

```
# BiocManager::install("devtools")
#library(devtools)
# devtools::install_local("Seurat_4.4.0.tar.gz")
library(Seurat)
set.seed(805181) ## for repeatability of cluster mapping.

## install escape for function enrichment analysis
# BiocManager::install("escape")
library(escape)
# remove.packages("escape")

# BiocManager::install("scuttle")
# library(scuttle)
# devtools::install_github("davismcc/scater")
library(scater)

library(SingleCellExperiment)
library(scran)
# BiocManager::install("scran")
```

1- prepare the meta.data

```
## input is seurat
load("R8_2_seurat_merged_annotated.Rdata")
seurat_merged-> selected_obj
```

```

## meta.data
metadata <- selected_obj@meta.data

metadata[,c(1:4,7,18,21,22)]-> selected_metadata

# export
library(openxlsx)
write.xlsx(selected_metadata,
           rowNames=T,
           "meta_data_info.xlsx")

```

2- Prepare the UMAP coordinates

```

selected_obj@reductions$umap@cell.embeddings-> umap_cor

write.xlsx(umap_cor,
           rowNames=T,
           "UMAP_coordinates.xlsx")

```

3- Stress check [UCell]

3.1- GSEA-stress genes

- UCell Scores are normalized to a range (usually between 0 and 1) based on the ranked enrichment, making the interpretation relative across cells.
- Scores near zero are robust indicators that the gene set is not significantly active in those cells.

```

##
library(Seurat)
#BiocManager::install("UCell")
library(UCell) ## works only on R4.3, can't install it on R4.1
library(ggplot2)

```

```

library(RColorBrewer)

#>> input stress gene list
library(readxl)
gene_table <- read_excel("RESPONSE_TO_STRESS_genes.xlsx", sheet =
"genes")

#>> stress gene list
markers <- list()
markers$Stress_response <- unique(gene_table$gene_symbol) ## 508
stress genes from GSEA.

#>> calculate the gene_set score
DefaultAssay(selected_obj) <- "RNA"
selected_obj <- AddModuleScore_UCell(selected_obj,
                                       features = markers) ## will
add one more column in meta.data for score
##
ACVRL1,ADORA3,AIF1,ALB,APCS,APOA4,APOA5,C4BPB,CARTPT,CCL11,CCL13,CC
L21,CCL23,CCL3,CCL3L3,CCL4,CCR1,CCR2,CCR5,CD40LG,CHRNA4,CLDN3,CRNN,
CRP,CXCR4,F7,F9,FPR2,GHSR,GP9,HMGB1P10,HOXB13,HSPB3,HSPB7,IFNA2,IL1
7C,CXCR1,IL9,KRT1,MBL2,ORM1,ORM2,PF4,PLA2G2D,PMS2P1,S100A7,TNP1,XCR
1

# plot the stress score >>>>>>>>>>>>>>>>>>>>>
library(colorspace)
library(scCustomize)
library(Seurat)
library(viridis)
library(RColorBrewer)

## plot_title
signature.names <- paste0(names(markers), "_UCell")
plot_color = rev(brewer.pal(11, "RdYlBu"))

p1<- FeaturePlot(selected_obj,

