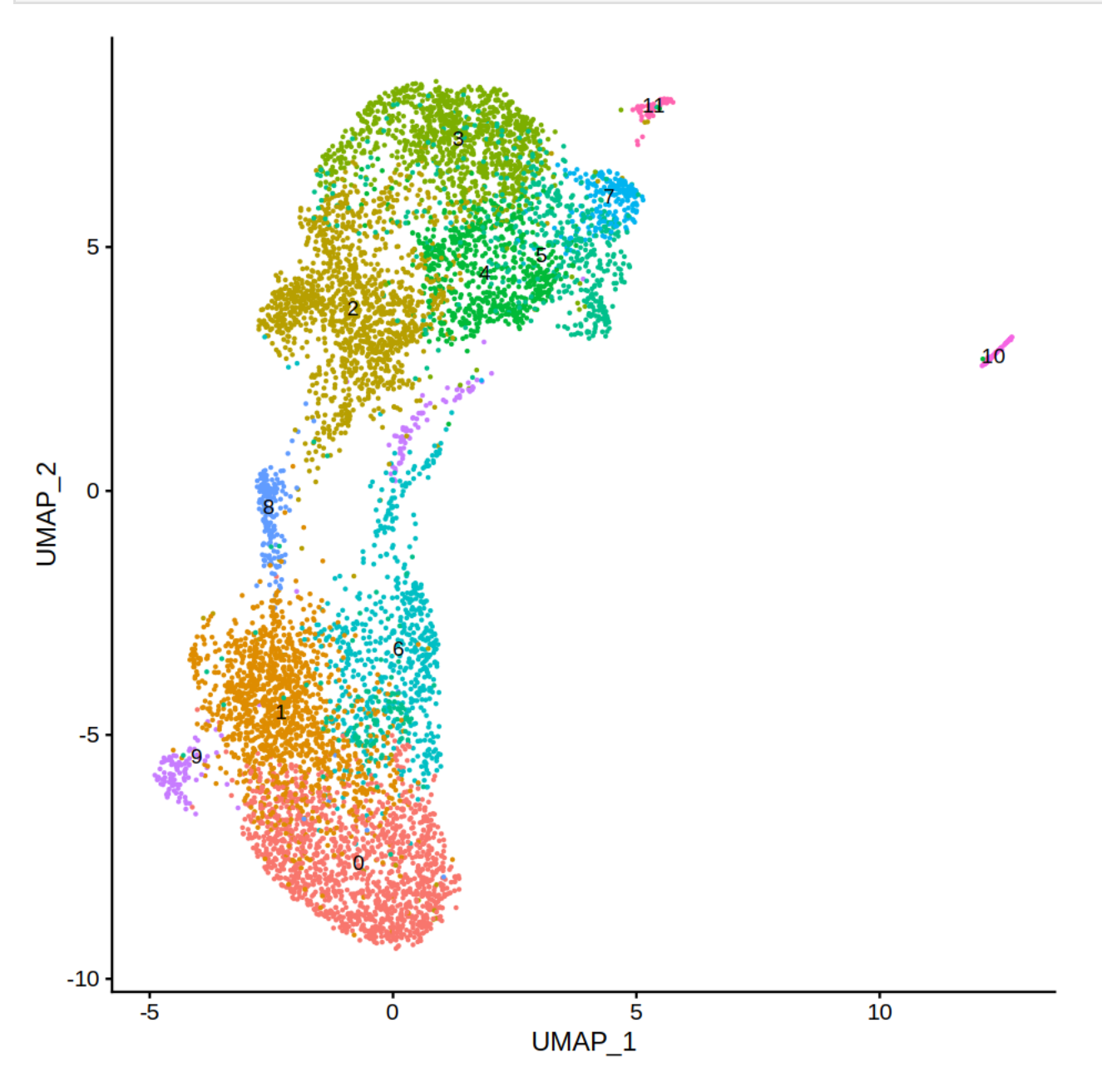
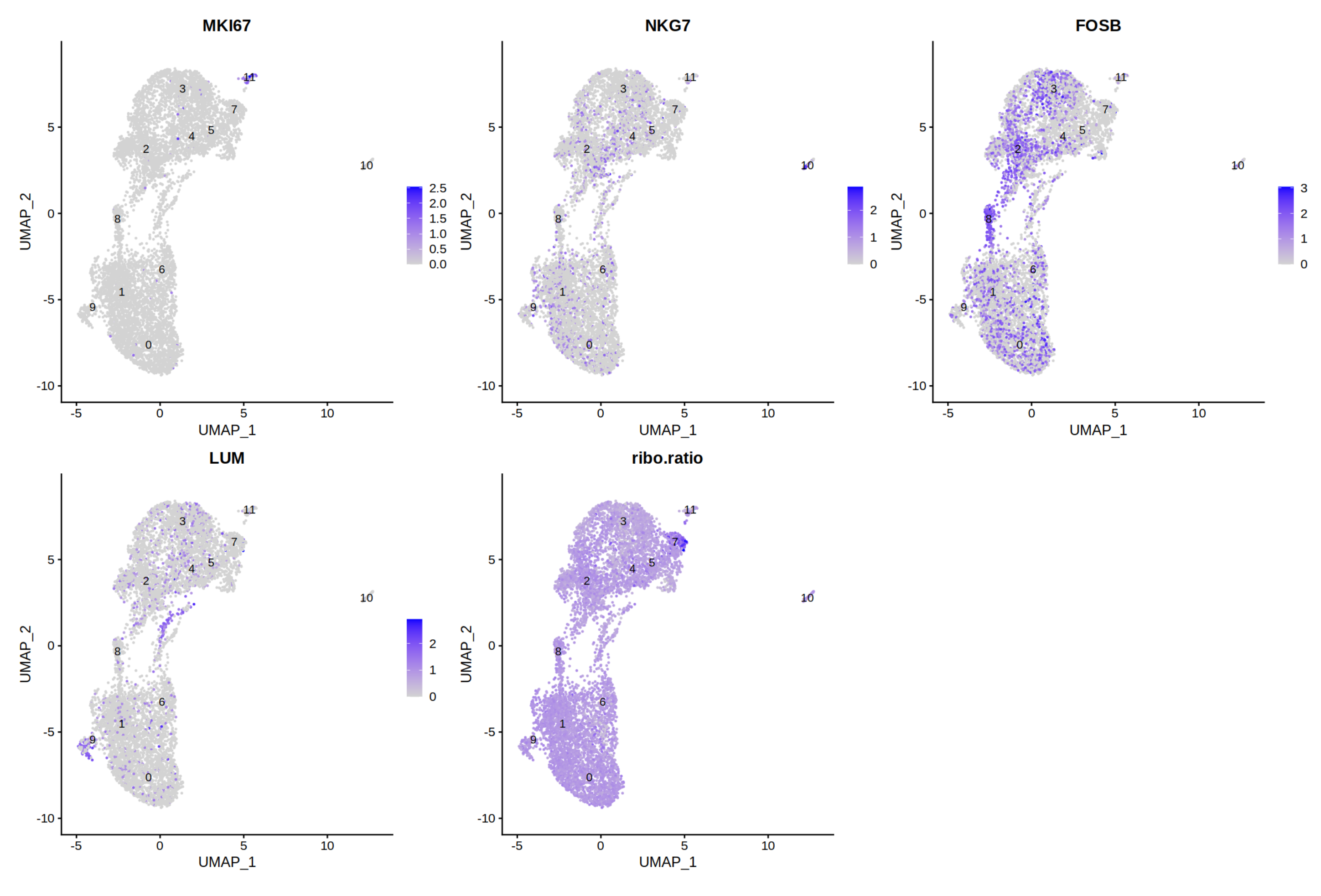
In [2]:

```r
library('Seurat')
library('data.table')
library('Matrix')
library('tidyverse')
library('ggplot2')
library('ggpubr')
library('biomaRt')
library('SeuratDisk')
library(org.Hs.eg.db)
library(stringr)
`%notin%` <- Negate(`%in%`)
library(pheatmap)
```

In [4]:

```r
# Endo <- LoadH5Seurat("/home/kevin/Data/Fibrosis/endo-2022_myeloid.h5seurat", assays = "RNA")

# Endo$integration <- as.character(Endo$PID)
# Endo$integration[ Endo$integration %in% c('C01', 'C02')] <- 'C0'
# Endo$integration[ Endo$integration %in% c('E10', 'E11')] <- 'EO'
# Endo <- Endo[,  Endo$subtypes %notin% c( 'mast cells', 'pDC', 'granulocytes', 'monocytes-CD16+', 'cDC1', 'cDC2',  'pre-cDC2', 'DC3', 'mDC', 'M$\\Phi$3-AP

# Endo <- Endo[ rowSums(Endo$RNA@counts > 0 )> 10, ]
# Endo$mito.ratio <- PercentageFeatureSet( Endo, pattern = '^MT-') / 100
# Endo$ribo.ratio <- (PercentageFeatureSet( Endo, pattern = '^RPS') +  PercentageFeatureSet( Endo, pattern = '^RPL') ) / 100
# Endo$heam.ratio <- (PercentageFeatureSet( Endo, pattern = '^HBB')  + PercentageFeatureSet( Endo, pattern = '^HBA')) / 100
# Endo <- Endo[, Endo$mito.ratio < .15 & Endo$ribo.ratio < .3 & Endo$heam.ratio < .001 & Endo$nFeature_RNA > 200 & Endo$nFeature_RNA < 5000]


# Endo <- SplitObject( Endo, split.by = 'integration')
# for(i in 1:length(Endo)) {
#     DefaultAssay( Endo[[i]]) <- 'RNA'
#     Endo[[i]] <- NormalizeData( Endo[[i]])
#     Endo[[i]] <- FindVariableFeatures( Endo[[i]] )

# }

# features <- SelectIntegrationFeatures(object.list = Endo)
# anchors <- FindIntegrationAnchors(object.list = Endo, dims = 1:20, anchor.features = features)
# Endo <- IntegrateData(anchorset = anchors)
# rm(anchors)
# memory.profile()

# Endo <- ScaleData(object = Endo, verbose = FALSE, vars.to.regress = c('nFeature_RNA', 'mito.ratio'))
# Endo <- RunPCA(object = Endo, npcs = 50, verbose = FALSE)
# Endo <- RunUMAP(object = Endo, reduction = "pca", dims = 1:20, verbose = F )

# DefaultAssay( Endo) <- "integrated"
# Endo <- FindNeighbors(Endo, reduction = "pca", dims = 1:20)
# Endo <- FindClusters(Endo, resolution = .5)

options(repr.plot.height = 8,  repr.plot.width = 8)
DimPlot(Endo, label = T) + NoLegend()

DefaultAssay( Endo) <- "RNA"
#Endo <- NormalizeData(Endo)

options(repr.plot.height = 12,  repr.plot.width = 18)
