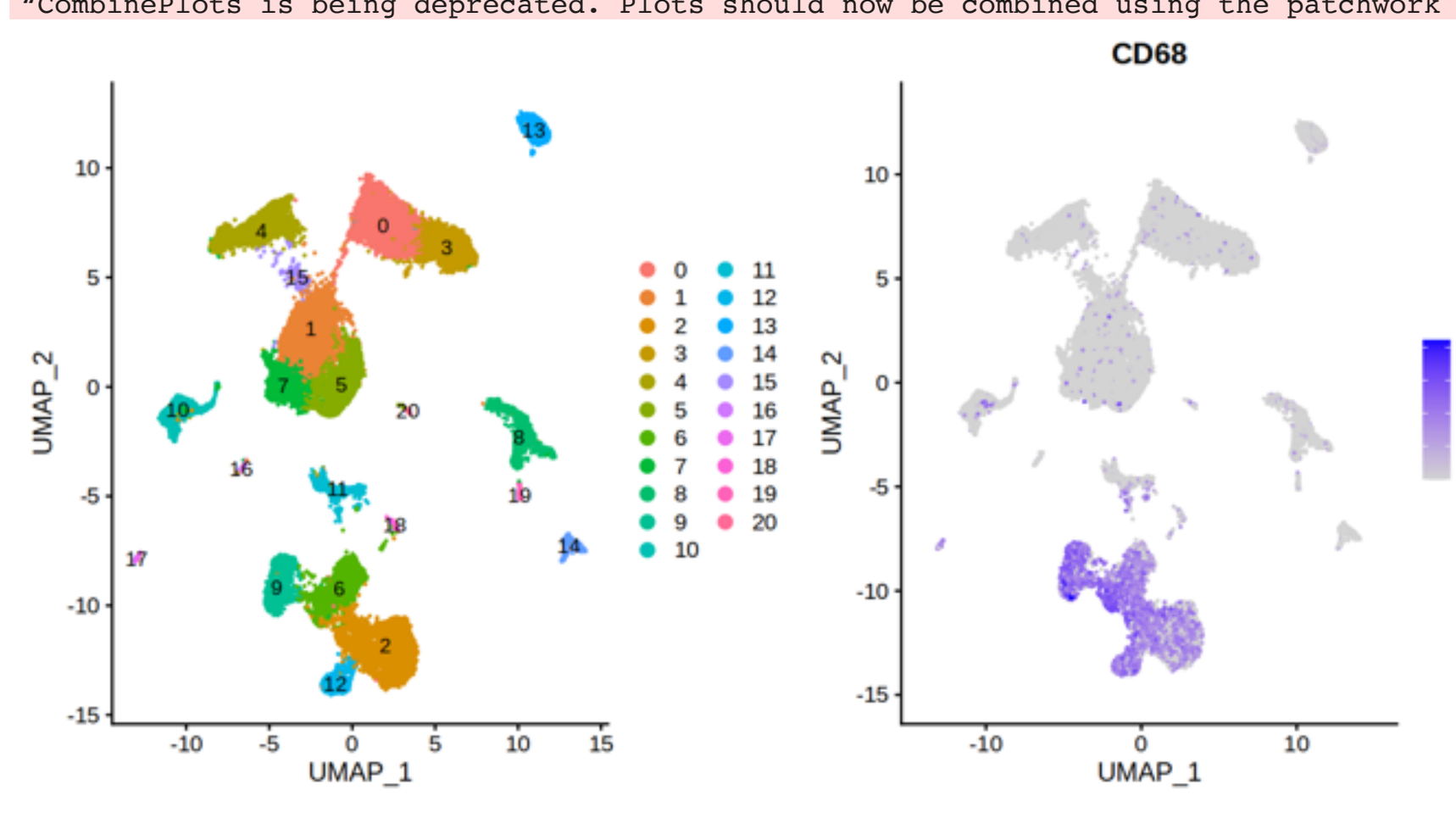
```
In [21]:
library(Seurat)
library(stringr)
library(SeuratDisk)
`%notin%` <- Negate(`%in%`)
library(harmony)
library(Matrix)
library(pheatmap)
library(cluster)
library(ggpubr)
```

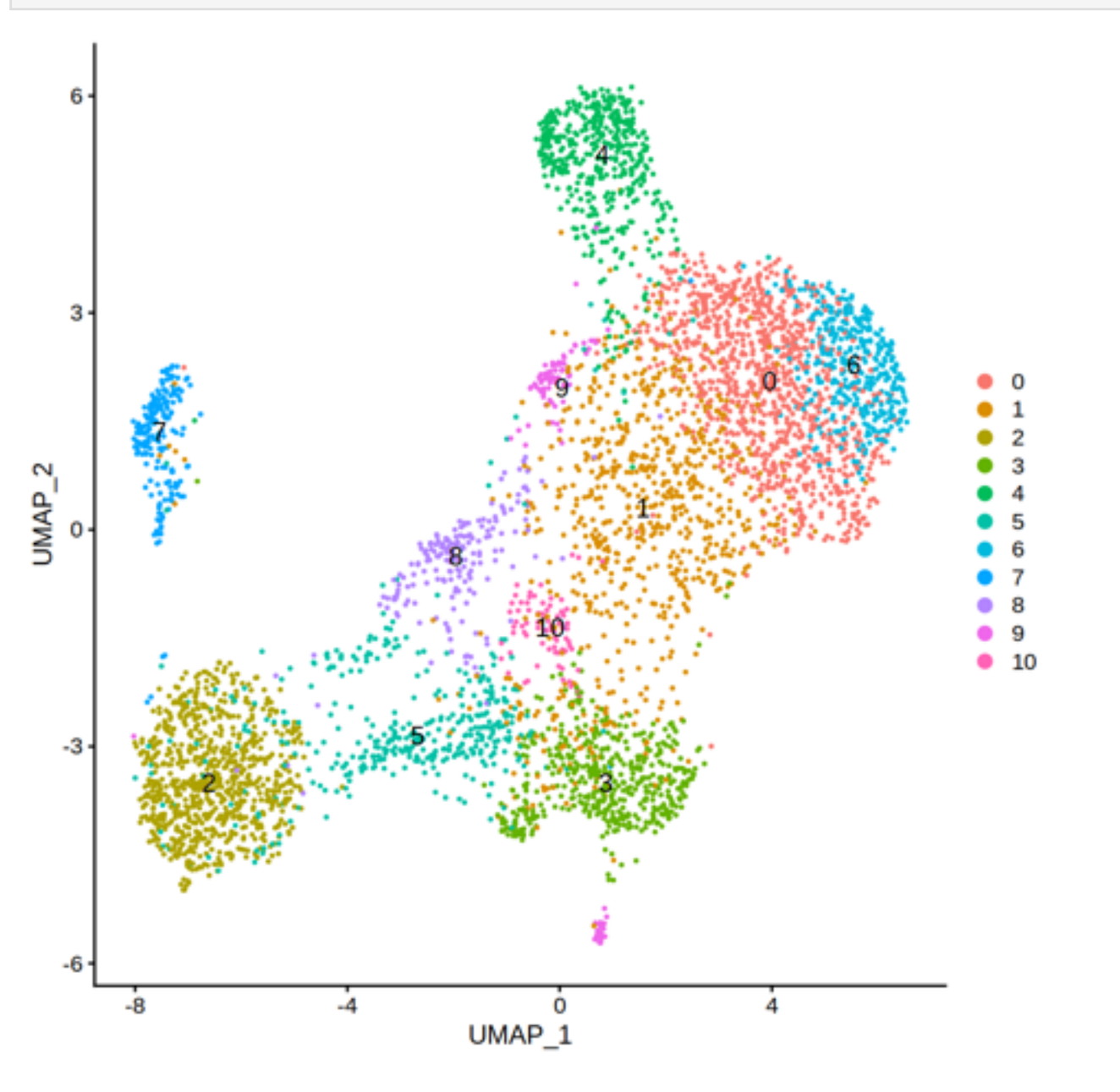
Read in data

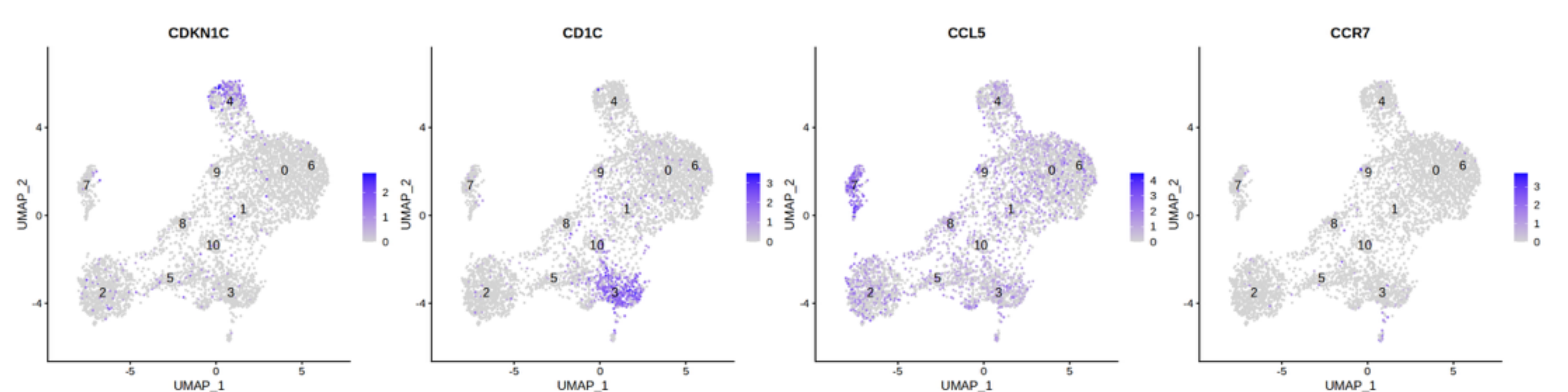
```
In [4]:
# dir <- '/mnt/storage/projects/huang_macrophage/liver/GSE136103/'
# file <- c('GSM4041150_healthy1_cd45+','GSM4041153_healthy2_cd45+', 'GSM4041155_healthy3_cd45+','GSM4041158_healthy4_cd45+', 'GSM4041160_he
#    'GSM4041161_cirrhotic1_cd45+', 'GSM4041164_cirrhotic2_cd45+', 'GSM4041166_cirrhotic3_cd45+', 'GSM4041168_cirrhotic4_cd45+', 'GSM404116

# Liver <- list()
# for(i in 1: length(file)) {
#     counts <- ReadMtx(mtx = paste0(dir, file[i], '_matrix.mtx.gz'), cells = paste0(dir, file[i], '_barcodes.tsv.gz'),  features = paste0(c
#     Liver[[i]] <- CreateSeuratObject(counts = counts, project = file[i],  min.features = 200)
#     Liver[[i]]$mito.ratio <- PercentageFeatureSet(Liver[[i]], pattern = "^MT-") / 100
#     Liver[[i]]$orig.ident <- file[[i]]
#     Liver[[i]] <- Liver[[i]][, Liver[[i]]$mito.ratio < .15 ]
#     rm(counts)
# }
# memory.profile()

# Liver <- merge( Liver[[1]], Liver[2: length(Liver)])
# DefaultAssay(Liver) <- 'RNA'

# Liver$rpl <- PercentageFeatureSet( Liver, pattern = 'RPL') / 100 - PercentageFeatureSet(  Liver, pattern = 'RPL*P') / 