KedarProject_LiuLab_UTSW

2025-11-27

```
# Load necessary libraries
library(Seurat)
```

```
## Warning: package 'Seurat' was built under R version 4.5.1
```

```
## Loading required package: SeuratObject
```

```
## Warning: package 'SeuratObject' was built under R version 4.5.1
```

```
## Loading required package: sp
```

```
## 'SeuratObject' was built with package 'Matrix' 1.7.3 but the current
## version is 1.7.4; it is recomended that you reinstall 'SeuratObject' as
## the ABI for 'Matrix' may have changed
```

```
##
## Attaching package: 'SeuratObject'
```

```
## The following objects are masked from 'package:base':
##
##     intersect, t
```

```
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(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.5.1
```

```r
library(pheatmap)
library(dplyr)
```

```r
# 1 Read the data for GSE72056
df_72056 <- read.table("/Users/kedarjoshi/Downloads/GSE72056.txt",
                       header = TRUE, sep = "\t", stringsAsFactors = FALSE)

# 2 Extract metadata (first 3 rows)
metadata_72056 <- df_72056[1:3, ]
rownames(metadata_72056) <- metadata_72056$Cell
metadata_72056 <- metadata_72056[, -1]  # remove "Cell" column


malignant_72056 <- as.factor(unlist(metadata_72056["malignant(1=no,2=yes,0=unresolved)", ]))
celltype_72056  <- as.factor(unlist(metadata_72056["non-malignant cell type (1=T,2=B,3=Macro.4=Endo.,5=C
tumor_72056     <- unlist(metadata_72056["tumor", ])


# 3 Extract gene expression matrix (all rows after metadata)
expr_df <- df_72056[-(1:3), ]
# Handle duplicate genes by summing duplicates
expr_numeric <- as.data.frame(lapply(expr_df[, -1], as.numeric))
expr_numeric$Gene <- expr_df$Cell
```

```r
### The below steps have been completed on cluster due to space issue in local system.
### The completed expr_matrix is saved as a csv from cluster and then read into local for quick analysi

# Sum duplicates
#expr_agg <- expr_numeric %>%
#  group_by(Gene) %>%
#  summarise(across(where(is.numeric), sum), .groups = "drop")

# Convert to data.frame
#expr_agg <- as.data.frame(expr_agg)
# Set rownames safely
#rownames(expr_agg) <- expr_agg$Gene
# Remove the redundant Gene column
#expr_agg <- expr_agg[, -1]
# Convert to numeric matrix for Seurat
#expr_matrix <- as.matrix(expr_agg)

# 4 Filter candidate genes
#genes_to_keep <- c("CASKIN2","EMC9","PDIK1L","DBNDD2", "FAM171A2")
#expr_final <- expr_agg[rownames(expr_agg) %in% genes_to_keep, ]

# Save expression matrix
#write.csv(expr_matrix, "/home/joshi.ked/expr_matrix.csv", row.names = TRUE)

### Cluster work completed, now the below chunk is run locally.
```

```r
# Load expression matrix GSE72056
expr_matrix_72056 <- read.csv("/Users/kedarjoshi/Downloads/expr_matrix.csv", row.names = 1)
expr_matrix_72056 <- as.matrix(expr_matrix_72056)  # make sure it's numeric matrix
```

```r
# Load expression matrix GSE115978

expr_matrix_115978 <- read.csv("/Users/kedarjoshi/Downloads/GSE115978_counts.csv", row.names = 1, check
expr_matrix_115978 <- as.matrix(expr_matrix_115978)
mode(expr_matrix_115978) <- "numeric"

# ------------------------------
# Load metadata table
# ------------------------------
meta115978 <- read.csv("/Users/kedarjoshi/Downloads/GSE115978_cell.annotations.csv",
                       stringsAsFactors = FALSE)

# Match metadata rows to expression matrix columns
meta115978 <- meta115978[ match(colnames(expr_matrix_115978), meta115978$cells), ]

rownames(meta115978) <- meta115978$cells
meta115978 <- meta115978[, -1]   # remove "cells" col since it's rownames
```

```r
# ------------------------------
# Create Seurat object for GSE72056
# ------------------------------
meta72056 <- data.frame(
  Malignant = malignant_72056,
  CellType  = celltype_72056,
  Tumor     = tumor_72056,
  row.names = colnames(expr_matrix_72056)
)

seurat_72056 <- CreateSeuratObject(counts = expr_matrix_72056,
                                   meta.data = meta72056)
```

```
## Warning: Feature names cannot have underscores ('_'), replacing with dashes
## ('-')
```

```
## Warning: Data is of class matrix. Coercing to dgCMatrix.
```

```r
seurat_72056$dataset <- "GSE72056"
```

```r
# ------------------------------
# Create Seurat object for GSE115978
# ------------------------------
seurat_115978 <- CreateSeuratObject(counts = expr_matrix_115978, meta.data = meta115978)
```

```
## Warning: Feature names cannot have underscores ('_'), replacing with dashes
## ('-')
```

```
## Warning: Data is of class matrix. Coercing to dgCMatrix.
```

```r
seurat_115978$dataset <- "GSE115978"
```

```
##########################################
# PART 3 - NORMALIZATION + INTEGRATION PREP
##########################################

# Normalize & find variable genes (recommended for Smart-seq2)
seurat_72056 <- NormalizeData(seurat_72056)
```

```
## Normalizing layer: counts
```

```
seurat_72056 <- FindVariableFeatures(seurat_72056)
```

```
## Finding variable features for layer counts
```

```
seurat_115978 <- NormalizeData(seurat_115978)
```

```
## Normalizing layer: counts
```

```
seurat_115978 <- FindVariableFeatures(seurat_115978)
```

```
## Finding variable features for layer counts
```

```
# Store in a list
obj_list <- list(seurat_72056, seurat_115978)
```

```
##########################################
# PART 4 - INTEGRATION (CCA-based)
##########################################

anchors <- FindIntegrationAnchors(
  object.list = obj_list,
  dims = 1:30
)
```

```
## Warning in CheckDuplicateCellNames(object.list = object.list): Some cell names
## are duplicated across objects provided. Renaming to enforce unique cell names.
```

```
## Computing 2000 integration features
```

```
## Scaling features for provided objects
```

```
## Finding all pairwise anchors
```

```
## Running CCA
```

```
## Merging objects
```

```
## Finding neighborhoods
```

```
## Warning: package 'future' was built under R version 4.5.1
```

```
## Finding anchors
```

```
##  Found 13374 anchors
```

```
## Filtering anchors
```

```
##  Retained 10248 anchors
```

```
combined <- IntegrateData(anchorset = anchors)
```

```
## Merging dataset 1 into 2
```

```
## Extracting anchors for merged samples
```

```
## Finding integration vectors
```

```
## Finding integration vector weights
```

```
## Integrating data
```

```
## Warning: Layer counts isn't present in the assay object; returning NULL
```

```
# Set integrated as default assay
DefaultAssay(combined) <- "integrated"
```

```
##########################################
# PART 5 - DIMENSION REDUCTION + CLUSTERING
##########################################

combined <- ScaleData(combined)
```

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

```
combined <- RunPCA(combined)
```

```
## PC_ 1
## Positive:  SPARC, SERPINE2, LMNA, CD63, GSN, LGALS3BP, GPNMB, S100B, CTTN, IFITM3
##     TIMP1, PRAME, VIM, CSPG4, APOD, PLP1, TYR, LGALS3, CAV1, LGALS1
##     A2M, HSPB1, CST3, SDCBP, MFGE8, SGK1, CTSB, SRPX, PMEL, ANXA5
## Negative:  PTPRC, TMSB4X, ARHGDIB, CORO1A, PTPRCAP, LAPTM5, CXCR4, RAC2, LCP1, IL2RG
##     CD53, CD3D, CD2, IL32, LSP1, LCK, CD52, SRGN, CD74, CD37
##     CD69, ITGB2, SLA, UCP2, SASH3, CST7, NKG7, ZAP70, TBC1D10C, CD48
## PC_ 2
## Positive:  PRAME, PLP1, PMEL, CD3D, TYR, MLANA, SERPINE2, CD2, GAS5, S100B
##     GPR143, CSPG4, CAPN3, CTTN, LCK, IL32, ZAP70, APOD, MFGE8, STMN1
##     CAV1, PLEKHB1, MIA, SLC24A5, SPARC, PIR, SLC45A2, LOXL4, CD8A, ITM2A
## Negative:  CSF1R, CD14, TYROBP, FCER1G, IFI30, CD163, LILRB4, HCK, TMEM176B, FCGR1A
```

```
##     LYZ, IGSF6, C1QA, C1QC, AIF1, VSIG4, CYBB, C1QB, LILRB2, CD68
##     FGL2, PILRA, SLCO2B1, NCF2, MS4A6A, SERPINA1, FCGR3A, MS4A4A, SLC7A7, TMEM176A
## PC_ 3
## Positive:  COL1A1, THY1, COL3A1, BGN, DCN, FBN1, EFEMP1, LUM, CCDC80, CDH11
##     ABI3BP, COL6A3, COL5A1, MMP2, ISLR, C1S, C1R, COL1A2, ELN, FBLN2
##     MXRA8, COL5A2, PODN, COL14A1, MXRA5, CXCL14, MFAP4, LOX, C3, DPT
## Negative:  TYR, PMEL, CAPN3, MLANA, S100B, PRAME, GPNMB, QPCT, PLP1, APOE
##     MIA, SGK1, GPR143, CSPG4, APOC2, APOD, LGALS3, SLC24A5, PIR, BIRC7
##     GDF15, SDCBP, SERPINE2, LGALS3BP, SAT1, MFGE8, FXYD3, SLC45A2, CTSD, MIF
## PC_ 4
## Positive:  MS4A1, CD79A, BANK1, CD19, CD22, CD79B, NAPSB, FCRL1, IRF8, TCL1A
##     HLA-DOB, IGLL5, FCER2, FCRLA, BLNK, CR2, NCF1C, NCF1, ADAM28, VPREB3
##     HVCN1, FCRL5, BTK, CNR2, STAP1, CLEC17A, HLA-DRA, CD83, CXCR5, NCF1B
## Negative:  NKG7, IL32, PRF1, CST7, CD8A, SRGN, CD2, GZMA, CCL4, KLRK1
##     CD3D, ID2, CTSW, TIGIT, GZMK, CD8B, PDCD1, GIMAP4, ITM2A, CTSD
##     GZMH, IFITM1, GAPDH, LCK, SIRPG, CCL3, CCL4L1, CCL4L2, CXCR6, ZAP70
## PC_ 5
## Positive:  COL1A1, DCN, LUM, COL3A1, COL6A3, ISLR, THBS2, COL5A1, PODN, PDGFRL
##     DPT, SFRP2, FBLN1, SFRP4, COL1A2, C3, MXRA8, COL14A1, CXCL14, MXRA5
##     CD248, MFAP5, SMOC2, PCOLCE, PDGFRB, SERPINF1, SVEP1, ASPN, SULF1, ITGA11
## Negative:  CDH5, VWF, CALCRL, ECSCR, EGFL7, CLDN5, HYAL2, PLVAP, ELTD1, CLEC14A
##     RAMP3, RAMP2, SDPR, ESAM, COL4A1, SLCO2A1, KDR, CCL14, COL4A2, CD34
##     MMRN1, PODXL, EMCN, LYVE1, ADAMTS9, LDB2, DARC, STAB1, AQP1, CYYR1
```