```

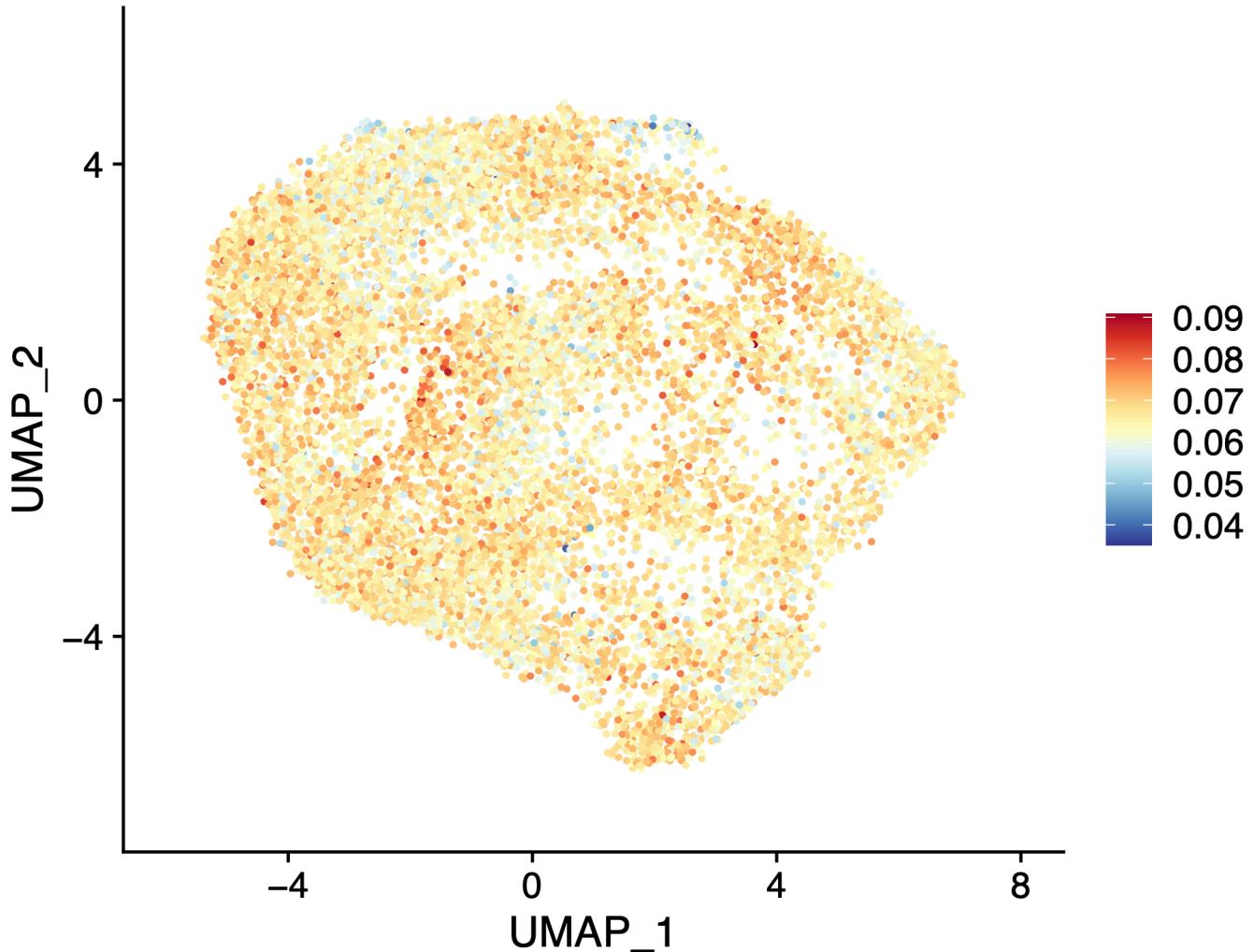
```

        features= signature.names,
        ncol = 1,
        pt.size=0.5) +
scale_color_gradientn(colors = plot_color) +
ggtitle("RESPONSE_TO_STRESS_gsea_UCell")

ggsave("review_Stress_geneset_1_GSEA.pdf",
plot = p1,
width = 6,
height =5 )