100
# Liver$rps <-  PercentageFeatureSet( Liver, pattern = 'RPS') / 100 - PercentageFeatureSet( Liver, pattern = 'RPs*P') / 100
# Liver$ribo.ratio <-  Liver$rpl + Liver$rps
# Liver$heam.ratio <- ( PercentageFeatureSet( Liver, pattern = '^HBB')  + PercentageFeatureSet( Liver, pattern = '^HBA')) / 100
# Liver <- Liver[, Liver$mito.ratio < .15 & Liver$ribo.ratio < .3 & Liver$heam.ratio < .001 & Liver$nFeature_RNA < 5000  ]

# Liver <- SplitObject(Liver, split.by = 'orig.ident')
# for( i in 1:length( Liver)) {
#     Liver[[i]] <- NormalizeData(Liver[[i]])
#     Liver[[i]] <- FindVariableFeatures( Liver[[i]] )
# }

# anchors <- FindIntegrationAnchors(object.list = Liver, dims = 1:30)
# Liver <- IntegrateData(anchorset = anchors, dims = 1:30)
# Liver <- ScaleData(Liver, verbose = FALSE, vars.to.regress = c('mito.ratio', 'nFeature_RNA'))
# rm(anchors)
# memory.profile()

# Liver <- RunPCA(Liver)
# Liver <- RunUMAP(Liver, dims = 1:30, verbose = F)

# DefaultAssay(Liver) <- 'integrated'
# Liver <- FindNeighbors(Liver, dims = 1:30)
# Liver <- FindClusters( Liver, resolution = .5)

options(repr.plot.width=12, repr.plot.height=6)
p1 <- DimPlot(Liver,label = T)
DefaultAssay( Liver) <- 'RNA'
Liver <- NormalizeData(Liver)
p2 <- FeaturePlot( Liver, 'CD68')
CombinePlots(list(p1, p2))
```

```
Warning message:
“CombinePlots is being deprecated. Plots should now be combined using the patchwork system.”