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

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

## 19:06:56 UMAP embedding parameters a = 0.9922 b = 1.112

## 19:06:56 Read 11831 rows and found 15 numeric columns

## 19:06:56 Using Annoy for neighbor search, n_neighbors = 30

## 19:06:56 Building Annoy index with metric = cosine, n_trees = 50

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

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

## **************************************************|
## 19:06:58 Writing NN index file to temp file /var/folders/b8/kjrfrjc57p98wsyrnd0041jr0000gn/T//Rtmp6b
## 19:06:58 Searching Annoy index using 1 thread, search_k = 3000
## 19:07:02 Annoy recall = 100%
## 19:07:02 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 19:07:03 Initializing from normalized Laplacian + noise (using RSpectra)
## 19:07:04 Commencing optimization for 200 epochs, with 510142 positive edges
## 19:07:04 Using rng type: pcg
## 19:07:11 Optimization finished
```

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

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

```
combined <- FindClusters(combined)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 11831
## Number of edges: 413575
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9057
## Number of communities: 25
## Elapsed time: 2 seconds
```

```
genes_to_keep <- c("CASKIN2","EMC9","PDIK1L","DBNDD2", "FAM171A2", "C1orf174", "LOC124903857", "TMEM161

# Compute average expression per cluster in RNA assay
avg_expr <- AverageExpression(combined, assays = "RNA", features = genes_to_keep, group.by = "dataset")
```

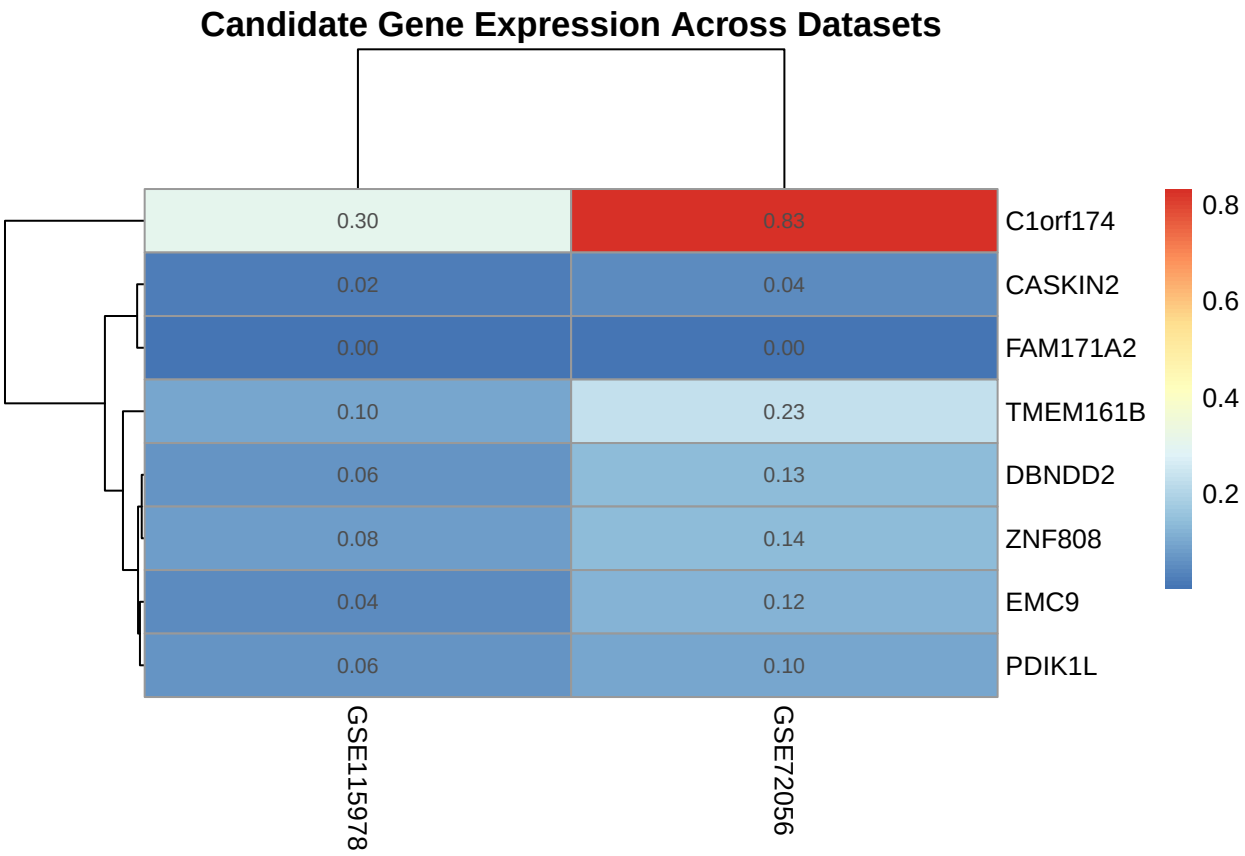
```
## As of Seurat v5, we recommend using AggregateExpression to perform pseudo-bulk analysis.
## This message is displayed once per session.
```

```
## Warning: The following 1 features were not found in the RNA assay: LOC124903857
```

```
avg_expr$RNA
```

```
## 8 x 2 sparse Matrix of class "dgCMatrix"
##             GSE115978    GSE72056
## C1orf174 0.3023288610 0.832888779
## CASKIN2  0.0225345673 0.043676003
## DBNDD2   0.0614950490 0.134404441
## EMC9     0.0434735843 0.121367351
## FAM171A2 0.0006034978 0.001484771
## PDIK1L   0.0645754812 0.100183661
## TMEM161B 0.0989882041 0.233176013
## ZNF808   0.0794759522 0.139241468
```

```
# plot the expression in form of a pheatmap
pheatmap(avg_expr$RNA,
         cluster_rows = TRUE,     # Cluster genes
         cluster_cols = TRUE,     # Cluster datasets (if more than 2)
         display_numbers = TRUE,  # Optional: show values
         main = "Candidate Gene Expression Across Datasets")
```




```
# Split by dataset
seurat_list <- SplitObject(combined, split.by = "dataset")

# For each dataset, find cluster markers in RNA assay
markers_list <- lapply(seurat_list, function(obj) {
  DefaultAssay(obj) <- "RNA"
  Idents(obj) <- "seurat_clusters"
  FindAllMarkers(obj, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)
})
```

```
## Calculating cluster 0

## For a (much!) faster implementation of the Wilcoxon Rank Sum Test,
## (default method for FindMarkers) please install the presto package
## --------------------------------------------
## install.packages('devtools')
## devtools::install_github('immunogenomics/presto')
## --------------------------------------------
## After installation of presto, Seurat will automatically use the more
## efficient implementation (no further action necessary).
## This message will be shown once per session

## 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 22

## Calculating cluster 23

## Calculating cluster 24

## 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

## Calculating cluster 23

## Calculating cluster 24
```

```
# Access markers for each dataset
markers_72056  <- markers_list[["GSE72056"]]
markers_115978 <- markers_list[["GSE115978"]]
```

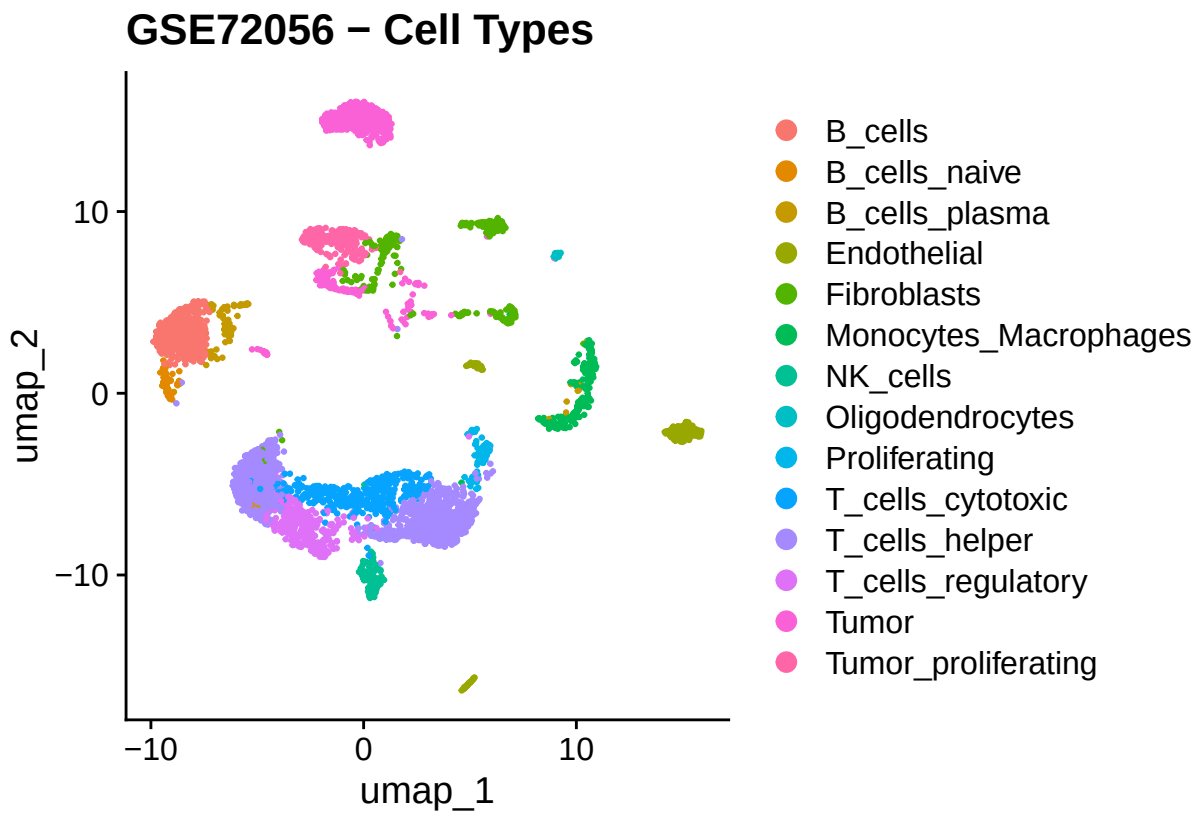
```r
filtered_markers_72056 <- markers_72056 %>%
  filter(p_val_adj < 0.05,        # statistically significant
         avg_log2FC > 0.5,        # enriched in cluster
         pct.1 > 0.25)            # expressed in  25% of cluster cells