FeaturePlot(Endo, c('MKI67', 'NKG7', 'FOSB', 'LUM', 'ribo.ratio'), label = T, ncol = 3) + NoLegend()
```



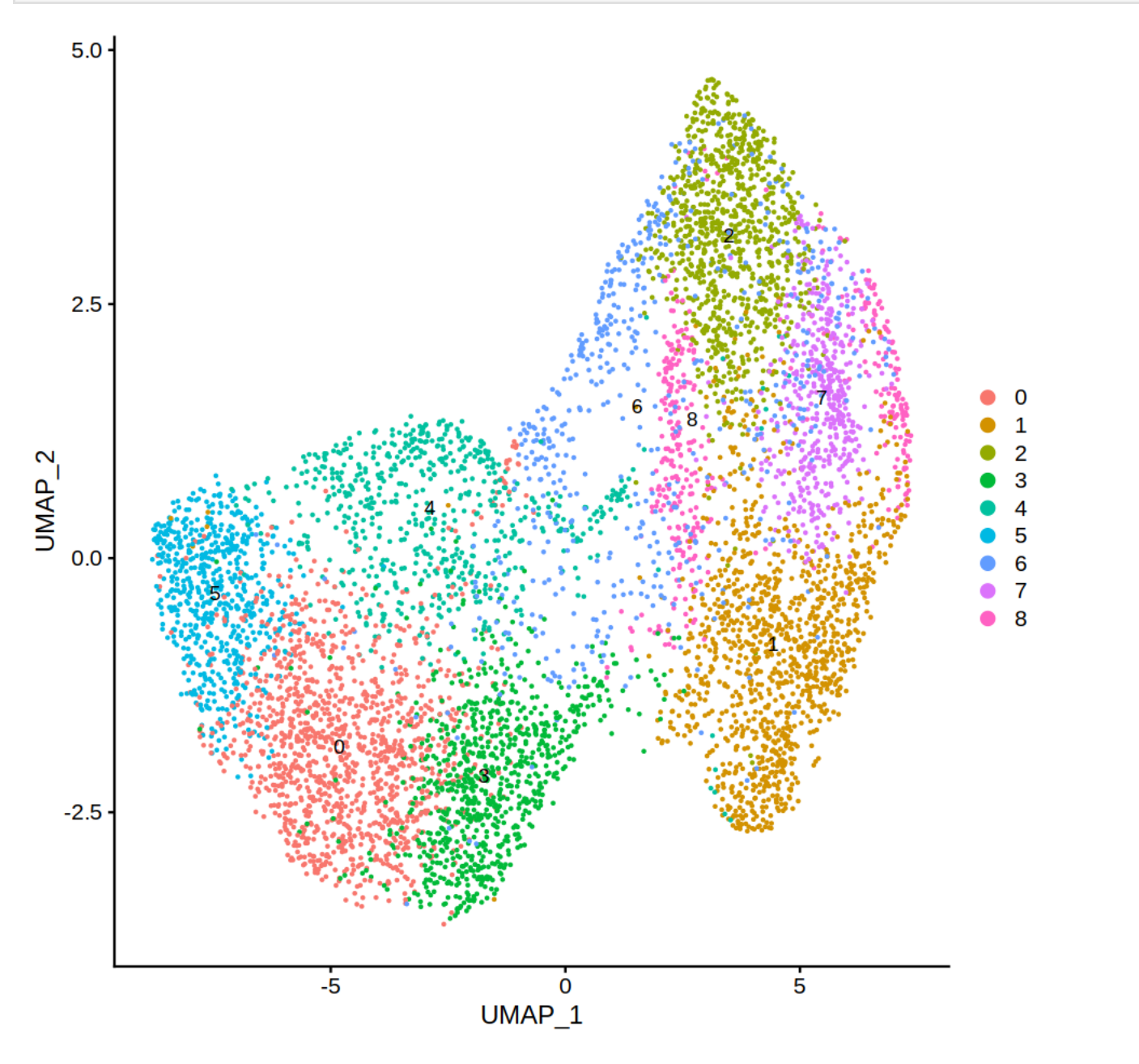


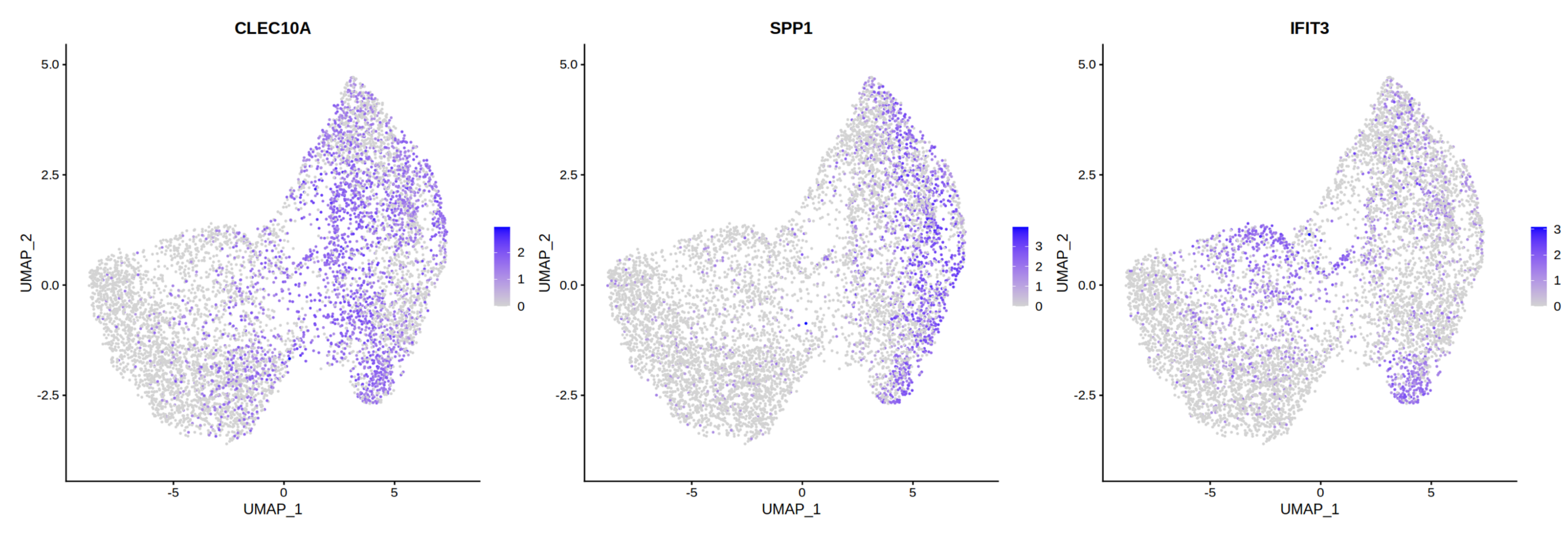
In [26]:

```r
t <- Endo
```

In [40]:

```r
# Endo <- Endo[, Idents(Endo) %notin% c(7, 8, 9, 10, 11)]

# Endo <- SplitObject( Endo, split.by = 'integration')
# for(i in 1:length(Endo)) {
#     DefaultAssay( Endo[[i]]) <- 'RNA'
#     Endo[[i]] <- NormalizeData( Endo[[i]])
#     Endo[[i]] <- FindVariableFeatures( Endo[[i]] )

# }

# features <- SelectIntegrationFeatures(object.list = Endo)
# anchors <- FindIntegrationAnchors(object.list = Endo, dims = 1:20, anchor.features = features)
# Endo <- IntegrateData(anchorset = anchors)
# rm(anchors)
# memory.profile()

# Endo <- ScaleData(object = Endo, verbose = FALSE, vars.to.regress = c('nFeature_RNA', 'mito.ratio'))
# Endo <- RunPCA(object = Endo, npcs = 50, verbose = FALSE)
# Endo <- RunUMAP(object = Endo, reduction = "pca", dims = 1:10, verbose = F, n.neighbors = 60 )

# DefaultAssay( Endo) <- "integrated"
# Endo <- FindNeighbors(Endo, reduction = "pca", dims = 1:10)
# Endo <- FindClusters(Endo, resolution = .5)

options(repr.plot.height = 8,  repr.plot.width = 8)