```

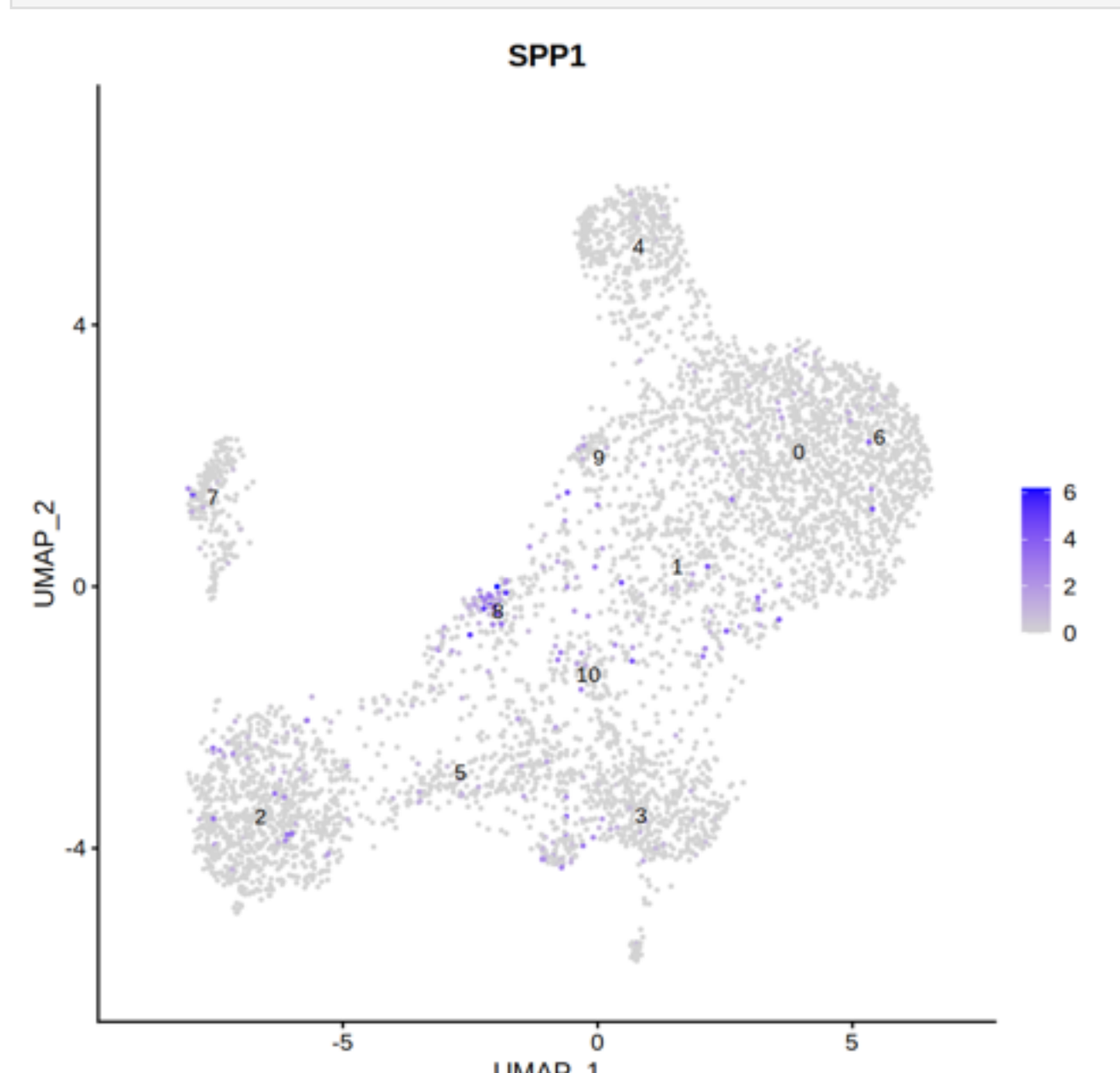



```
In [6]:
# Liver <- Liver[, Idents(Liver) %in% c(2, 6, 9, 12)]

# Liver <- SplitObject(Liver, split.by = 'orig.ident')
# for( i in 1:length( Liver)) {
#     Liver[[i]] <- NormalizeData(Liver[[i]])
#     Liver[[i]] <- FindVariableFeatures( Liver[[i]] )
# }

# anchors <- FindIntegrationAnchors(object.list = Liver, dims = 1:30)
# Liver <- IntegrateData(anchorset = anchors, dims = 1:30)
# Liver <- ScaleData(Liver, verbose = FALSE, vars.to.regress = c('mito.ratio', 'nFeature_RNA'))
# rm(anchors)
# memory.profile()

# Liver <- RunPCA(Liver)
# Liver <- RunUMAP(Liver, dims = 1:20, verbose = F)

# DefaultAssay(Liver) <- 'integrated'
# Liver <- FindNeighbors(Liver, dims = 1:20)
# Liver <- FindClusters( Liver, resolution = .8)

options(repr.plot.width=8, repr.plot.height=8)
DimPlot(Liver, label = T, label.size = 5)

DefaultAssay(Liver) <- 'RNA'
Liver <- NormalizeData(Liver)

options(repr.plot.width=24, repr.plot.height=6)
FeaturePlot(Liver, label = T, label.size = 5, features = c('CDKN1C', 'CD1C', 'CCL5', 'CCR7'), ncol = 4)
