Exploratory Analysis

Ricardo Martins-Ferreira

2023-07-27

Representative pipeline of the differential expression analysis

```
libs <- c("Seurat", "tidyverse", "Nebulosa", "clusterProfiler",
          "dorothea","viper","OmnipathR", "ggplot2",
          "ggpubr","grid","gridExtra","org.Hs.eg.db",
          "data.table")
suppressMessages(
  suppressWarnings(sapply(libs, require, character.only = TRUE))
)
```

```
##          Seurat       tidyverse        Nebulosa clusterProfiler        dorothea
##            TRUE            TRUE            TRUE            TRUE            TRUE
##           viper       OmnipathR         ggplot2          ggpubr            grid
##            TRUE            TRUE            TRUE            TRUE            TRUE
##       gridExtra    org.Hs.eg.db      data.table
##            TRUE            TRUE            TRUE
```

Upload Seurat object

The Seurat object consists of the 64,438 nuclei already clustered and annotated for clinicopathological features of the subjects. This object is divided in 15 clusters.

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

Differentially expressed genes (DEGs)

DEGs for all clusters

```
# Find all markers (with normalized RNA assay)
# Find markers for every cluster compared to all remaining cells
DefaultAssay(Seurat)<- "RNA"
Seurat <- NormalizeData(Seurat)
Seurat <- ScaleData(Seurat)

# DEGs for each cluster vs all other
Seurat.markers <- FindAllMarkers(Seurat, only.pos = FALSE, min.pct = 0.25,
                                 logfc.threshold = 0.25, test.use = "MAST")
sig_markers <- Seurat.markers[Seurat.markers$p_val_adj<0.05,]

# DEGs only between Microglia clusters
Microglia.markers <- FindAllMarkers(subset(Seurat, idents = c(0,1, 2, 3,4,5,6,7,10,11,14),
                                           only.pos = FALSE, min.pct = 0.25,
                                           logfc.threshold = 0.25, test.use = "MAST")
```

```
Microglia_sig_markers <- Microglia.markers[Microglia.markers$p_val_adj<0.05,]

# DEGs between specific groups of nuclei
##(example for Hoemos1 (cluster 0) vs other homeostatic clusters (5, 10, 11))
Homeos.markers <- FindMarkers(Seurat,ident.1 = c("0"),
                              ident.2 = c("5","10","11"),
                              min.pct = 0.25,
                              logfc.threshold = 0.25, test.use = "MAST")

Homoes_sig_markers <- Homoes_sig_markers[Homoes_sig_markers$p_val_adj<0.05,]

# DEGs calculation to use in Transcription Factor motif enrichment

Seurat.markers.TF <- FindAllMarkers(Seurat, only.pos = FALSE, min.pct = 0.10,
                                    logfc.threshold = 0, test.use = "MAST")
```