DimPlot(Endo, label = T)

options(repr.plot.height = 7,  repr.plot.width = 21)
FeaturePlot( Endo, c('CLEC10A', 'SPP1', 'IFIT3'), ncol = 3)
```

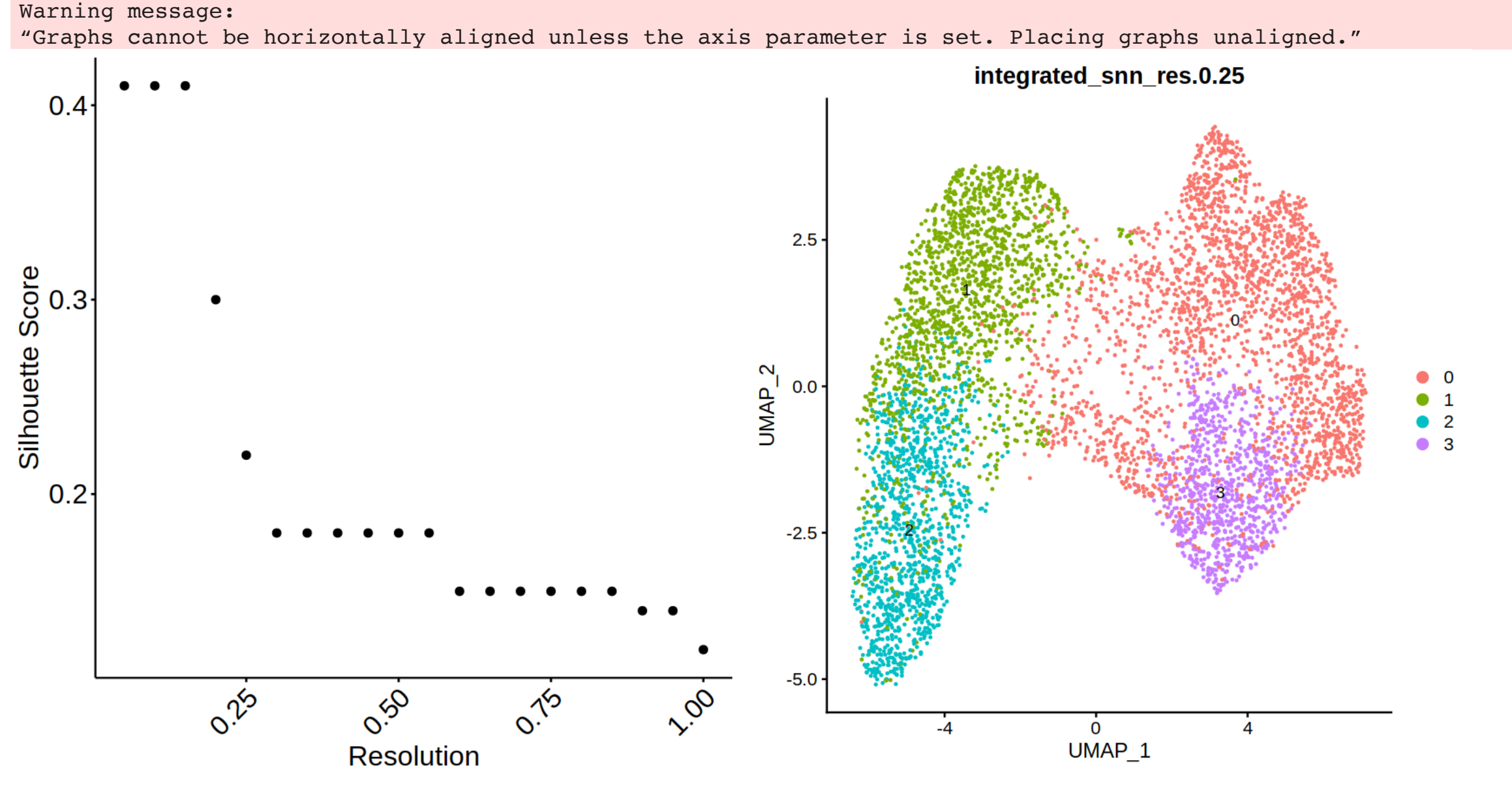




In [44]:

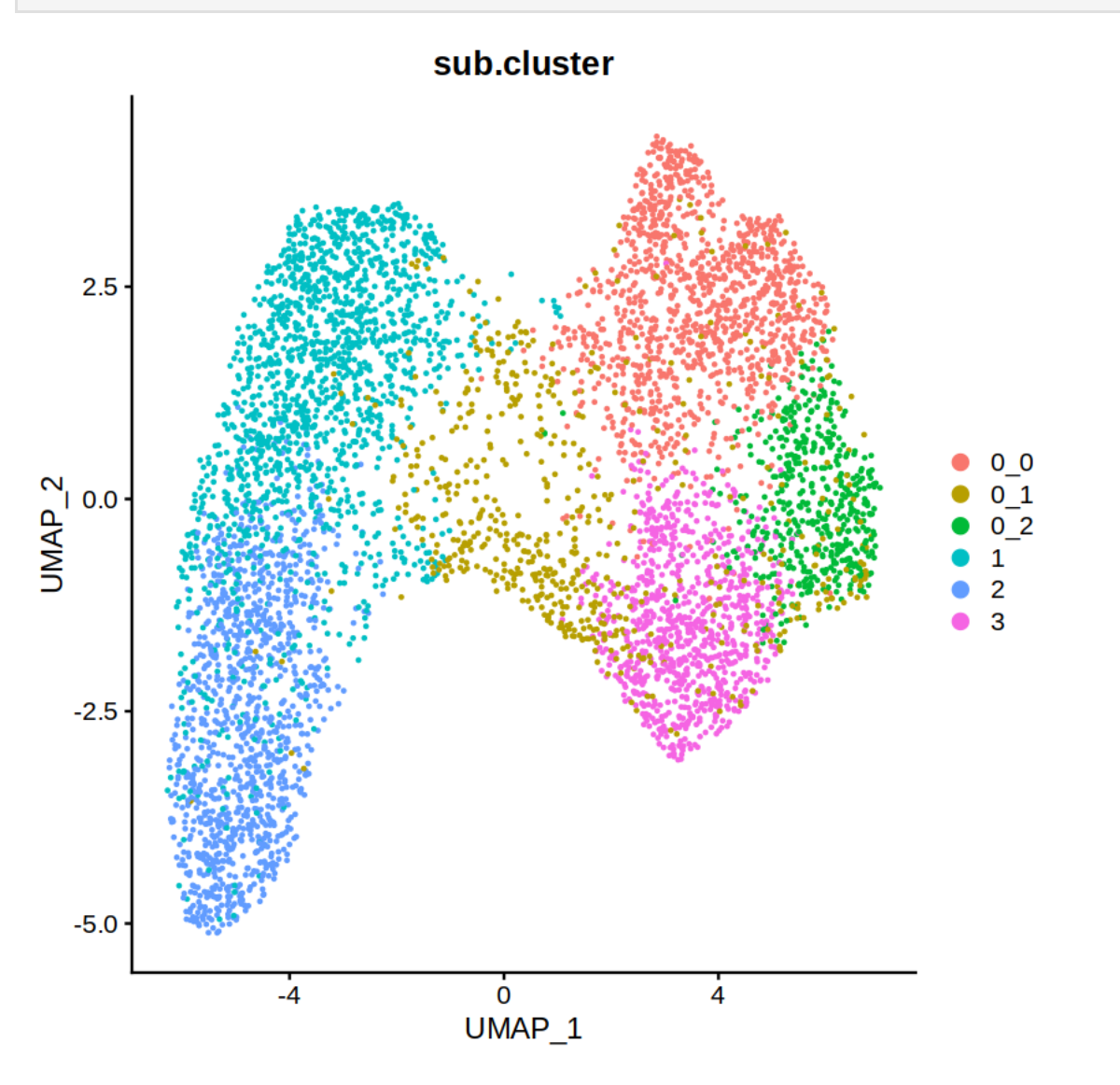
```r
# Endo <- Endo[, Idents(Endo) %notin%  c(8, 4)]

# DefaultAssay( Endo) <- "integrated"
# Endo <- RunPCA(Endo)
# Endo <- RunUMAP(Endo, dims = 1:10 )
# Endo <- FindNeighbors(Endo, reduction = "pca", dims = 1:10, k.param = 50)
# Endo <- FindClusters(Endo, resolution = seq(from = 0.05, to = 1, by = .05))


# DefaultAssay(Endo) <- 'integrated'
# resolution <- colnames( Endo@meta.data )[ startsWith(x =colnames( Endo@meta.data ), prefix = 'integrated_snn_res')]

# silhouetteee <- list()
# i <- 1
# dist.matrix <- dist(x = Embeddings(object = Endo[['pca']])[,1:10])
# for(sil in sort(resolution)) {
#     s <- cluster::silhouette(x = as.numeric(x = as.factor(x = Endo@meta.data[, sil])), dist = dist.matrix)
#     silhouetteee[[i]] <- summary(s)
#     i <- i + 1
# }

# sil_score <- c()
# for( sil in silhouetteee) {
#     sil_score <- c(sil_score, mean(sil$clus.avg.widths))
# }

# to_plot <- data.frame('Score' = sil_score[1: length(sil_score)], 'Res' = seq(from = 0.05, to = 1, by = .05))
# to_plot$Score <- round(to_plot$Score,digits = 2)

p2 <- ggscatter(data = to_plot, x = 'Res', y = 'Score', xlab = 'Resolution', ylab = 'Silhouette Score') +
 rotate_x_text(angle = 45) +
   theme(text = element_text(size = 19), plot.title = element_text(hjust = 0.5))


options(repr.plot.height = 7, repr.plot.width = 14)
p1 <- DimPlot(Endo, label = T, group.by = 'integrated_snn_res.0.25')
CombinePlots(list(p2, p1))
```

```
Warning message:
“CombinePlots is being deprecated. Plots should now be combined using the patchwork system.”
Warning message:
“Graphs cannot be vertically aligned unless the axis parameter is set. Placing graphs unaligned.”
Warning message:
“Graphs cannot be horizontally aligned unless the axis parameter is set. Placing graphs unaligned.”
```



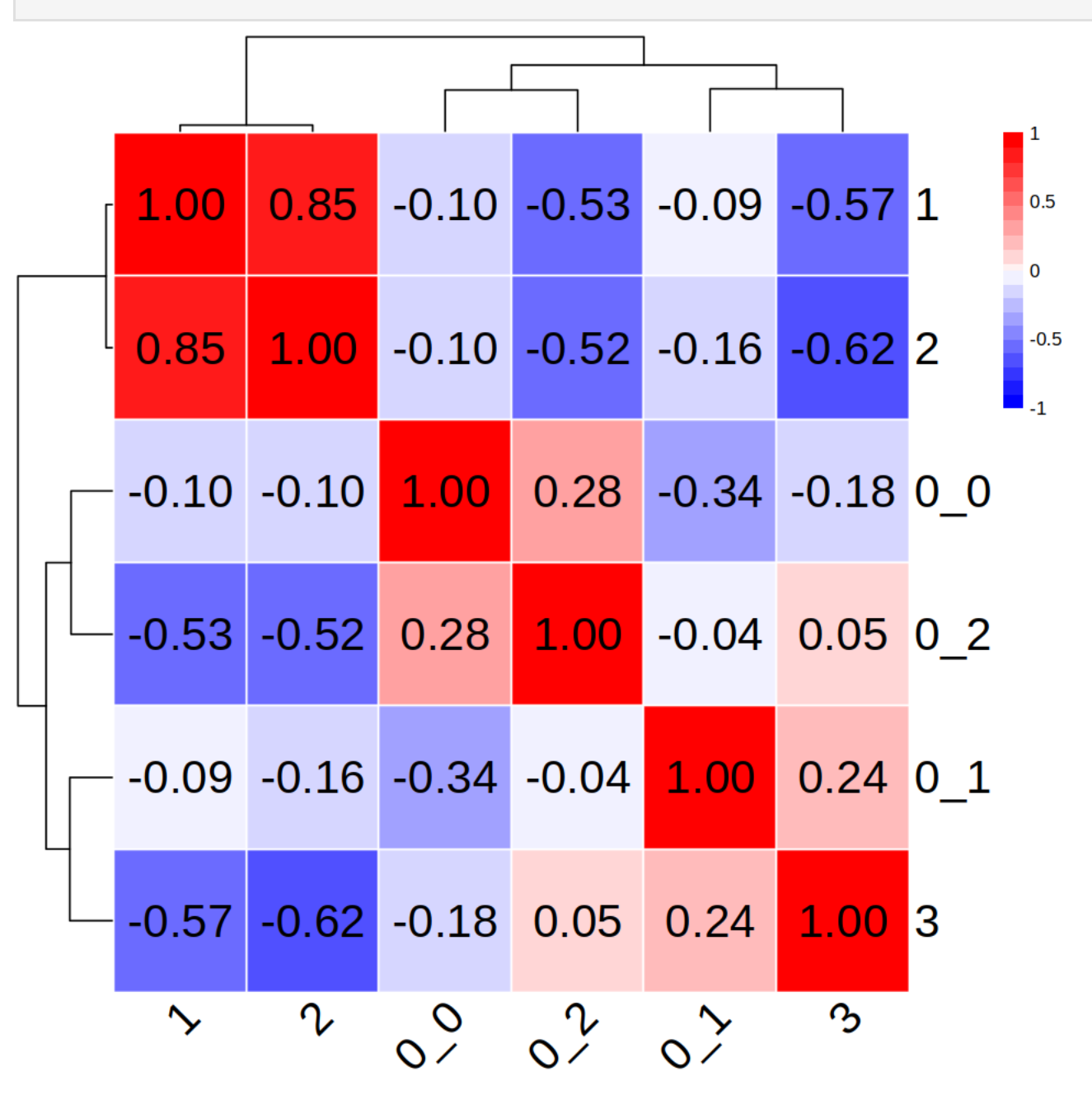
In [49]:

```r
# DefaultAssay( Endo) <- "integrated"
# Endo <- RunUMAP(Endo, dims = 1:10, n.neighbors = 60, verbose = F )
# Idents(Endo) <- Endo$integrated_snn_res.0.25
# Endo <- FindSubCluster( object = Endo, cluster = c(0), graph.name = 'integrated_snn', resolution = .25)


options(repr.plot.height = 7,  repr.plot.width = 7)