top_markers_72056 <- filtered_markers_72056 %>%
  group_by(cluster) %>%
  slice_max(order_by = avg_log2FC, n = 5) %>%  # top 5 per cluster
  ungroup()

filtered_markers_115978 <- markers_115978 %>%
  filter(p_val_adj < 0.05,        # statistically significant
         avg_log2FC > 0.5,        # enriched in cluster
         pct.1 > 0.25)            # expressed in  25% of cluster cells

top_markers_115978 <- filtered_markers_115978 %>%
  group_by(cluster) %>%
  slice_max(order_by = avg_log2FC, n = 5) %>%  # top 5 per cluster
  ungroup()
```

```r
# Based on manual review of the top markers filtered in the above chunk we annotate the clusters

cluster2celltype_72056 <- c(
  "0"="T_cells_helper", "1"="T_cells_helper", "2"="B_cells",
  "3"="T_cells_cytotoxic", "4"="Tumor", "5"="T_cells_regulatory",
  "6"="Tumor_proliferating", "7"="Monocytes_Macrophages", "8"="Fibroblasts",
  "9"="Endothelial", "10"="NK_cells", "11"="Proliferating",
  "12"="Tumor", "13"="Fibroblasts", "14"="Tumor", "15"="B_cells_plasma",
  "16"="B_cells_naive", "17"="Fibroblasts", "18"="Tumor",
  "19"="Endothelial", "20"="Endothelial", "22"="Oligodendrocytes",
  "23"="Tumor", "24"="Fibroblasts"
)

cluster2celltype_115978 <- c(
  "0"="T_cells_helper", "1"="T_cells_helper", "2"="B_cells",
  "3"="T_cells_cytotoxic", "4"="Tumor", "5"="T_cells_regulatory",
  "6"="Tumor_proliferating", "7"="Monocytes_Macrophages", "8"="Fibroblasts",
  "9"="Endothelial", "10"="NK_cells", "11"="Proliferating",
  "12"="Tumor", "13"="Fibroblasts", "14"="Tumor", "15"="B_cells_plasma",
  "16"="B_cells_naive", "17"="Fibroblasts", "18"="Tumor",
  "19"="Endothelial", "20"="Endothelial", "21"="Tumor",
  "22"="Fibroblasts", "23"="Tumor", "24"="Fibroblasts"
)
```

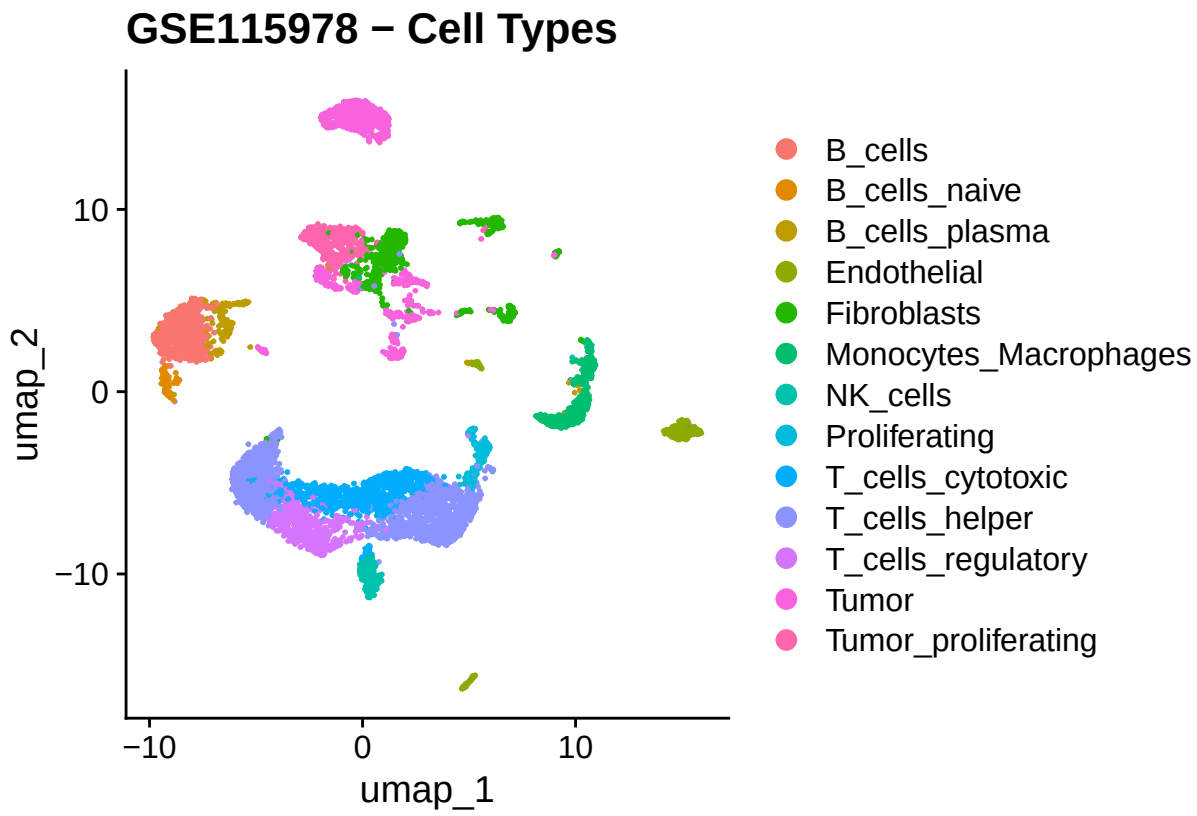
```r
# Function to map clusters to cell types
map_clusters_manual <- function(seurat_obj, mapping) {
  Idents(seurat_obj) <- "seurat_clusters"
  clusters <- as.character(Idents(seurat_obj))
  celltype <- mapping[clusters]
  celltype[is.na(celltype)] <- clusters[is.na(celltype)]
  names(celltype) <- names(clusters)
  seurat_obj[["celltype"]] <- factor(celltype)
  Idents(seurat_obj) <- "celltype"
```

```
  return(seurat_obj)
}

# Apply mapping
seurat_list[["GSE72056"]] <- map_clusters_manual(seurat_list[["GSE72056"]], cluster2celltype_72056)
seurat_list[["GSE115978"]] <- map_clusters_manual(seurat_list[["GSE115978"]], cluster2celltype_115978)
```