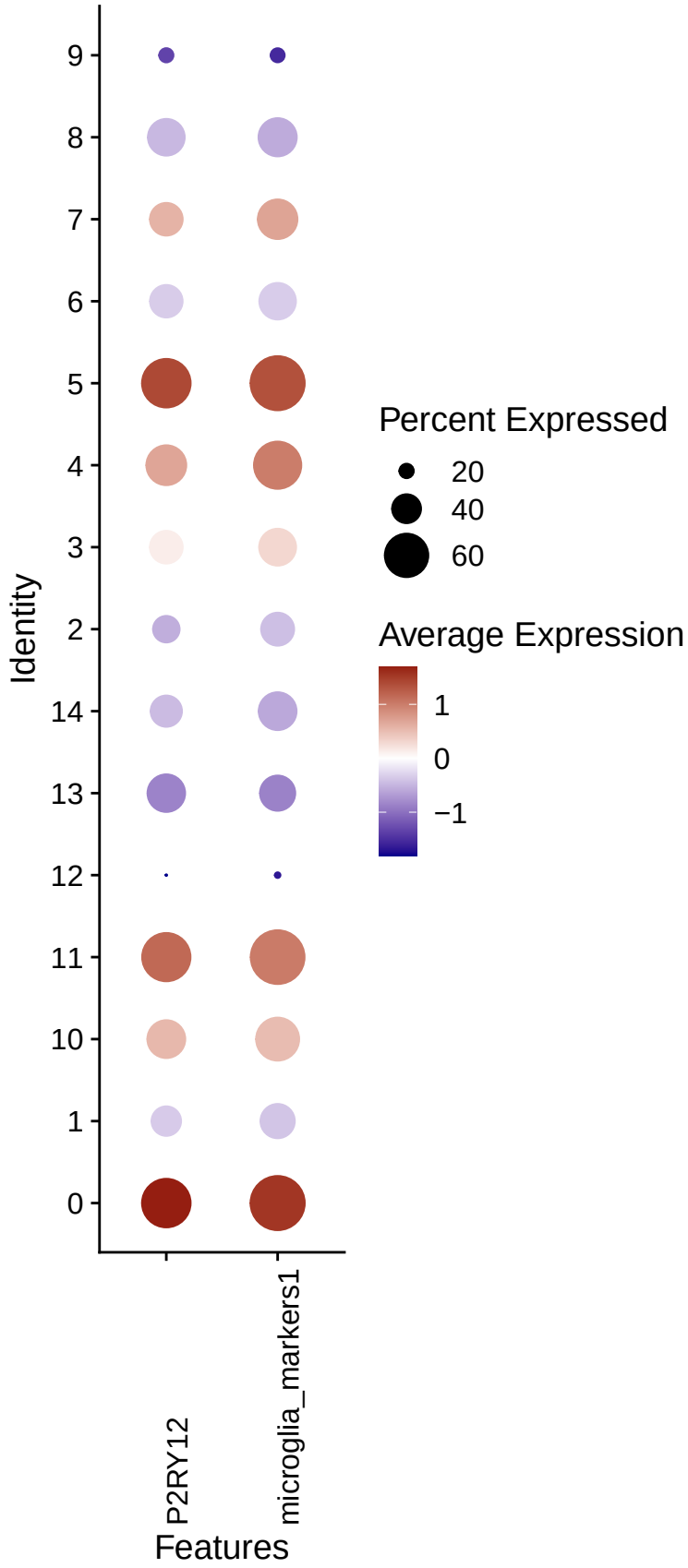
Module Score expression of groups of genes

```
microglia_markers <- c("CX3CR1","P2RY12")

Seurat <- AddModuleScore(Seurat,
                         features =list(microglia_markers),
                         name='microglia_markers')

# check module scores created
names(x = Seurat[[]])
```