DimPlot(Endo, group.by = 'sub.cluster')
```



In [50]:

```r
DefaultAssay(Endo) <- 'RNA'
Endo_bulk <- AverageExpression(Endo, group.by = 'sub.cluster', assays = 'RNA', features = Endo$integrated@var.features )$RNA
Endo_bulk <- Endo_bulk / rowSums(Endo_bulk); Endo_bulk[ rowSums( is.na(Endo_bulk)  ) > 0 ] <- NULL
corr <- cor(x = Endo_bulk,  y =Endo_bulk,  method = 'spearman')

options(repr.plot.height = 8, repr.plot.width = 8)


paletteLength <- 20
myColor <- colorRampPalette(c("blue", "white", "red"))(paletteLength)
myBreaks <- c(seq(-1, 0, length.out=ceiling(paletteLength/2) + 1), seq(max(corr)/paletteLength, max(corr), length.out=floor(paletteLength/2)))

pheatmap(corr, display_numbers = T, breaks = myBreaks, color=myColor, number_format = '%.02f', border_color = 'white', number_color = 'black',
         fontsize_number = 25, fontsize_row = 25, fontsize_col = 25,  angle_col = 45, cluster_rows = T, cluster_cols = T)

Endo$merged <- Endo$sub.cluster
Endo$merged[ Endo$merged %in% c(1, 2)] <- 2
```

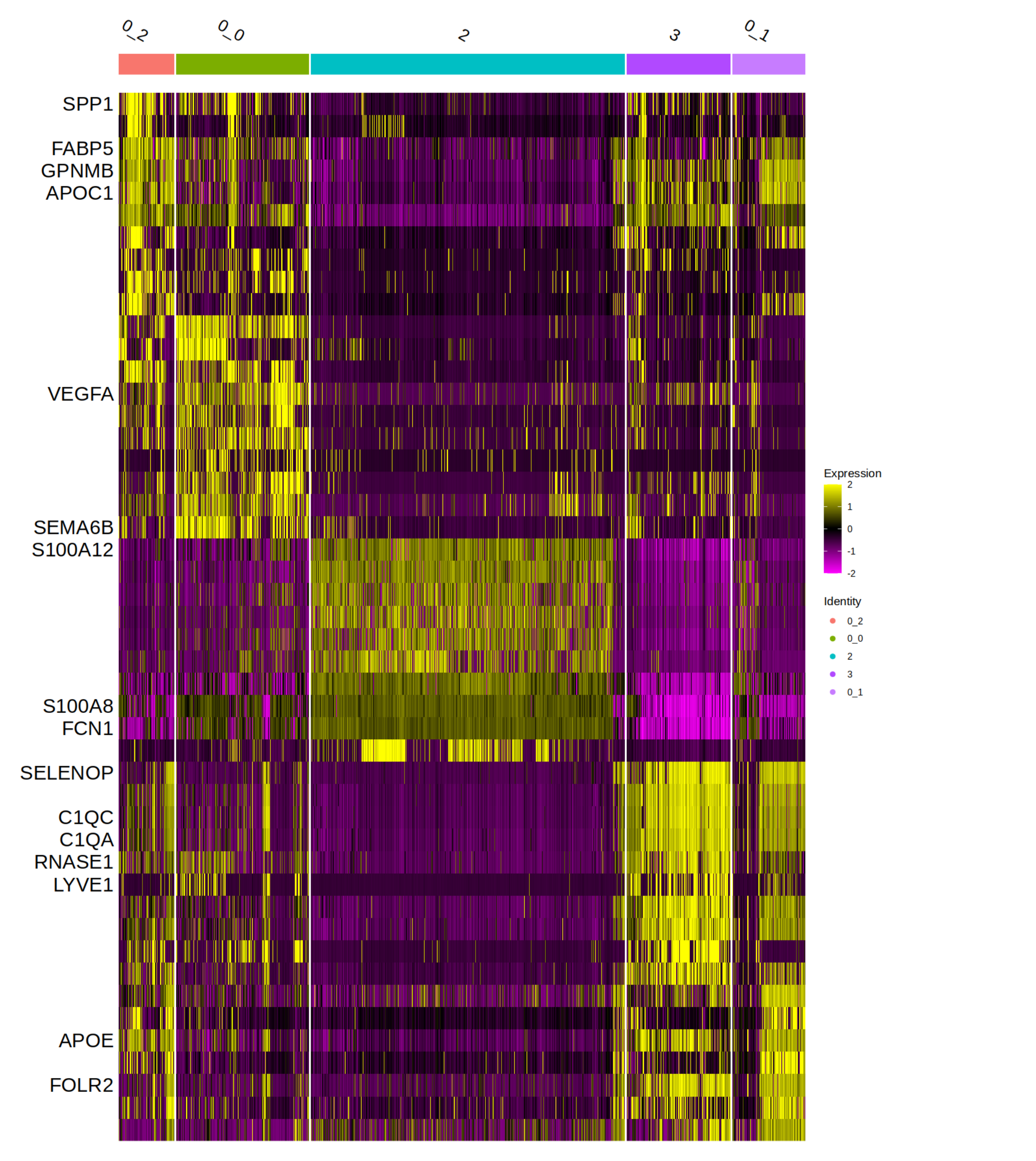


	1	2	0_0	0_2	0_1	3
1	1.00	0.85	-0.10	-0.53	-0.09	-0.57
2	0.85	1.00	-0.10	-0.52	-0.16	-0.62
0_0	-0.10	-0.10	1.00	0.28	-0.34	-0.18
0_2	-0.53	-0.52	0.28	1.00	-0.04	0.05