```
# Plot DimPlot once per dataset
DimPlot(seurat_list[["GSE72056"]], reduction = "umap", label = FALSE) +
  ggtitle("GSE72056 - Cell Types")
```

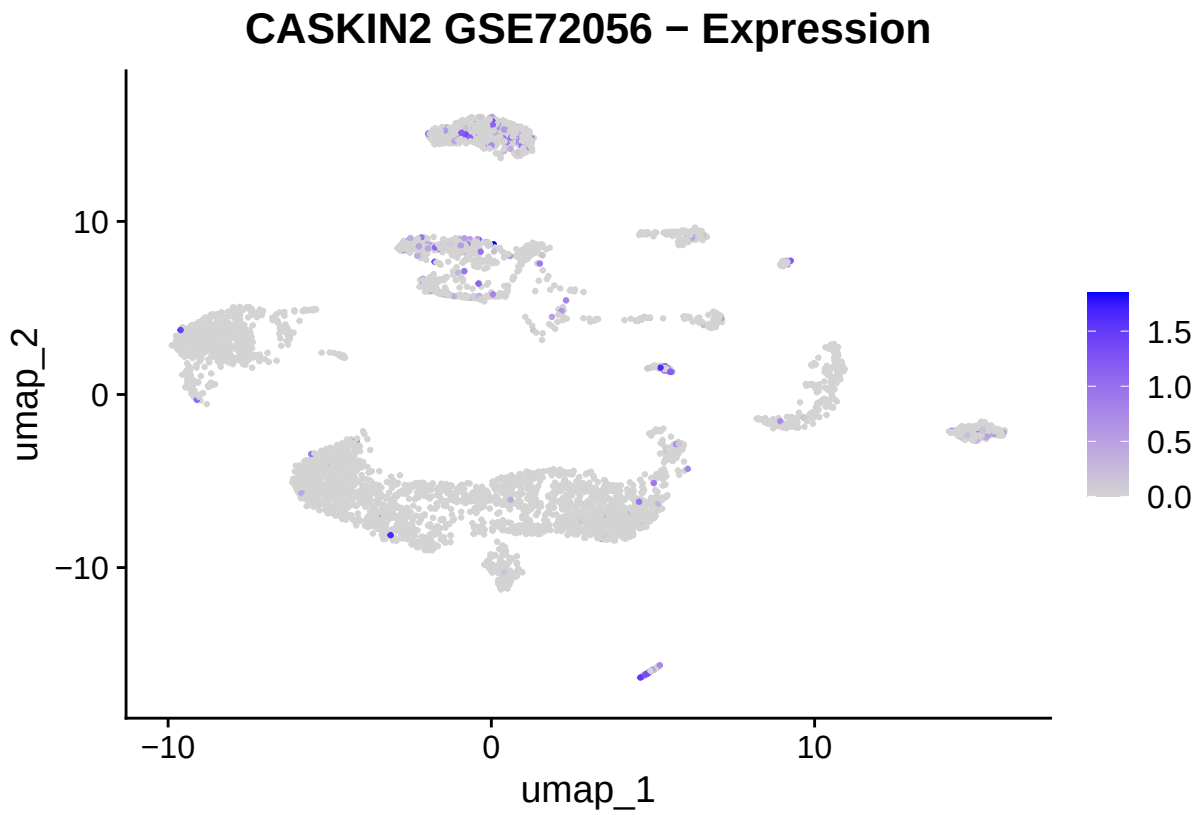


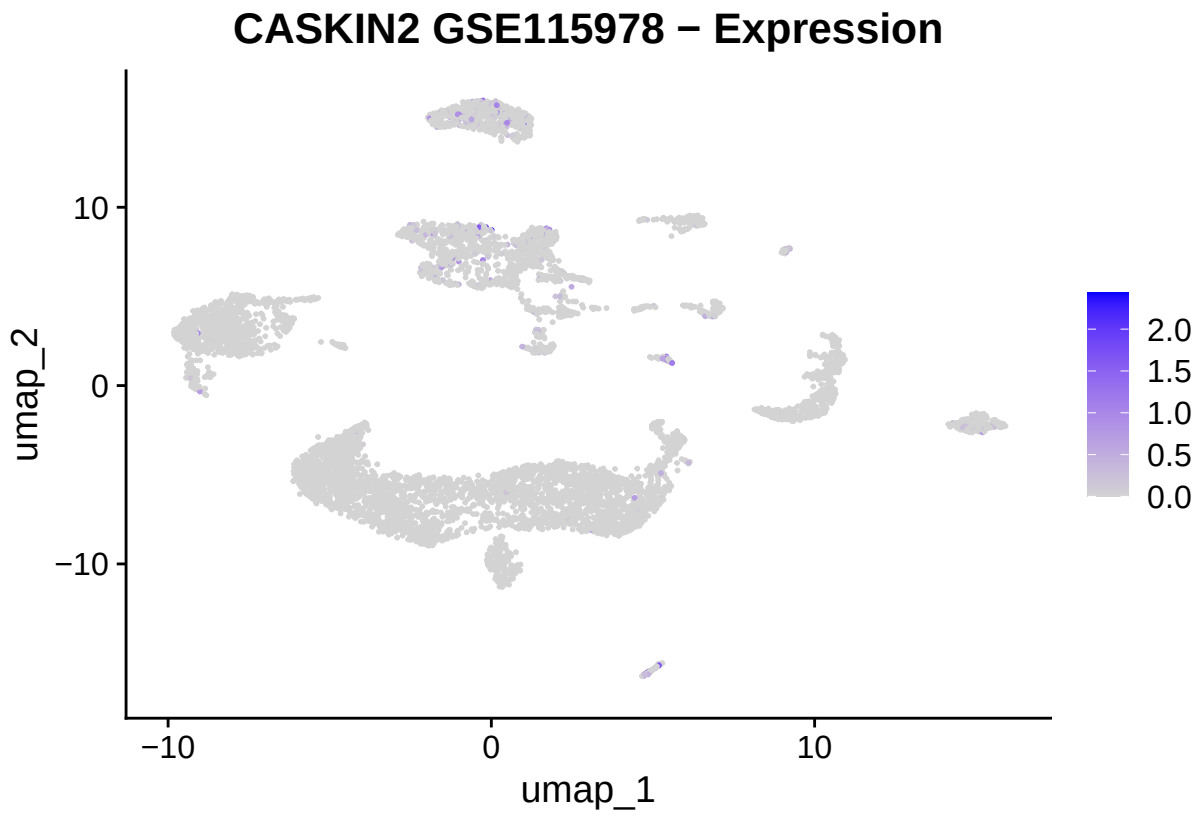

```
DimPlot(seurat_list[["GSE115978"]], reduction = "umap", label = FALSE) +
  ggtitle("GSE115978 - Cell Types")
```

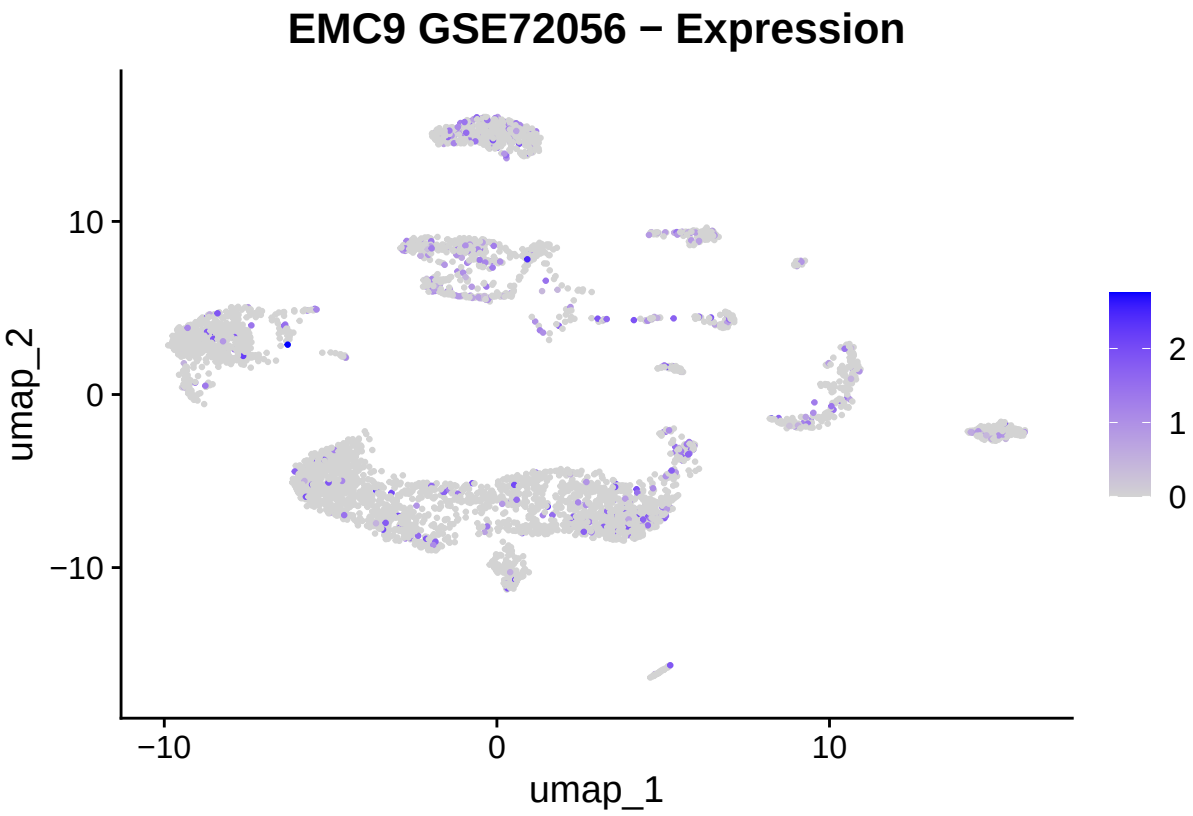


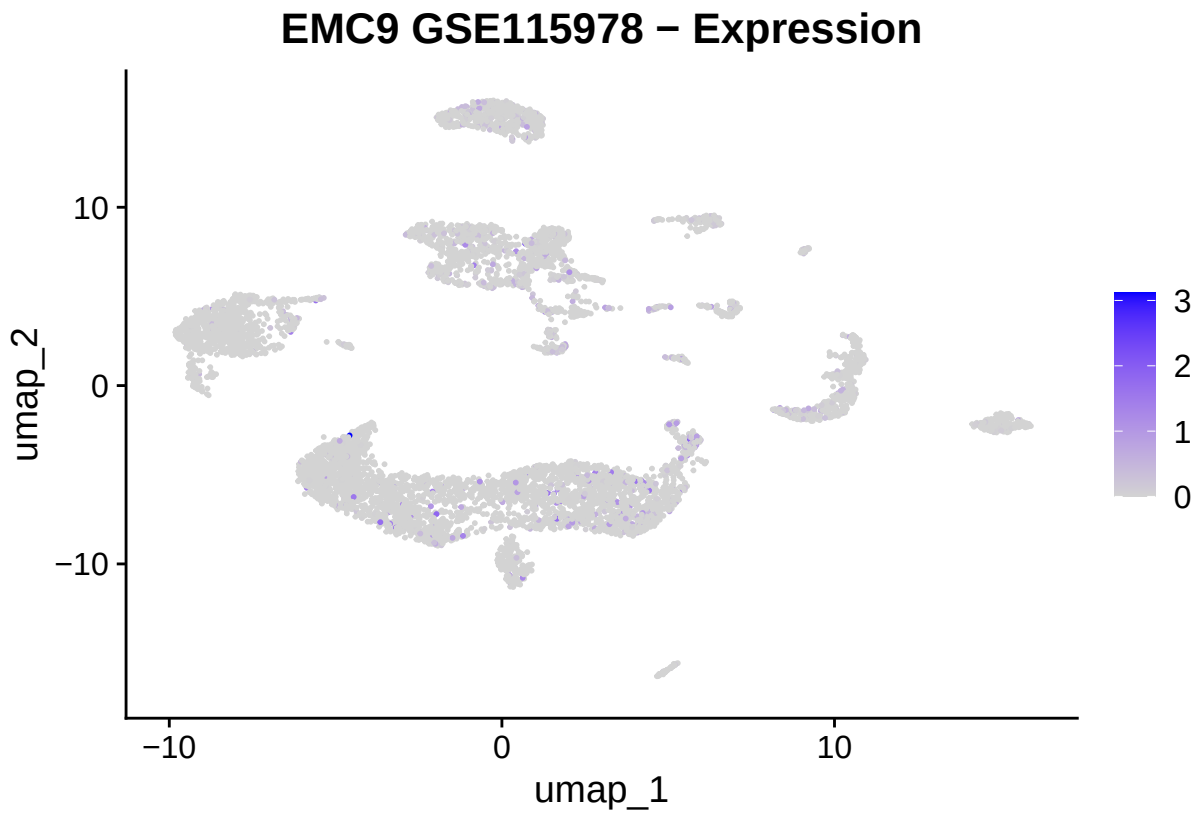