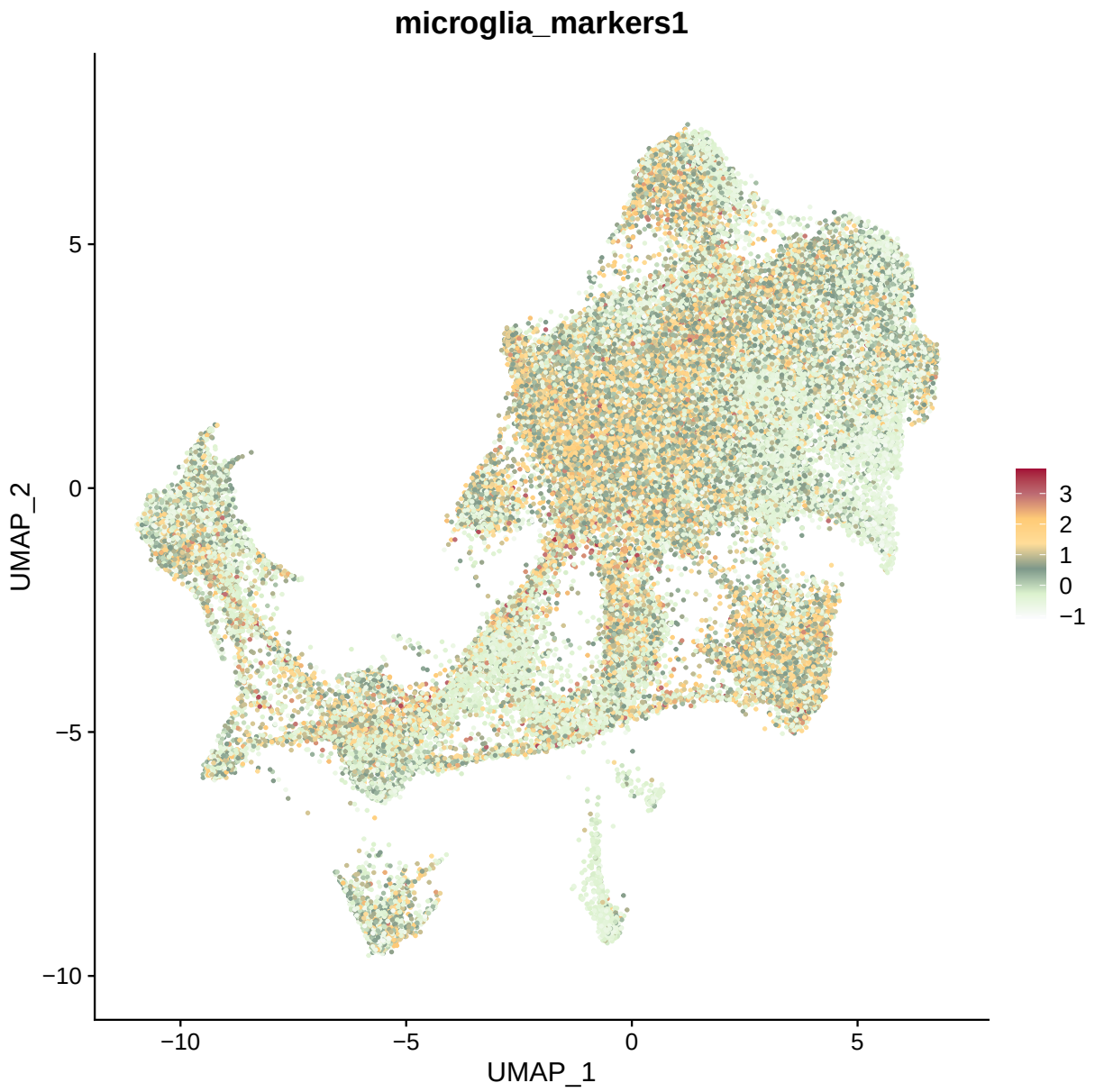
```
##  [1] "orig.ident"              "nCount_RNA"
##  [3] "nFeature_RNA"            "percent.ribo"
##  [5] "projid"                  "Subject"
##  [7] "nCount_SCT"              "nFeature_SCT"
##  [9] "SCT_snn_res.0.05"        "seurat_clusters"
## [11] "Mg_genes1"               "SCT_snn_res.0.08"
## [13] "percent.mt"              "TAG"
## [15] "integrated_snn_res.0.25" "Technical replicate_ID"
## [17] "Study"                   "Group"
## [19] "Gender"                  "Age"
## [21] "Tissue_type"             "Tissue_condition"
## [23] "Specific diagnosis"      "Cause of death"
## [25] "Age_conc"                "microglia_markers1"
```