```

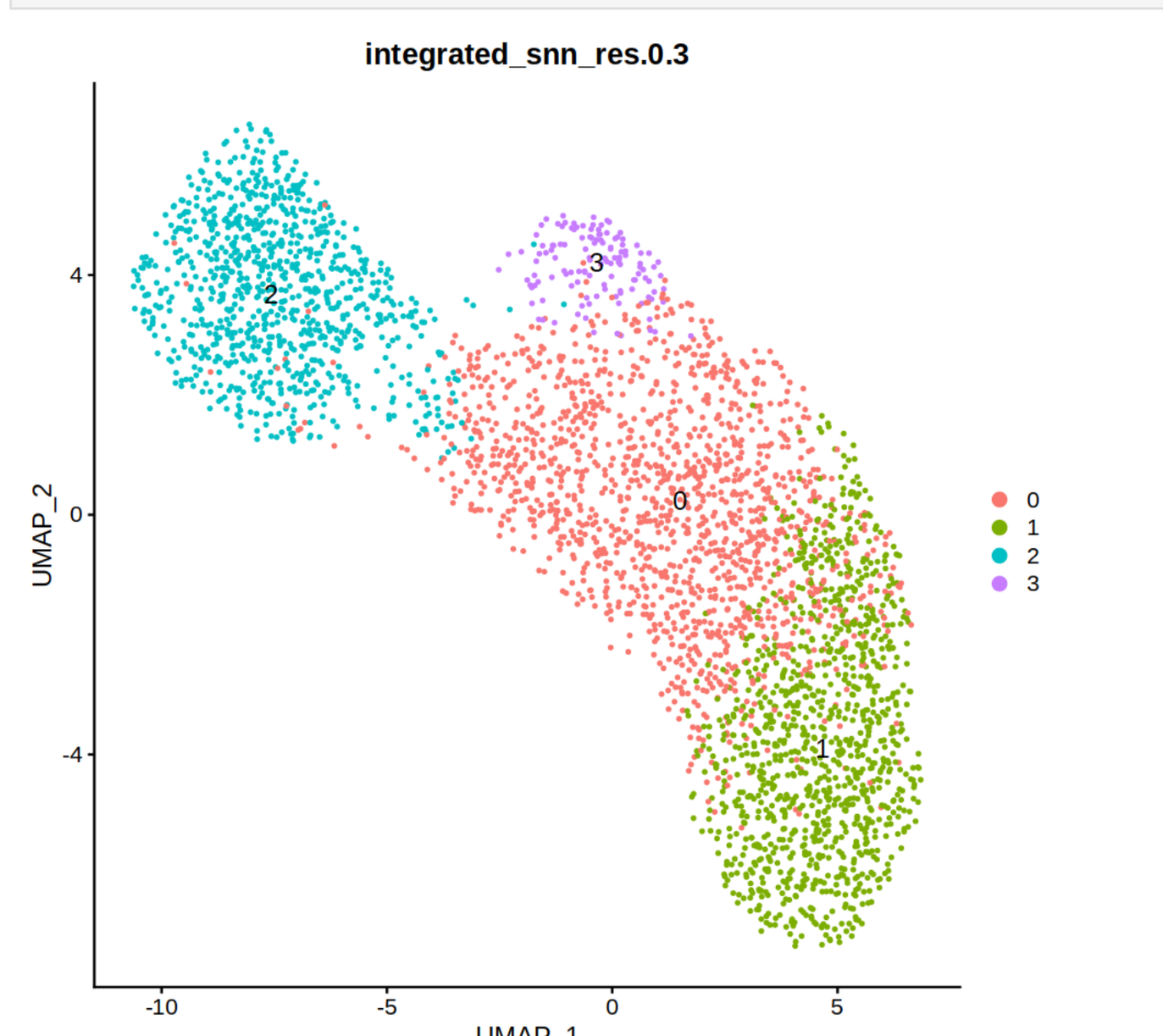


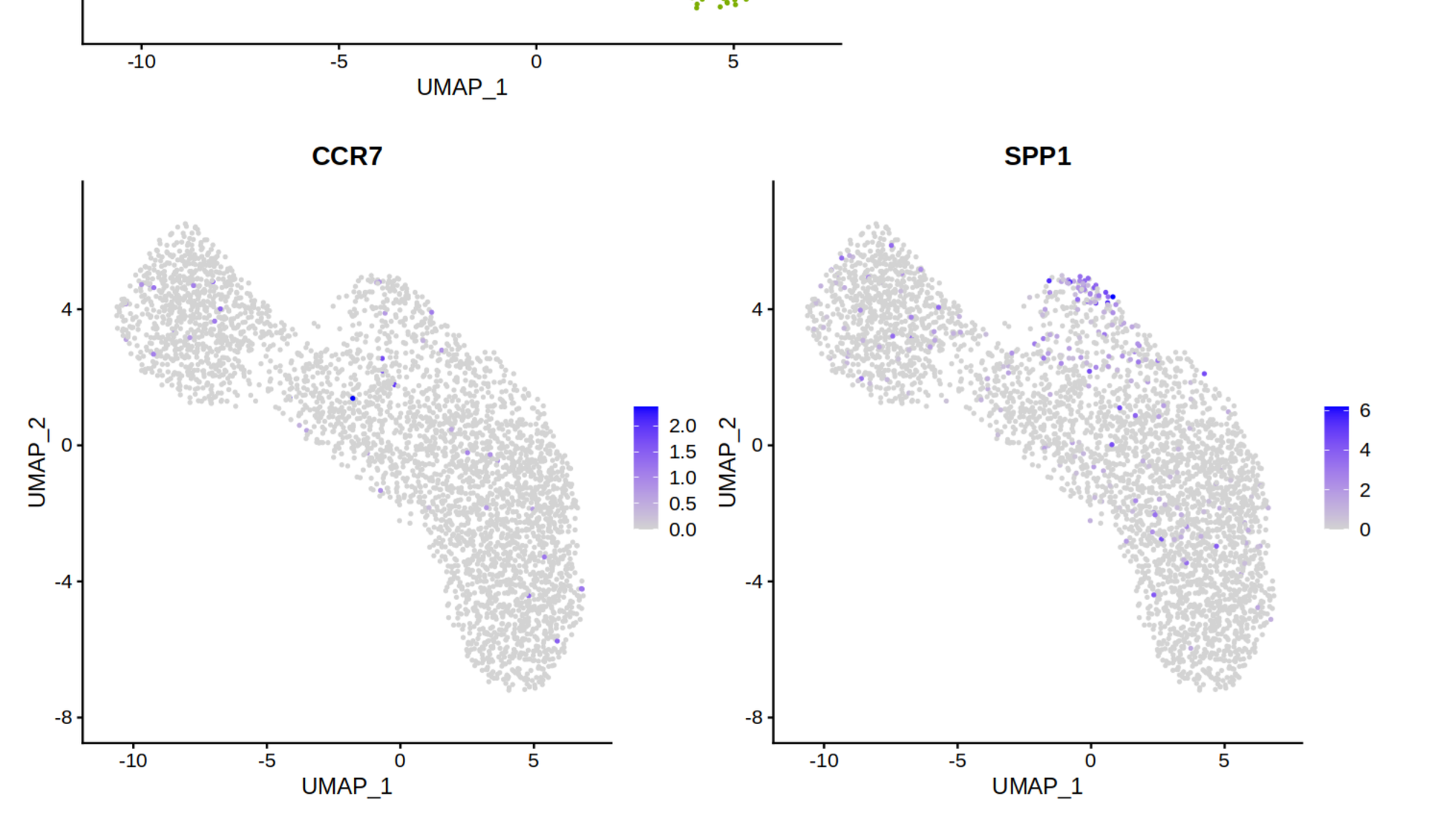



```
In [10]:
#Liver <- t
options(repr.plot.width=8, repr.plot.height=8)
FeaturePlot(Liver, 'SPP1', label = T)
```



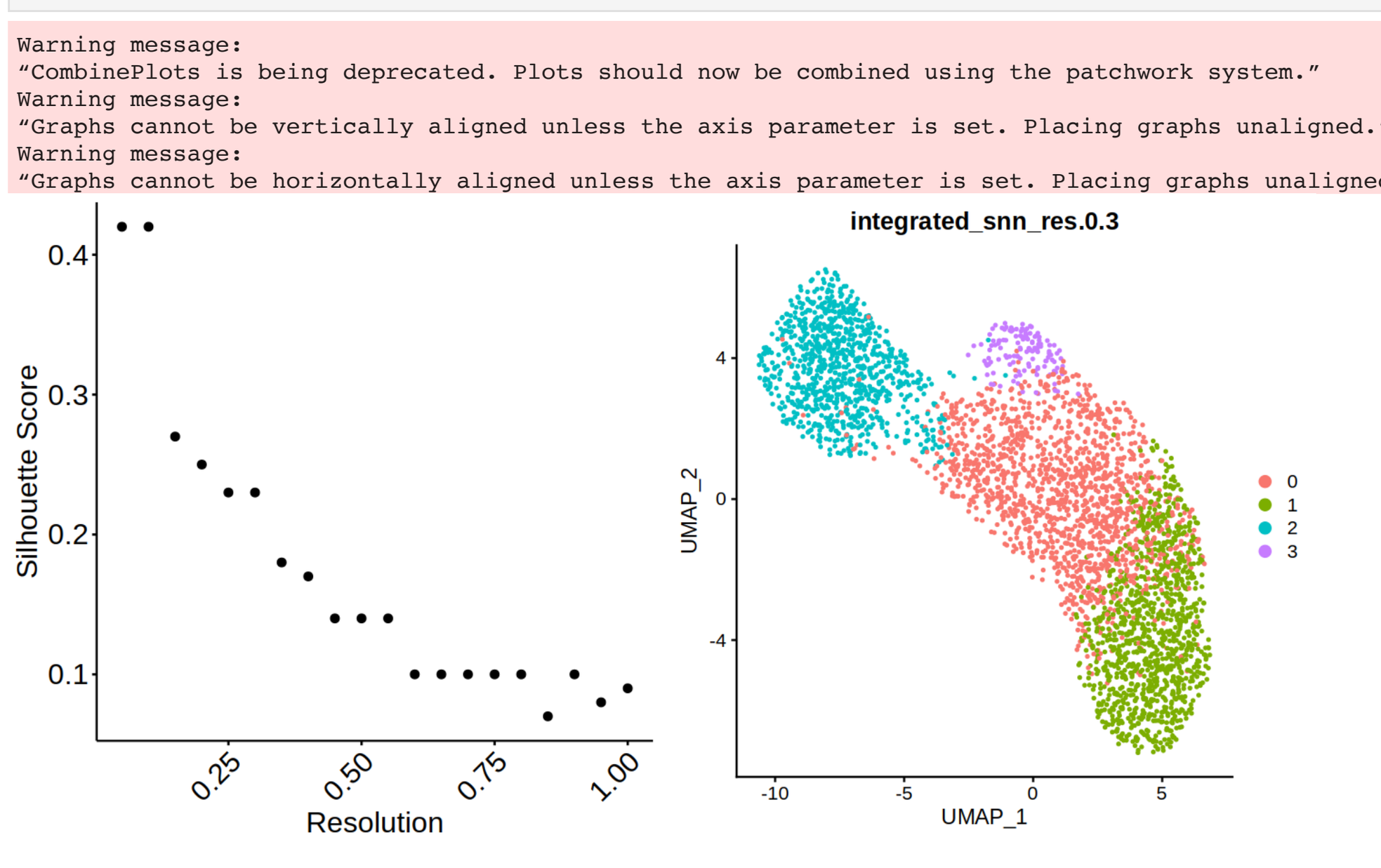

```
In [16]:
# Liver <- Liver[, Idents(Liver) %notin% c(3, 4, 7, 9)]
# DefaultAssay(Liver ) <- 'RNA'
# Liver <- Liver[ rowSums( Liver$RNA@counts >0) > 10, ]

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

# anchors <- FindIntegrationAnchors(object.list = Liver, dims = 1:20)
# Liver <- IntegrateData(anchorset = anchors, dims = 1:20)
# Liver <- ScaleData(Liver, verbose = FALSE, vars.to.regress = c('mito.ratio', 'nFeature_RNA'))
# rm(anchors)
# memory.profile()


# Liver <- RunPCA(Liver)
# Liver <- RunUMAP(Liver, dims = 1:12, verbose = F, min.dist = .6)

# DefaultAssay(Liver) <- 'integrated'
# Liver <- FindNeighbors(Liver, dims = 1:12)
# Liver <- FindClusters( Liver, resolution = .25)

options(repr.plot.width=8, repr.plot.height=8)
DimPlot(Liver, label = T, label.size = 5, group.by = 'integrated_snn_res.0.3')