```

RESPONSE_TO_STRESS_gsea_UCell



```
ca_obj <- subset(selected_obj, subset = orig.ident == "Ca")
```

```

p1<- FeaturePlot(ca_obj,
                  features= signature.names,
                  pt.size=0.5) +
  scale_color_gradientn(colors = plot_color) +
  ggtitle("Ca_STRESS_gsea_UCell")

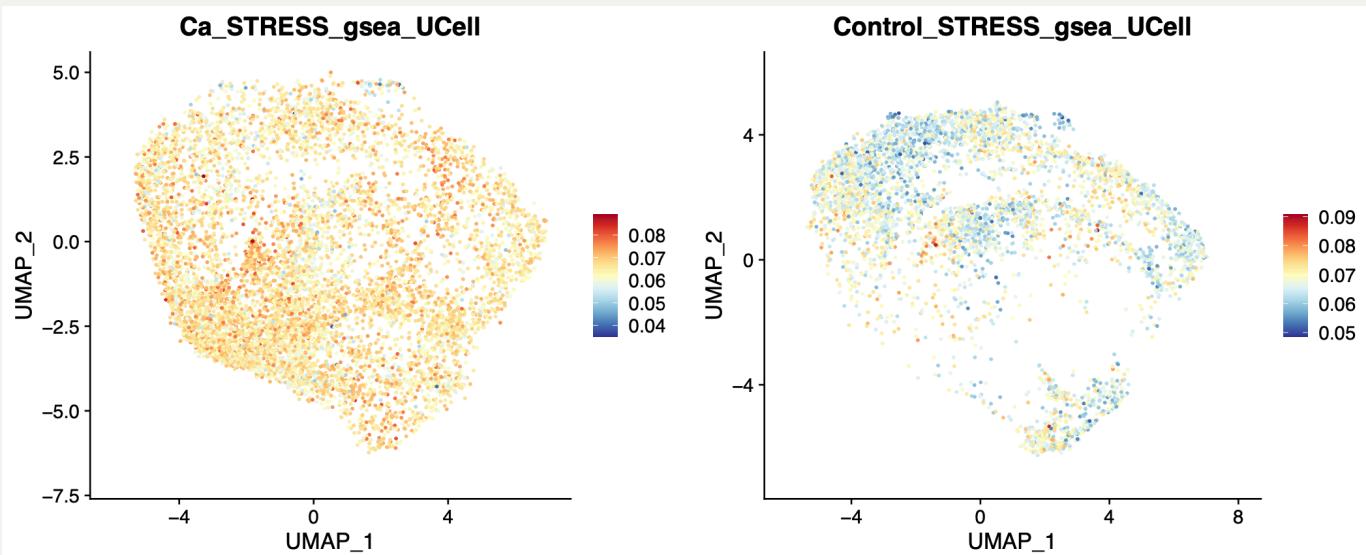
Control_obj <- subset(selected_obj, subset = orig.ident ==
  "Control")
p2<- FeaturePlot(Control_obj,
                  features= signature.names,
                  pt.size=0.5) +
  scale_color_gradientn(colors = plot_color) +
  ggtitle("Control_STRESS_gsea_UCell")

library(ggpubr)

p12<- ggarrange(p1, p2,
                 ncol = 2)

ggsave("review_Stress_geneset_1_GSEA_split_plot.pdf",
       plot = p12,
       width = 12,
       height =5 )

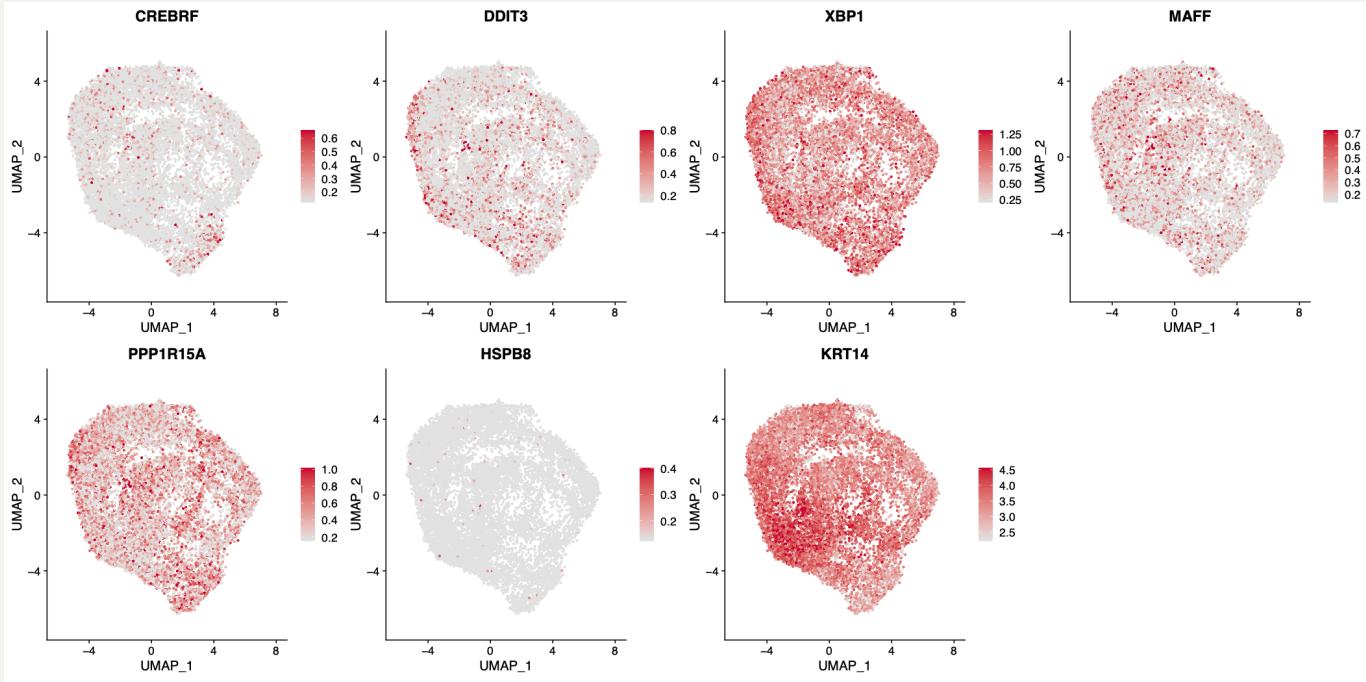
```