```
# Make sure RNA assay is active
DefaultAssay(seurat_list[["GSE72056"]]) <- "RNA"
DefaultAssay(seurat_list[["GSE115978"]]) <- "RNA"

genes_to_keep <- c("CASKIN2","EMC9","PDIK1L","DBNDD2", "FAM171A2", "C1orf174", "LOC124903857", "TMEM161

for (gene in genes_to_keep) {

  # GSE72056
  if (gene %in% rownames(seurat_list[["GSE72056"]])) {
    print(
      FeaturePlot(
        seurat_list[["GSE72056"]],
        features = gene,
        reduction = "umap",
        cols = c("lightgrey", "blue")
      ) + ggtitle(paste(gene, "GSE72056 - Expression"))
    )
  } else {
    message("Gene not found in GSE72056: ", gene)
  }

  # GSE115978
  if (gene %in% rownames(seurat_list[["GSE115978"]])) {
    print(
      FeaturePlot(
```

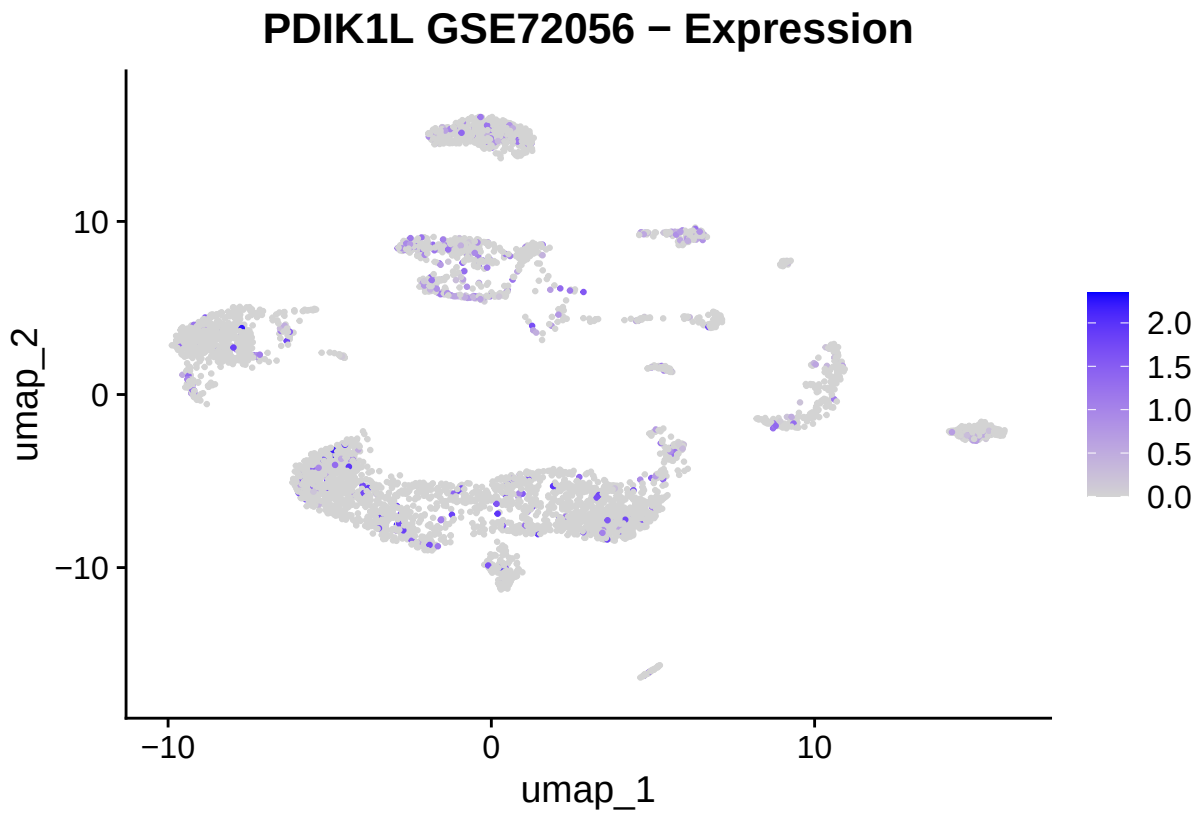
```
        seurat_list[["GSE115978"]],
        features = gene,
        reduction = "umap",
        cols = c("lightgrey", "blue")
      ) + ggtitle(paste(gene, "GSE115978 - Expression"))
    )
  } else {
    message("Gene not found in GSE115978: ", gene)
  }
}
```

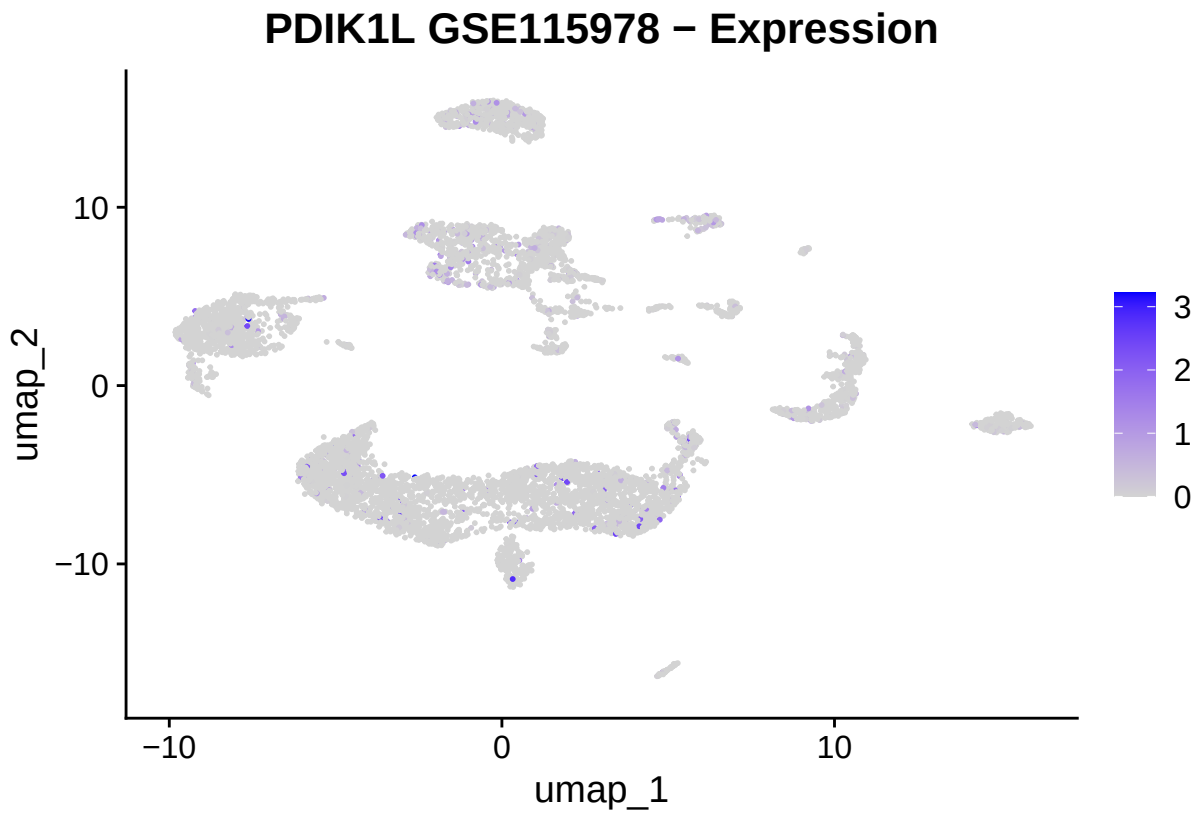


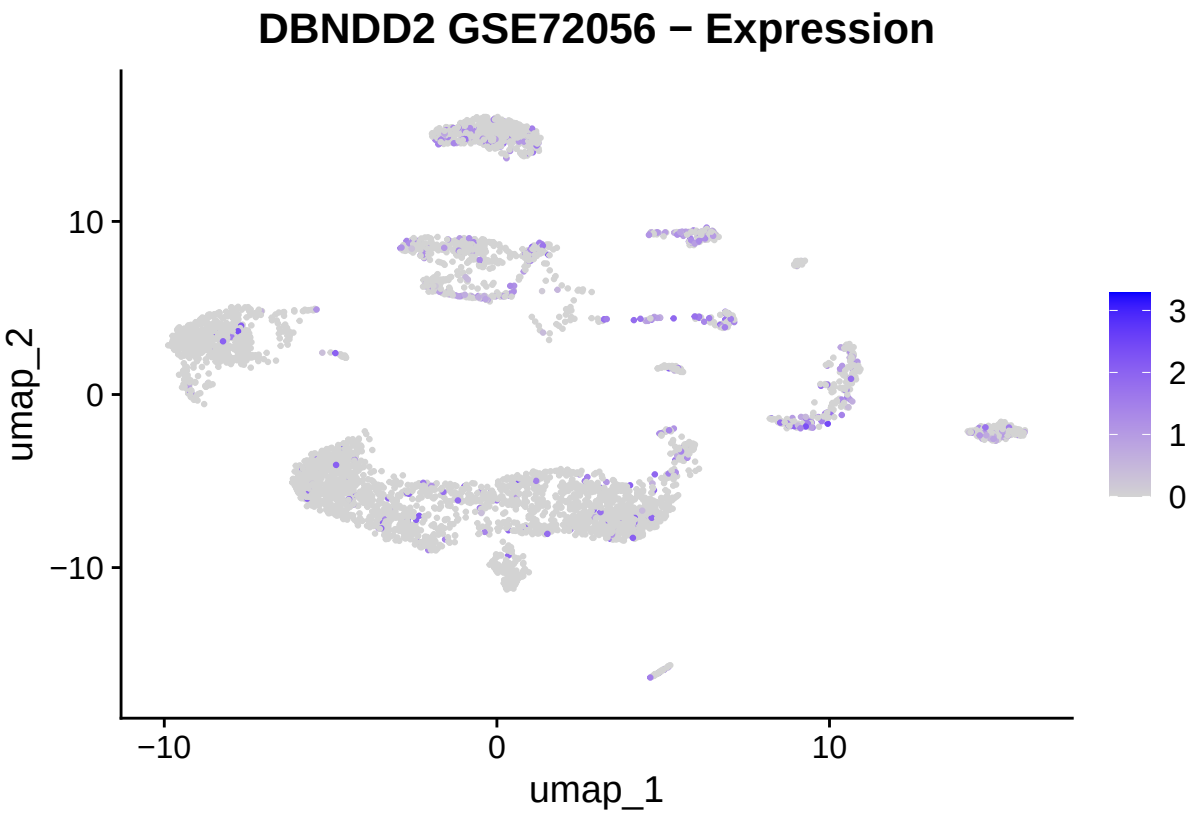

CASKIN2 GSE115978 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
2.0
1.5
1.0
0.5
0.0

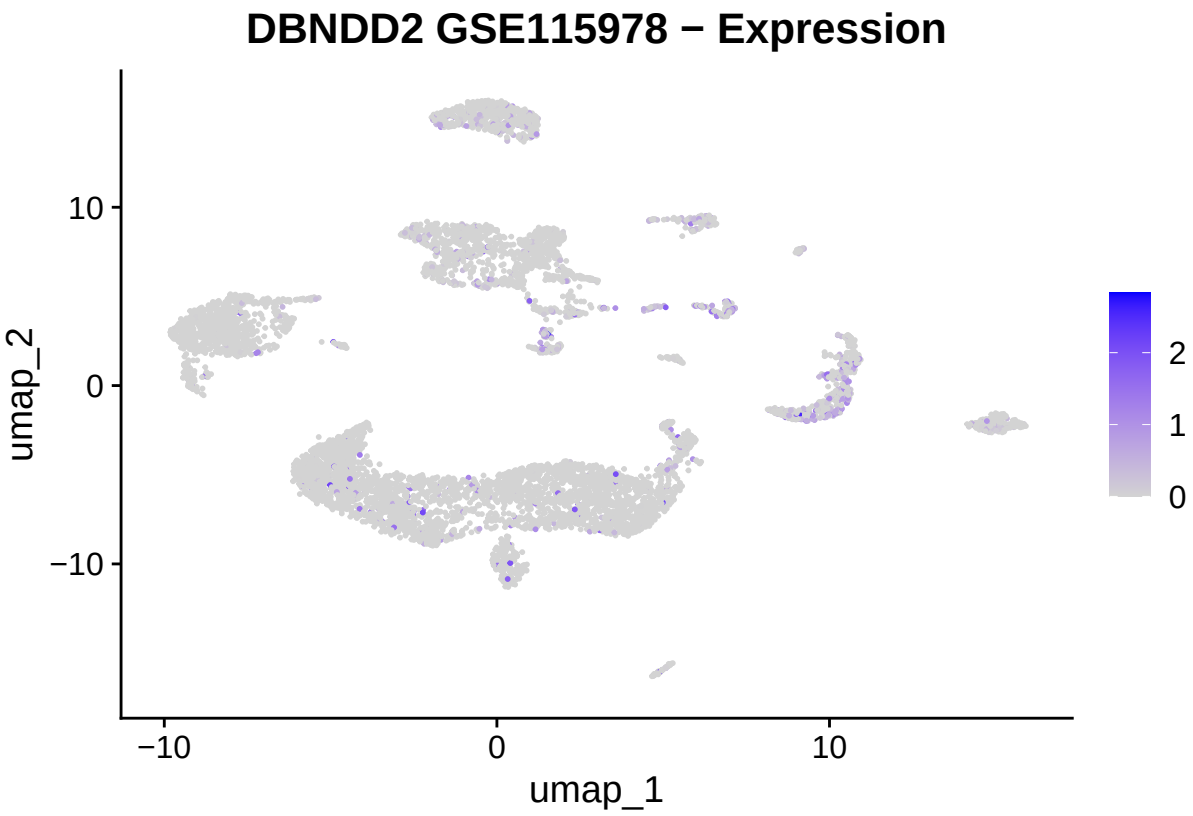

EMC9 GSE72056 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
2
1
0

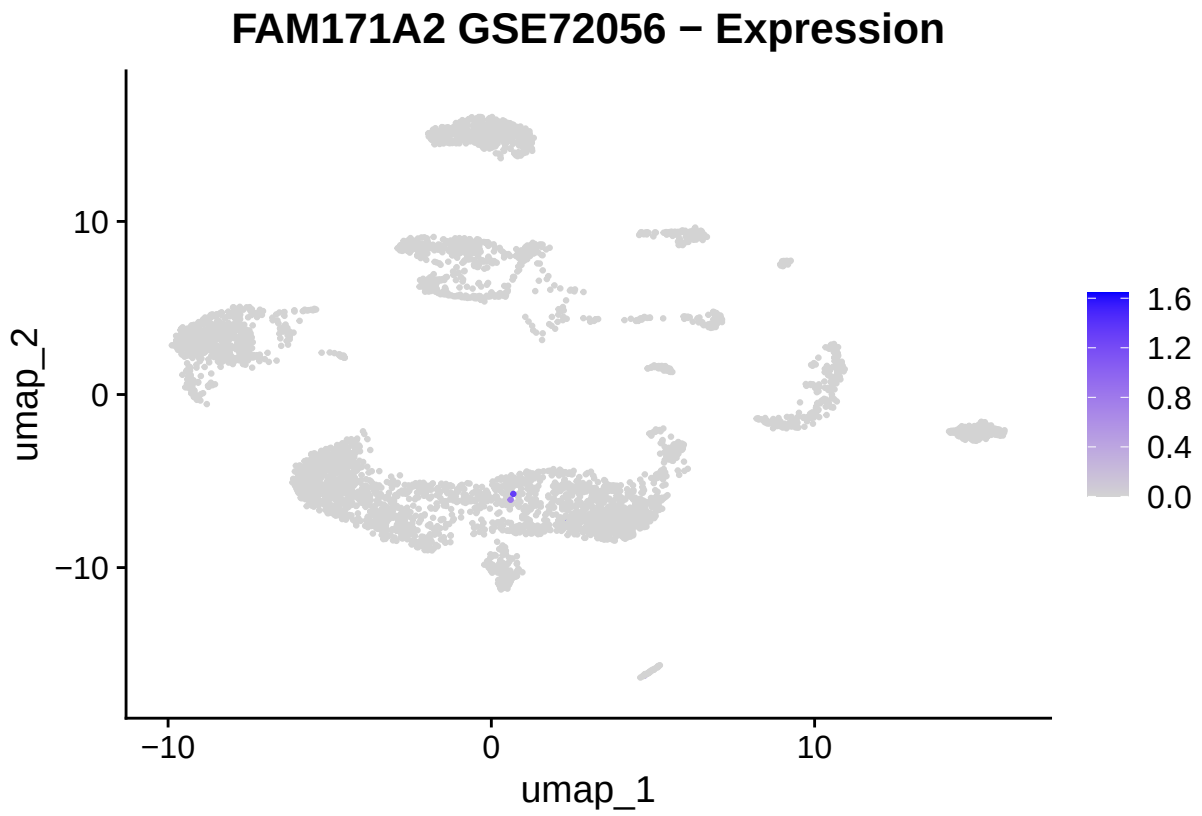