options(repr.plot.width=12, repr.plot.height=6)
DefaultAssay(Liver) <- 'RNA'
FeaturePlot( Liver, c('CCR7', 'SPP1'), ncol = 2)
```





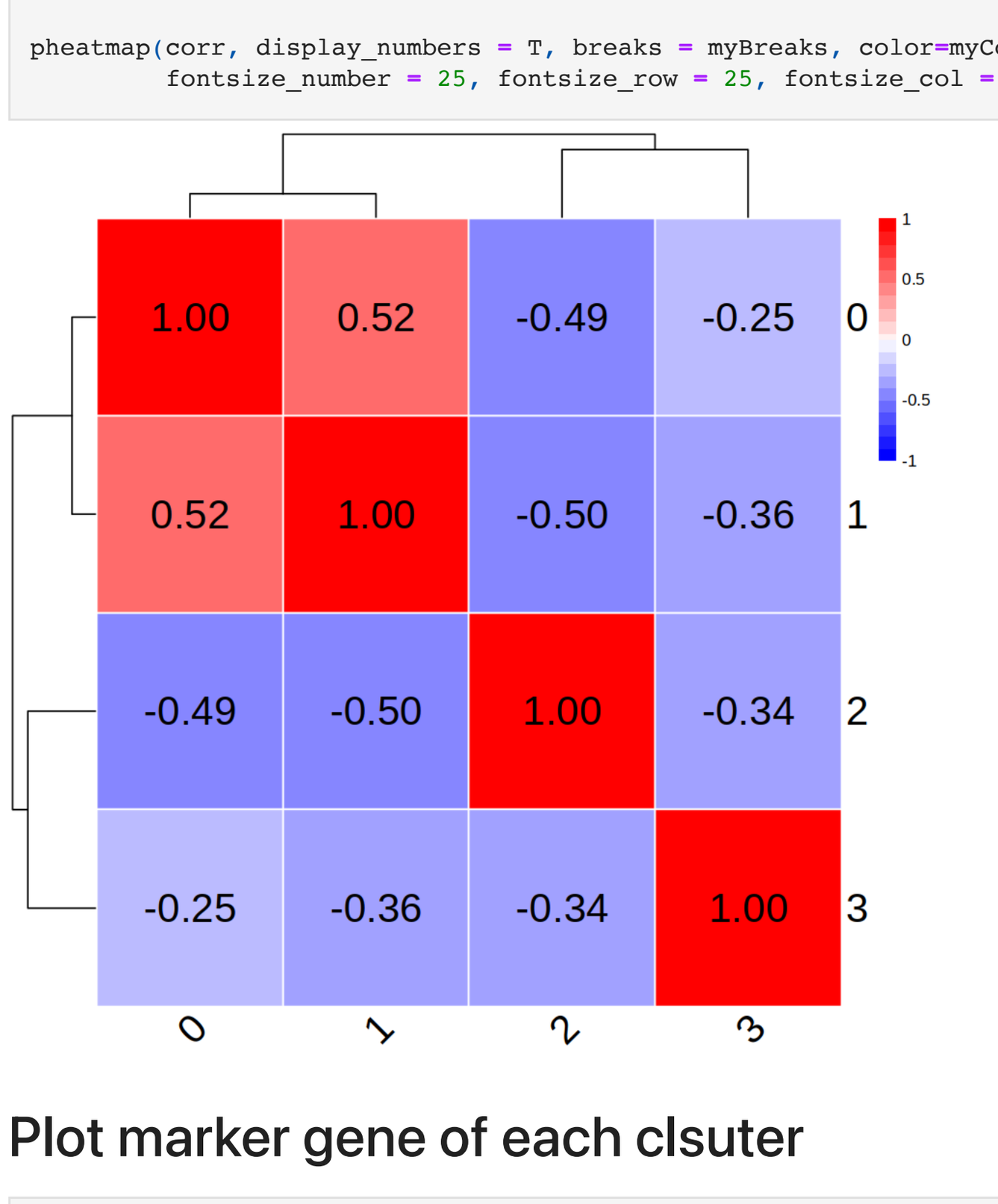

```
In [12]:
# DefaultAssay( Liver) <- "integrated"
# Liver <- FindNeighbors(Liver, dims = 1:12)
# Liver <- FindClusters(Liver, resolution = seq(from = 0.05, to = 1, by = .05))

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

# silhouetteee <- list()
# i <- 1
# dist.matrix <- dist(x = Embeddings(object = Liver[['pca']])[,1:12])
# for(sil in sort(resolution)) {
#     s <- silhouette(x = as.numeric(x = as.factor(x = Liver@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)

# p1 <- 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))
# p2 <- DimPlot(Liver, group.by = 'integrated_snn_res.0.3')

options(repr.plot.width=12, repr.plot.height=6)
CombinePlots(list(p1, p2))
```

```
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.”
```



Merge clusters with similiar transcriptome

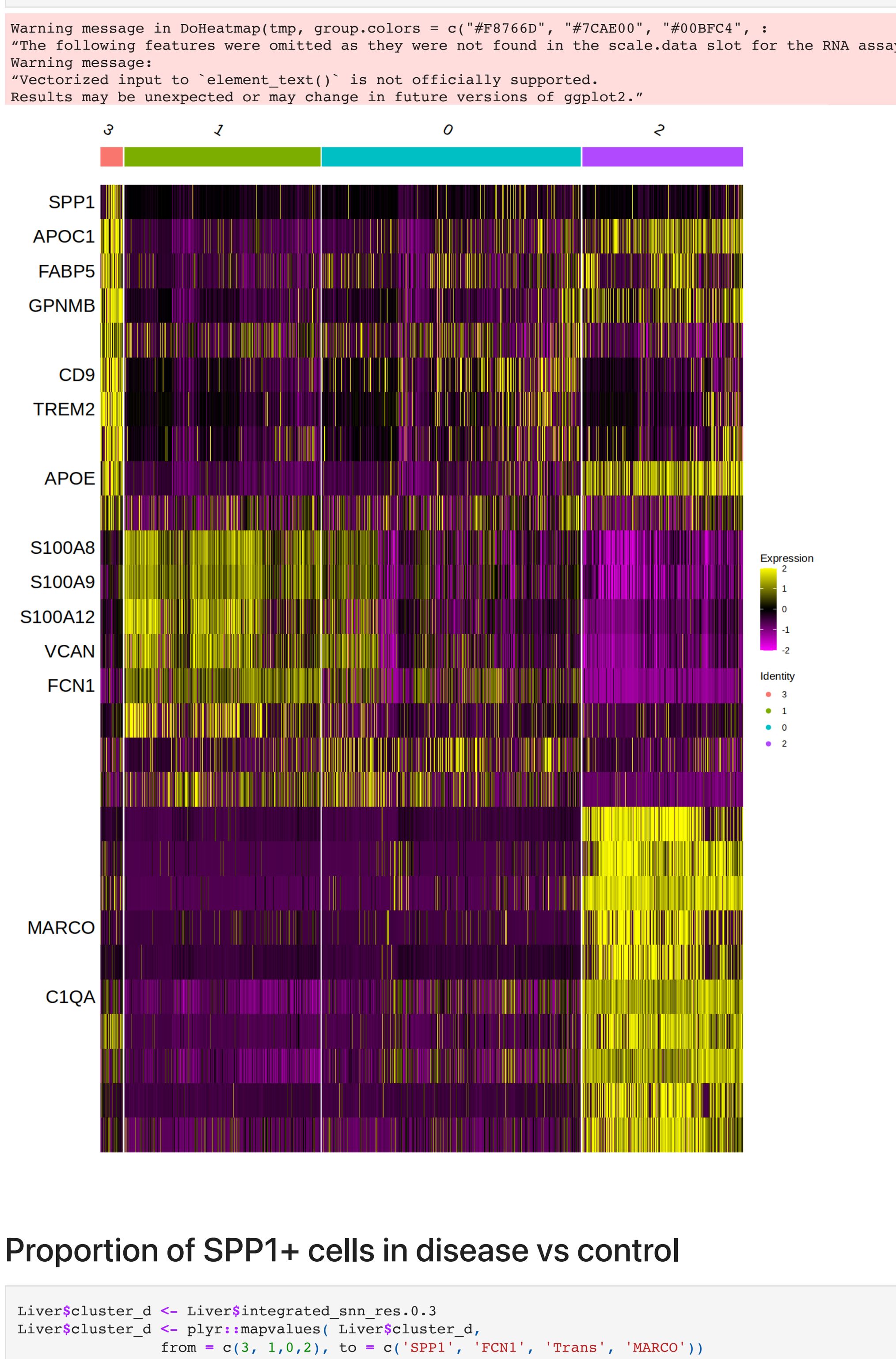
```
In [17]:
DefaultAssay(Liver) <- 'RNA'