0_1	-0.09	-0.16	-0.34	-0.04	1.00	0.24
3	-0.57	-0.62	-0.18	0.05	0.24	1.00

In [54]:

```r
# Idents(Endo) <-Endo$merged
# DefaultAssay(Endo) <- 'RNA'
# Endo <- NormalizeData(Endo)
# Endo <- FindVariableFeatures(Endo)
# Endo <- ScaleData(Endo, vars.to.regress = c('mito.ratio', 'nFeature_RNA'))
# Endo.m <- FindAllMarkers(Endo, only.pos = T, min.diff.pct = .1)
# Endo.m <- Endo.m[ order( Endo.m$avg_log2FC, decreasing = T), ]

tmp <- Endo
DefaultAssay(tmp) <- 'RNA'
Idents(tmp) <- as.character(tmp$merged)
tmp$try <- factor( tmp$merged , levels = c('0_2', '0_0',   2,  3, '0_1')); res <- 'try'

Heart_all_prev_m <- Endo.m [ Endo.m $pct.1 - Endo.m $pct.2 > .2 & Endo.m $pct.1 > .4,]

tmp_8_gene <- c()
for(clus in intersect(levels(tmp$try), unique(Heart_all_prev_m$cluster))) {
    tmp_8_gene <- c(tmp_8_gene, Heart_all_prev_m[ Heart_all_prev_m$cluster == clus, ]$gene[1:10])
}


genes.to.label <- c('SELENOP','C1QA','RNASE1' ,'C1QC','FCN1', 'S100A9','S100A8','VCAN', 'APOE','GPNMB','CD9','STAB1','SPP1', 'FABP5', 'TREM2', 'F13A1',
                   'CCL20', 'FOLR2', 'INHBA',  'CHI3L1', 'SPP1',  'MATK','PTGS2','IL1B', 'FABP4', 'LYVE1', 'APOC1', 'CCL4','S100A12',
