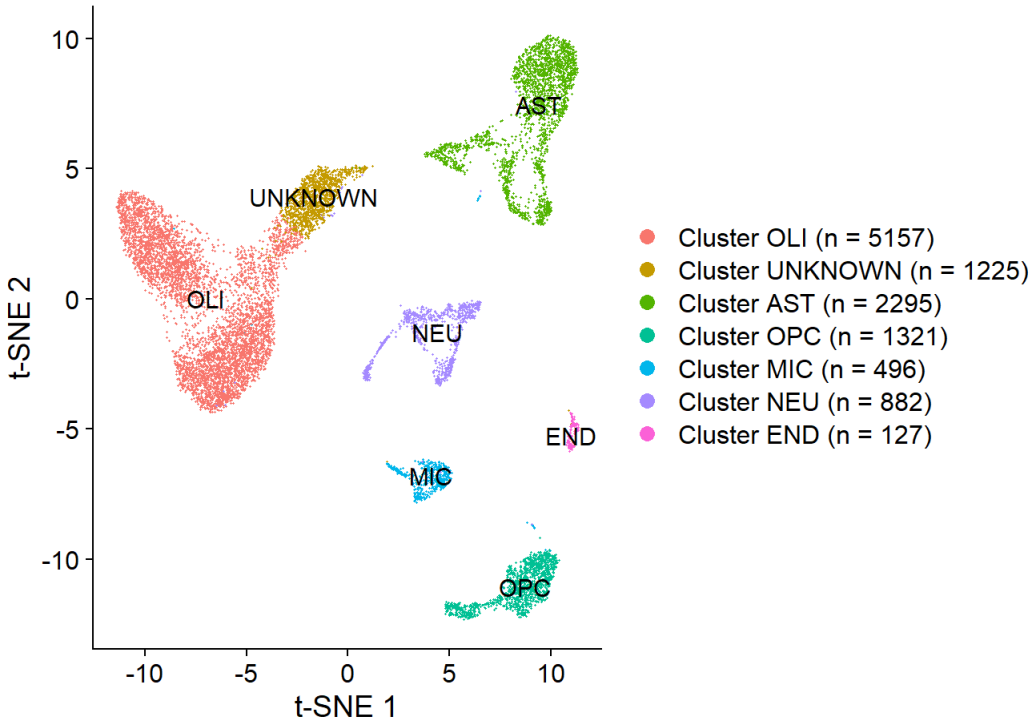
```
#How many Cells are in each cluster

head(Idents(GSE_seurat_Cell), 15)
```

```
## AAACCTGGTAGAAAGG_AD5_AD6 AAACCTGGTAGCGATG_AD5_AD6 AAACCTGTCAGTCAGT_AD5_AD6
##                      OLI                  UNKNOWN                      OLI
## AAACCTGTCCAAACAC_AD5_AD6 AAACCTGTCCAGTATG_AD5_AD6 AAAGCAAGTCGAATCT_AD5_AD6
##                      OLI                      OLI                      OLI
## AAAGTAGGTAATCACC_AD5_AD6 AAATGCCCAATAGCGG_AD5_AD6 AAATGCCGTCATCCCT_AD5_AD6
##                      OLI                      OLI                      OLI
## AACACGTAGCTGTCTA_AD5_AD6 AACACGTTCCTCCTAG_AD5_AD6 AACCATGCACATGGGA_AD5_AD6
##                      AST                      OLI                  UNKNOWN
## AACCATGGTTGTGGAG_AD5_AD6 AACCATGTCCGTACAA_AD5_AD6 AACCATGTCTGATACG_AD5_AD6
##                      OLI                  UNKNOWN                      AST
## Levels: OLI UNKNOWN AST OPC MIC NEU END
```

```
DimPlot(GSE_seurat_Cell, reduction = "umap", label = TRUE, pt.size = 0.5) + NoLegend()
```




```
# Calculate number of cells per cluster from object@ident
cell.num <- table(Idents( GSE_seurat_Cell ))
# Add cell number per cluster to cluster labels
ClusterLabels = paste("Cluster", names(cell.num), paste0("(n = ", cell.num, ")"))
View(ClusterLabels)
# Order legend labels in plot in the same order as 'ClusterLabels'
ClusterBreaks = names(cell.num)

# Plot tSNE with new legend labels for clusters
DimPlot(GSE_seurat_Cell , reduction ="umap",label = TRUE ) +
  scale_colour_discrete(breaks = ClusterBreaks,
                        labels = ClusterLabels) +
  labs(x = "t-SNE 1",
       y = "t-SNE 2")
```




```
save(GSE_seurat_Cell, file="C:/Lab08-Data-Wrangling-scRNAseq/GSE138852/data/GSESeuratCellType.rds")
```