Liver<- FindVariableFeatures(Liver, nfeatures = 2000 )
Endo_bulk <- AverageExpression(Liver, group.by = 'integrated_snn_res.0.3', assays = 'RNA', features = Liver$RNA@var.features )$RNA
Endo_bulk <- Endo_bulk / rowSums(Endo_bulk); Endo_bulk[ rowSums( is.na(Endo_bulk)  ) > 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)
```



Plot marker gene of each clsuter

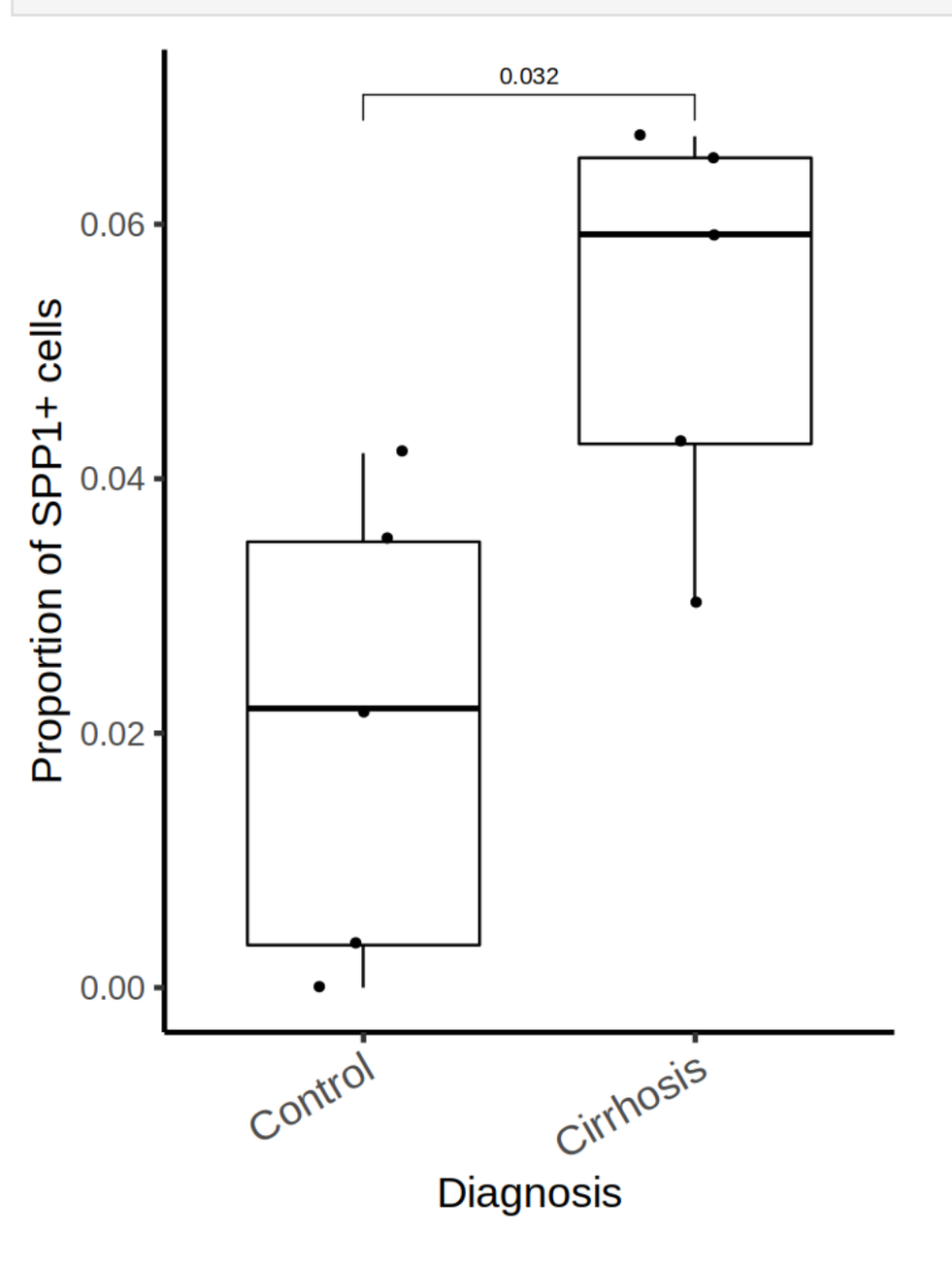
```
In [23]:
# Idents(Liver) <- Liver$integrated_snn_res.0.3
# DefaultAssay(Liver) <- 'RNA'
# Liver <- NormalizeData(Liver)
# Liver <- FindVariableFeatures(Liver)
# Liver <- ScaleData(Liver, vars.to.regress = c('mito.ratio', 'nFeature_RNA'))
# All_Heart.m <- FindAllMarkers(Liver, only.pos = T, min.diff.pct = .1)
# All_Heart.m <- All_Heart.m[ order( All_Heart.m$avg_log2FC, decreasing = T), ]

tmp <- Liver
DefaultAssay(tmp) <- 'RNA'
Idents(tmp) <- as.character(tmp$integrated_snn_res.0.3 )
tmp$try <- factor( tmp$integrated_snn_res.0.3 , levels = c(3,1, 0,  2)); res <- 'try'

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