Visualization of gene and module score expression

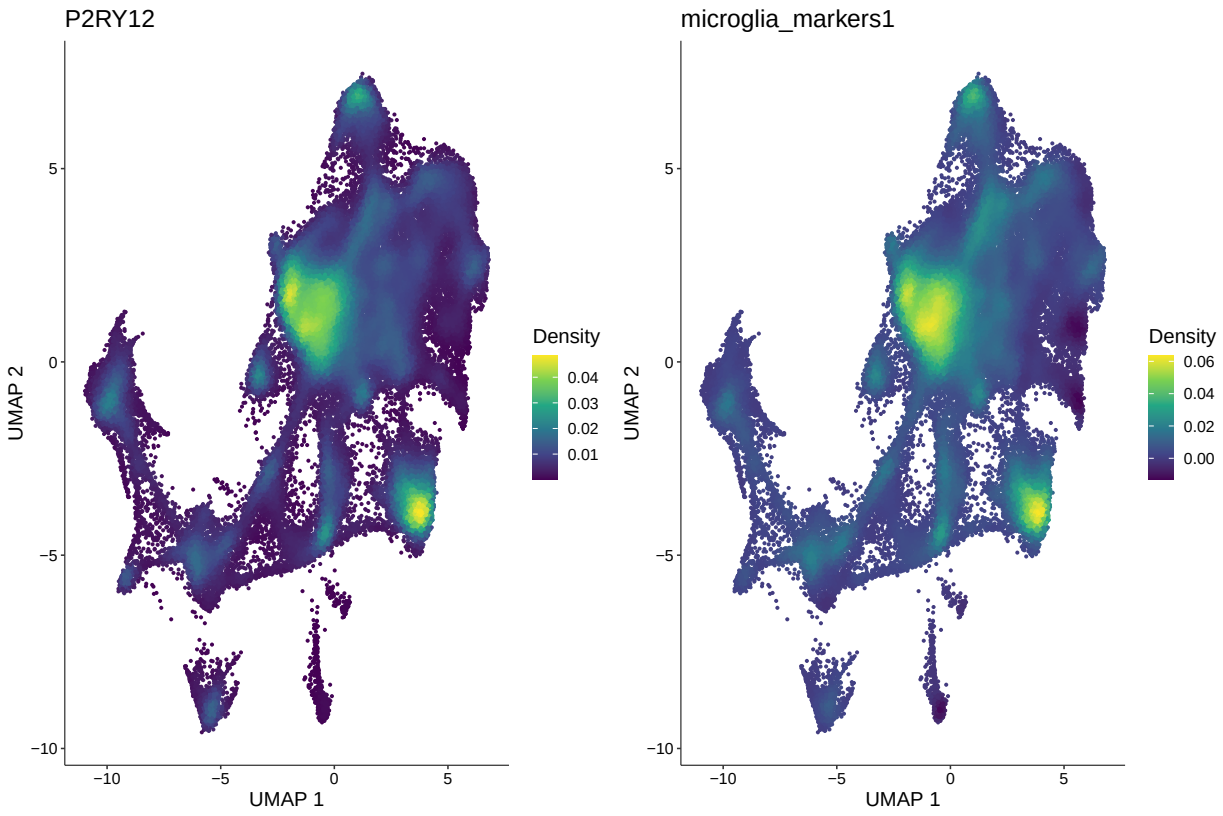
```
# DotPlot
DotPlot(Seurat, features = c("P2RY12", "microglia_markers1"),
  dot.scale = 10, group.by ="integrated_snn_res.0.25") +
  scale_colour_gradient2(low = "darkblue", mid = "white", high = "darkred")+
  theme(axis.text.x = element_text(angle=90, hjust = 0))
```


Identity
9
8
7
6
5
4
3
2
14
13
12
11
10
1
0
P2RY12
microglia_markers1
Features
Percent Expressed
20
40
60
Average Expression
1
0
−1

```r
# FeaturePlot
FeaturePlot(Seurat, features = 'microglia_markers1', label = F, repel = TRUE,
            pt.size = 0.5) &
  scale_colour_gradientn(colours = c("#FCFCFF","#DCF2CE","#7E998A",
                                     "#FFDD99","#FFCB77","#BD6B73","#A30B37"))
```




```r
# Nebulosa plot
plot_density(Seurat, c("P2RY12", "microglia_markers1"))
```