EMC9 GSE115978 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
3
2
1
0

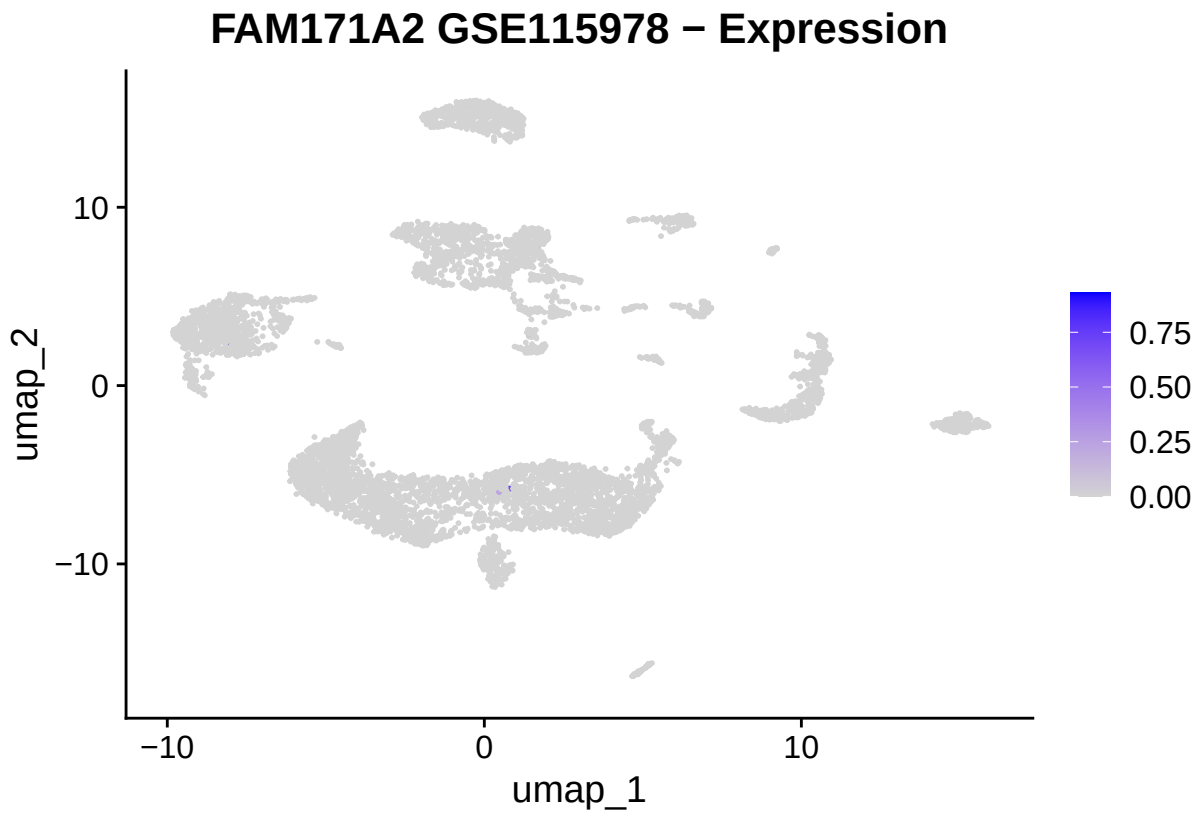

PDIK1L GSE72056 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
2.0
1.5
1.0
0.5
0.0

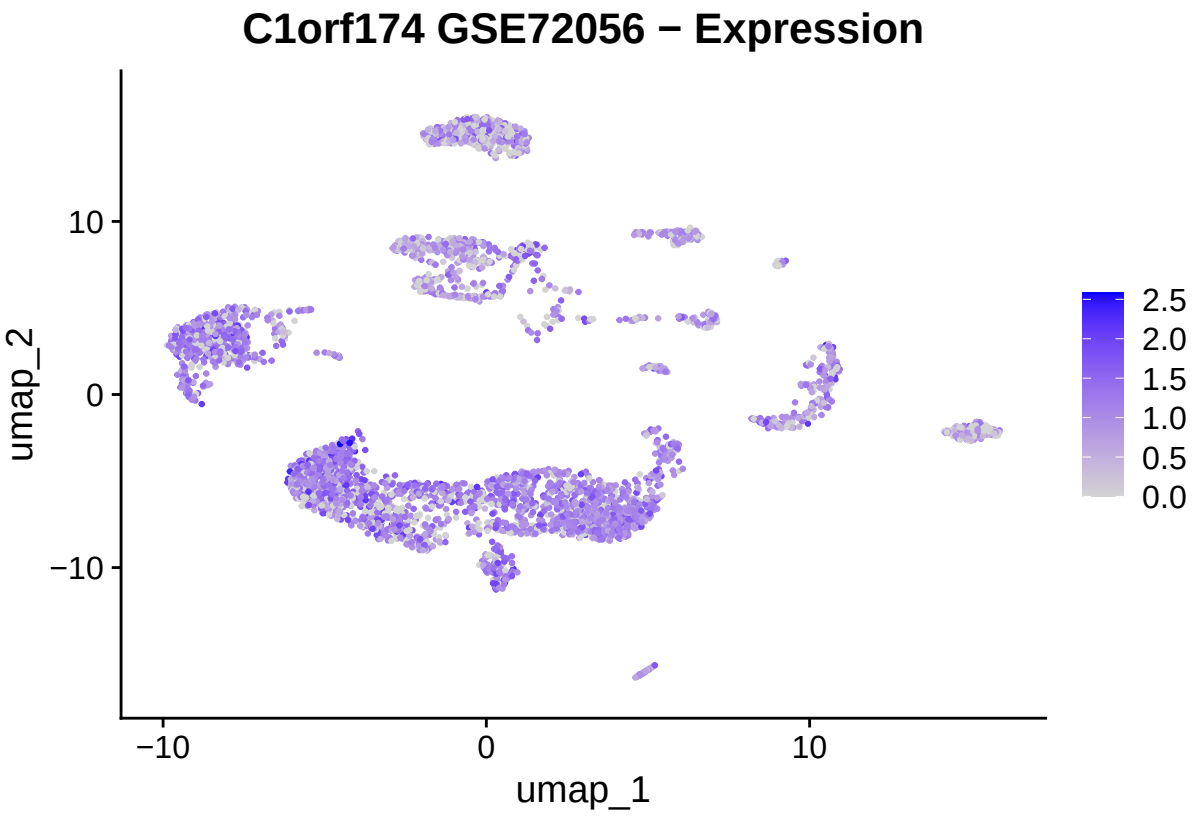

PDIK1L GSE115978 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
3
2
1
0

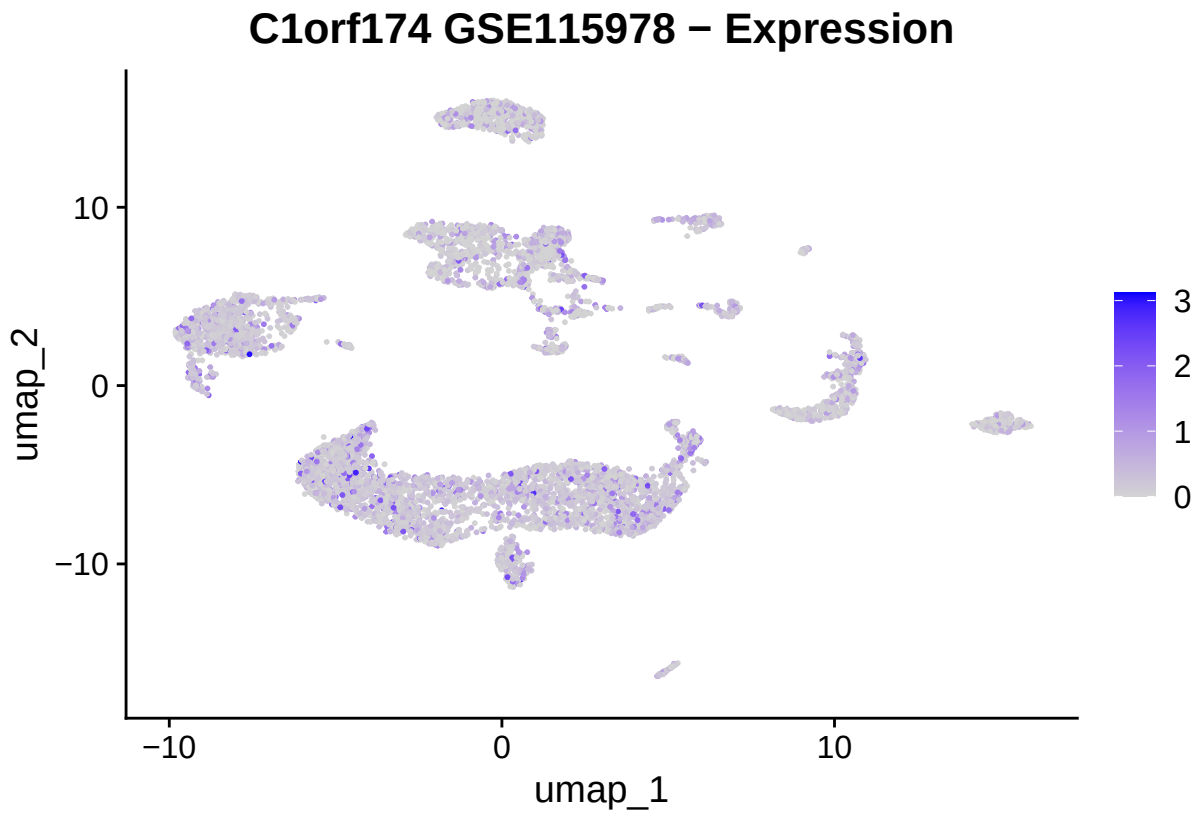

DBNDD2 GSE72056 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
3
2
1
0


DBNDD2 GSE115978 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
2
1
0

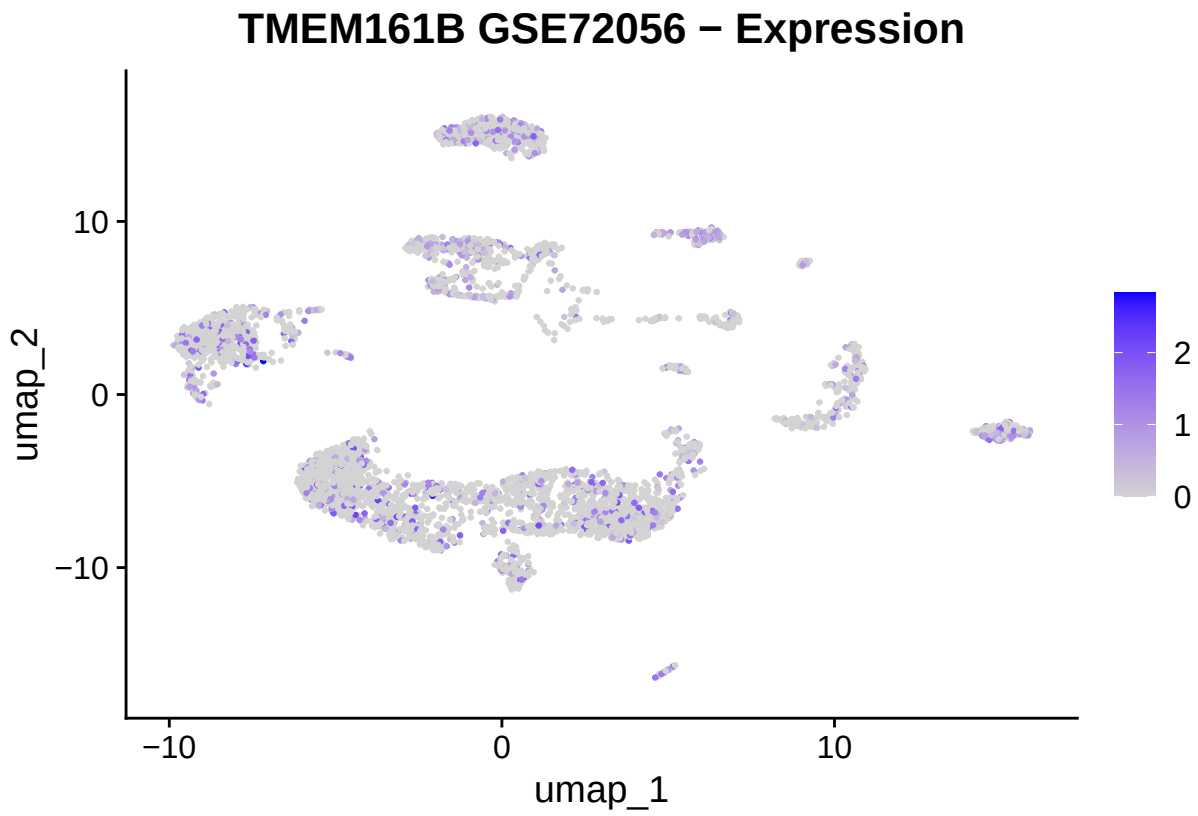

FAM171A2 GSE72056 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
1.6
1.2
0.8
0.4
0.0

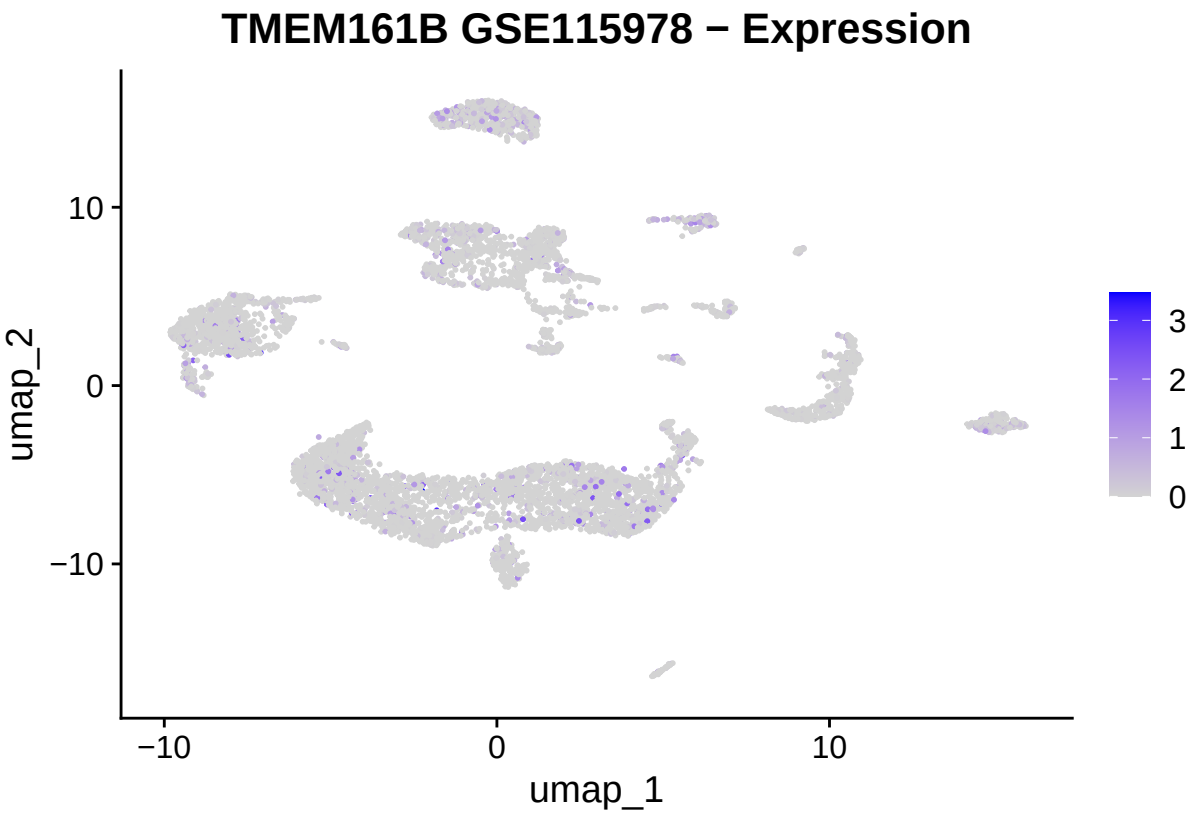