                   'CCL3', 'DAB2','LYVE1','CCL4L2', 'PHLDA3', 'EGR1', 'ATF3', 'VEGFA', 'SEMA6B', 'IFI44L', 'IFIT3', 'IFIT1')
all.genes <- intersect(tmp_8_gene, rownames( tmp$RNA@scale.data))
labels <- rep(x = "transparent", times = length(x = all.genes))
labels[match(x = genes.to.label, table = all.genes)] <- "black"



options(repr.plot.height = 16, repr.plot.width = 12)
DoHeatmap( tmp, group.colors = c('#F8766D','#7CAE00','#00BFC4', '#b149ff','#c77cff', '#ddafff'),
        features = tmp_8_gene, slot = 'scale.data', assay = 'RNA', disp.min = -2, disp.max = 2,
          angle = -30, hjust = 1, group.by = res) +
  theme(axis.text.y = element_text(color = rev(x = labels), size = 19))
```

```
Warning message:
“Vectorized input to `element_text()` is not officially supported.
Results may be unexpected or may change in future versions of ggplot2.”
```



In [59]:

```r
Endo@meta.data[, colnames(Endo@meta.data) %in%
                colnames(Endo@meta.data)[ startsWith( colnames(Endo@meta.data), prefix = 'integrated')]] <- NULL

Endo$cluster_d <- plyr::mapvalues( Endo$merged, from = c('0_2', '0_0',   2,   3, '0_1'), to = c('SPP1', 'VEGFA Trans', 'FCN1', 'RNASE1', 'TRANS'))

saveRDS(Endo, '~/Endo_no_Ifit.RDS')
```