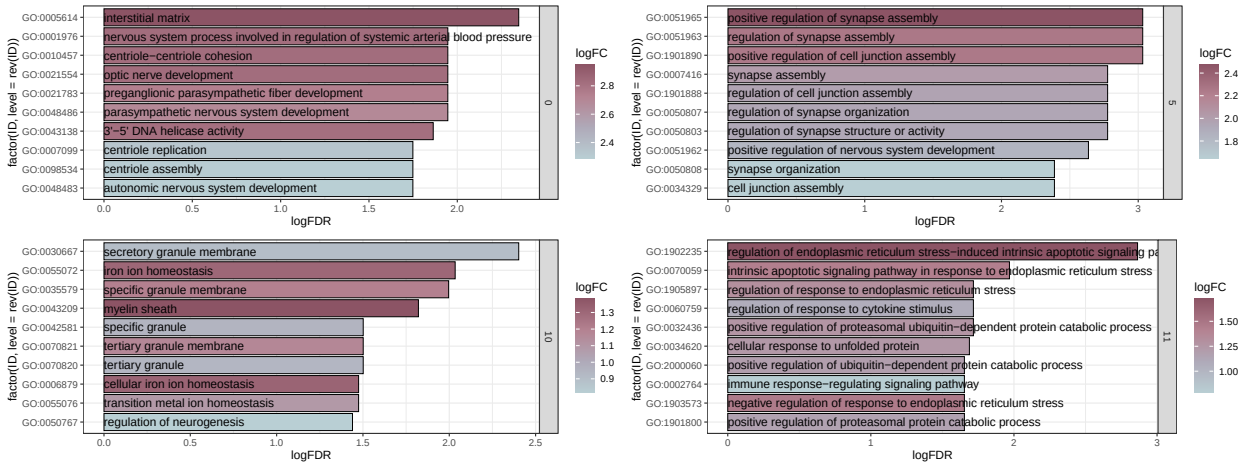
Gene Ontology with enrichGO

```
# example using sig_genes, DEGs of each cluster vs all others
## include only upregulated DEGs
up_genes <- sig_genes[sig_genes$avg_log2FC>0,]

# Create a loop for each cluster
i = 0
results <- list()
go_all <- data.frame()
for(comparison in levels(factor(up_genes$cluster))){
  i = i+1
  print(paste(comparison))
  ego <- enrichGO(gene              = up_genes[up_genes$cluster == comparison,]$gene,
                  OrgDb             = org.Hs.eg.db,
                  ont               = "ALL",
                  keyType           = "SYMBOL",
                  pAdjustMethod     = "BH",
                  pvalueCutoff      = 1,
                  qvalueCutoff      = 0.05,
                  readable          = T )


    results[[comparison]] <- ego@result %>%
      separate(GeneRatio, into = c("gene_pos", "gene_total"), sep = "/") %>%
      separate(BgRatio, into = c("bg_pos", "bg_total"), sep = "/") %>%
```

```
      mutate(FC = (as.numeric(gene_pos)/as.numeric(gene_total)) /
               (as.numeric(bg_pos)/as.numeric(bg_total)),
             cluster = comparison) %>%
      arrange(.$p.adjust)
    go_all <- rbind(go_all, results[[comparison]])
  }
```

Plot GO output The plotting is exemplified with the results obtained for the results represented in Supplementary Figure 6B. The top 10 most significant GO terms to the comparison of differential expression specifically in each of the homeostatic subpopulations.

```
# Plot top 10 most significant GO terms for each cluster
clusterlist <- c("0","5","10","11")
results <- go_all
p<-list()

for (i in 1:length(clusterlist)) {

  df <- results[results$cluster %in% clusterlist[i],]
  data <- df[1:10,]

  data$logFDR<- -log10(data$p.adjust)
  data$logFC<- log10(data$FC)



  p[[i]]<-  ggplot(data,
                   aes(fill=logFC,
                       x=factor(ID, level = rev(ID)),
                       y=logFDR,
                       color="black")) +
    geom_col(color="black", size=0.2) +
    geom_text(aes(label = Description), hjust=0,position = position_fill(vjust = 0),
              size = 4, color="black")+
    theme_bw() +
    facet_grid(cluster~., shrink = TRUE, scales = "free", space="free")+
    scale_fill_gradientn(colours = c("#B9CFD4","#AFAAB9","#B48291","#8C5465"))+
    #scale_fill_viridis(option = "plasma")+
    coord_flip()

}

do.call(grid.arrange,p)
```