FAM171A2 GSE115978 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
0.75
0.50
0.25
0.00

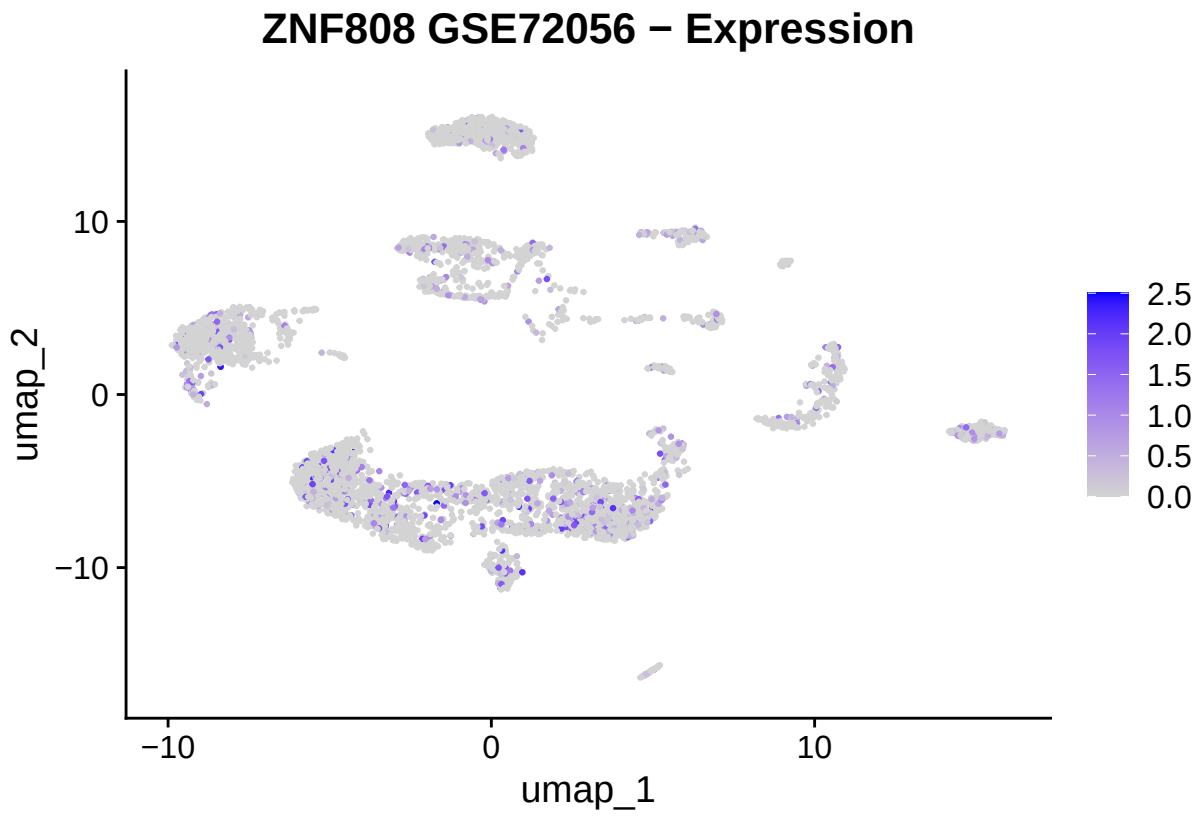

C1orf174 GSE72056 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
2.5
2.0
1.5
1.0
0.5
0.0

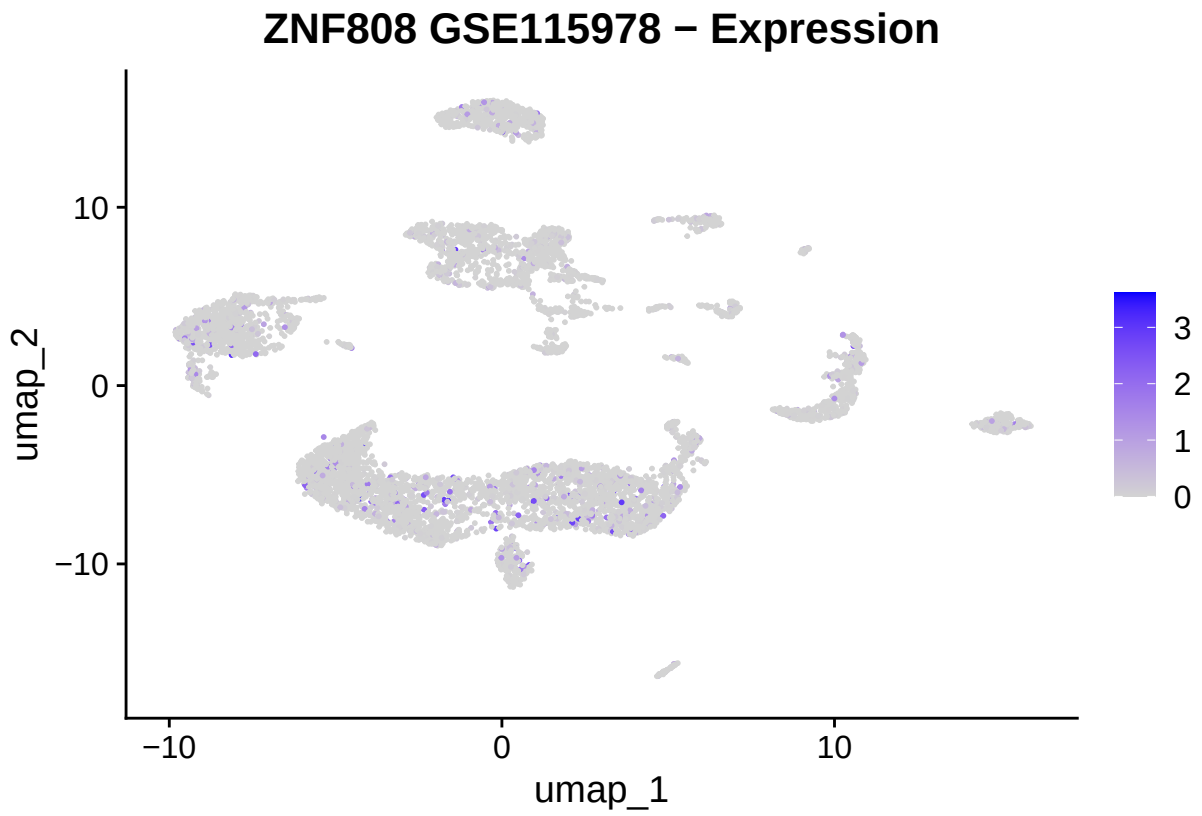



```
## Gene not found in GSE72056: LOC124903857

## Gene not found in GSE115978: LOC124903857
```


TMEM161B GSE72056 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
2
1
0


TMEM161B GSE115978 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
3
2
1
0


ZNF808 GSE72056 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
2.5
2.0
1.5
1.0
0.5
0.0


ZNF808 GSE115978 − Expression
umap_2
10
0
−10
−10
0
10
umap_1
3
2
1
0