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' ,'MARCO','C1QC','FCN1', 'S100A9','S100A8','VCAN', 'APOE','GPNMB','CD9','STAB1','SPP1', 'FABP5'
                    '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))
```

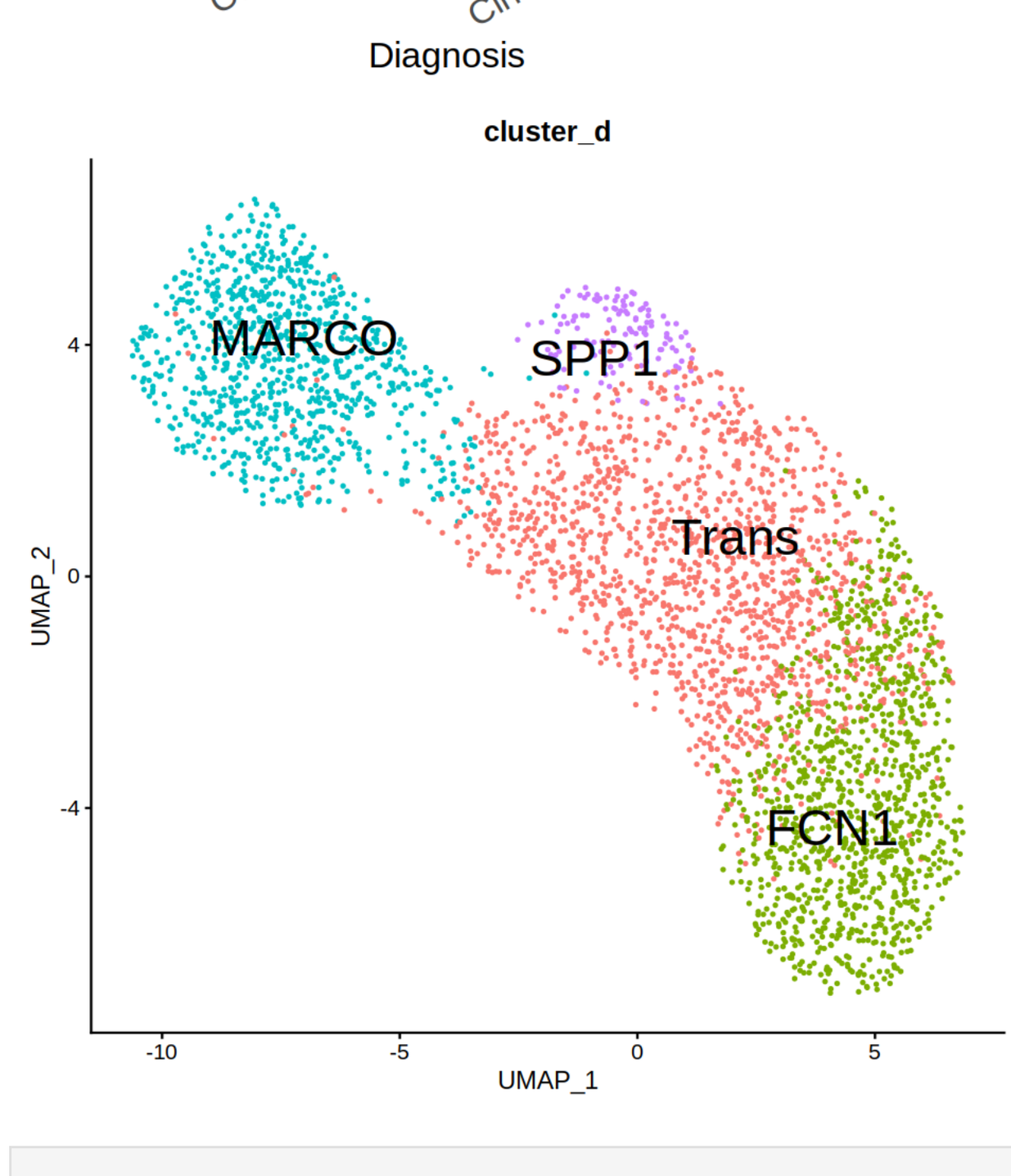
```
Warning message in DoHeatmap(tmp, group.colors = c("#F8766D", "#7CAE00", "#00BFC4", :
“The following features were omitted as they were not found in the scale.data slot for the RNA assay: NA, CSTA, CD55, S100A6, MNDA”
Warning message:
“Vectorized input to `element_text()` is not officially supported.
Results may be unexpected or may change in future versions of ggplot2.”
```



Proportion of SPP1+ cells in disease vs control

```
In [20]:
Liver$cluster_d <- Liver$integrated_snn_res.0.3
Liver$cluster_d <- plyr::mapvalues( Liver$cluster_d,
                from = c(3, 1,0,2), to = c('SPP1', 'FCN1', 'Trans', 'MARCO'))

Liver$Diagnosis <- 'Control'
Liver$Diagnosis[ Liver$orig.ident %in% c('GSM4041161_cirrhotic1_cd45+',
        'GSM4041164_cirrhotic2_cd45+', 'GSM4041166_cirrhotic3_cd45+', 'GSM4041168_cirrhotic4_cd45+',