Transcription factor (TF)

```
# upload A_viperRegulon.rdata
viper_regulon -> regulon_A

#Clean TF names
names(regulon_A)=sapply(strsplit(names(regulon_A),split=" - "),head, 1)

# upload differential expression output

data <- Seurat.markers.TF

# Exclude probes with unknown or duplicated gene symbol
DEsignature = subset(data, gene != "" )

DEsignature = subset(DEsignature, ! duplicated(gene))

# Estimatez-score values for the GES. Cheeck VIPER manual for details

myStatistics = matrix(DEsignature$avg_log2FC, dimnames = list(DEsignature$gene,

                                                              'avg_log2FC') )

myPvalue = matrix(DEsignature$p_val_adj, dimnames = list(DEsignature$gene, 'padj') )

mySignature = (qnorm(myPvalue/2, lower.tail = FALSE) * sign(myStatistics))[, 1]

mySignature = mySignature[order(mySignature, decreasing = T)]

# Estimate TF activities

mrs = msviper(ges = mySignature, regulon = regulon_A, minsize = 4, ges.filter = F)

TF_activities = data.frame(Regulon = names(mrs$es$nes),

                           Size = mrs$es$size[ names(mrs$es$nes) ],

                           NES = mrs$es$nes,
```

```
                          p.value = mrs$es$p.value,

                          FDR = p.adjust(mrs$es$p.value, method = 'fdr'))

TF_activities = TF_activities[ order(TF_activities$p.value), ]
```

Plot TF enrichment The plotting is exemplified with the results obtained for the results represented in Supplementary Figure 6C. TF enrichment of all significant TFs (p value < 0.01) to the comparison of differential expression of all homeostatic clusters vs all other microglia, and specifically in each of the homeostatic subpopulations.

```
# upload DEG output for TF enrichment

# All homoestatic vs other microglia
Homeos_ALL_TF$cluster <- "ALL"
sig_Homeos_ALL_TF <- Homeos_ALL_TF[Homeos_ALL_TF$p.value<0.01,]

# Cluster 0 vs clusters 5, 10, 11
sig_Homeos_0_TF <- Homeos_0_TF[Homeos_0_TF$p.value<0.01,]
Homeos_0_TF$cluster <- "0"

# Cluster 5 vs clusters 0, 10, 11
sig_Homeos_5_TF <- Homeos_5_TF[Homeos_5_TF$p.value<0.01,]
Homeos_5_TF$cluster <- "5"

# Cluster 10 vs clusters 0, 5, 11
sig_Homeos_10_TF <- Homeos_10_TF[Homeos_10_TF$p.value<0.01,]
Homeos_10_TF$cluster <- "10"

# Cluster 11 vs clusters 0, 5, 10
sig_Homeos_11_TF <- Homeos_11_TF[Homeos_11_TF$p.value<0.01,]
Homeos_11_TF$cluster <- "11"


# Include significant TF only (p.value < 0.01)


tfs.to.plot <- unique(c(sig_Homeos_ALL_TF$Regulon,
                        sig_Homeos_0_TF$Regulon,
                        sig_Homeos_5_TF$Regulon,
                        sig_Homeos_10_TF$Regulon,
                        sig_Homeos_11_TF$Regulon))


common_features <- tfs.to.plot
tf_all <- rbind(Homeos_ALL_TF,Homeos_0_TF,Homeos_5_TF,Homeos_10_TF,Homeos_11_TF)

tf_all2 <- tf_all %>%
  mutate(logFDR = -log(FDR),
         cluster = as.character(cluster)) %>%
  mutate(cluster = factor(cluster, levels = c("ALL","0","5","10","11")))

ggplot(tf_all2 %>% dplyr::filter(Regulon %in% common_features),
```

```
      aes(cluster, y= factor(Regulon),color=NES,size =logFDR)) +
geom_point() +
#scale_size_continuous(range=c(3,9))+
scale_color_gradient2(low="#47663D",mid = "white",high="#B86B00")+
theme_pubr(border = 1)
```