                                        'GSM4041169_cirrhotic5_cd45+')] <- 'Cirrhosis'

options(repr.plot.height = 8, repr.plot.width = 6)
SPP1 <- (table(Liver$orig.ident, Liver$cluster_d) / rowSums(table(Liver$orig.ident, Liver$cluster_d)))[table(Liver$orig.ident) > 0, 'SPP1' ]
Diagnosis <-  plyr::mapvalues( names(table(Liver$orig.ident))[table(Liver$orig.ident) > 0],
    from = Liver$orig.ident[ !duplicated(Liver$orig.ident)], to = Liver$Diagnosis[ !duplicated(Liver$orig.ident)])
tmp <- data.frame('SPP1' = SPP1, 'Diagnosis' = Diagnosis)

ggboxplot(tmp, x = 'Diagnosis', y = 'SPP1',  add = "jitter") +
   stat_compare_means( method = 'wilcox.test', comparisons = list(c('Cirrhosis', 'Control'))) +
  theme_classic( base_size = 20) +
 theme(axis.text.x=element_text(angle = 30, hjust =1, vjust = 1, size = 20)) +
  ylab('Proportion of SPP1+ cells')


options(repr.plot.height = 8, repr.plot.width = 8)
DimPlot(Liver, group.by = 'cluster_d', label.size = 10, label = T, repel = T) + NoLegend()
```






```
In [26]:
saveRDS(Liver, file = 'Liver_code_final.RDS')
```