3.1- Selected stress genes from paper

- https://static-content.springer.com/esm/art%3A10.1038%2Fs41467-021-22779-9/MediaObjects/41467_2021_22779_MOESM1_ESM.pdf
- <https://pubmed.ncbi.nlm.nih.gov/33947848/>
- CREBRF, DDIT3, XBP1, MAFF, PPP1R15A, HSPB8

```
known_marker=c(  
    # the stress marker used in 2d culture paper  
    "CREBRF", "DDIT3", "XBP1", "MAFF", "PPP1R15A",  
    "HSPB8", "KRT14")  
  
## define the default assay  
DefaultAssay(selected_obj)<- "RNA"  
  
## for original obj  
p1<- FeaturePlot(selected_obj,  
                    pt.size = 0.5,  
                    reduction= "umap",  
                    features= unique(known_marker),  
                    ncol = 4,  
                    cols = c('grey90','#CC0033'),  
                    min.cutoff = 'q2',max.cutoff = 'q98')  
  
## cols = c('grey90','#CC0033'),  
  
ggsave("Review_Stress_genes_2_plot.pdf",  
       plot = p1, width = 20, height =10)
```



4- Geneset Enrichment plot

- To investigate whether the genes associated with a given trait exhibit enrichment, we use gene sets from human MSigDB collections <<https://www.gsea-msigdb.org/gsea/msigdb/genesets.jsp>>[], which includes four assays: "GO_CC", "GO_BP", "GO_MF," and "Hallmark".

```
load("R11_4_geneset_enrich_normalized_seurat.Rdata")
# selected_obj
```

CC Plot for "Chromatin organization" and "Lysosome acidity"

```
DefaultAssay(selected_obj)<- "go_CC_ssgSEA_normalized"
selected_genesets<- c("GOCC-CHROMATIN-SILENCING-COMPLEX",
                      "GOCC-CHROMATIN",
                      "GOCC-CYTOPLASMIC-SIDE-OF-LYSOSOMAL-
MEMBRANE",
                      "GOCC-LYSOSMAL-LUMEN")
```

```
library(RColorBrewer)
```

```

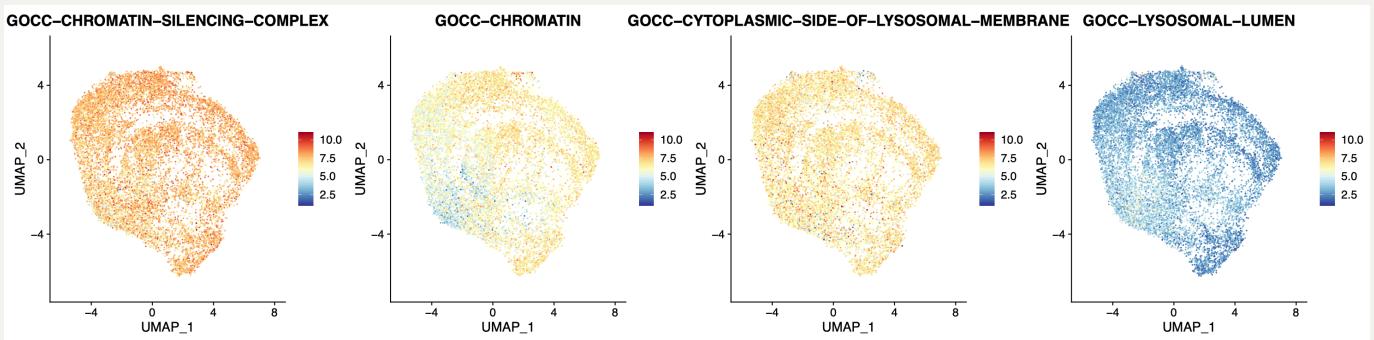
plot_color = rev(brewer.pal(11, "RdYlBu"))

p1<-FeaturePlot(selected_obj,
                  features = selected_genesets,
                  cols = plot_color,
                  # pt.size = 2,
                  ncol = 4)

# scale_color_gradientn(colors = plot_color)
# +theme(plot.title = element_text(color="red",
size=10,face="bold.italic"))

ggsave("review_Extra_GOCC_geneset_plot.pdf",
       plot = p1, width = 20, height = 5)

```



```

# define cluster color
mycolor= c(
  "H1"= "#99CC00CC",
  "H2"= "#33CC00CC",
  "H3"= "#5DB1DDCC",
  "H4"= "#00CC99CC",
  "M1"= "#CC9900CC",
  "M2"= "#C75127CC",
  "P1"= "#D595A7CC",
  "P2"= "#802268CC",
  "P3"= "#7A65A5CC",
  "P4"= "#AE1F63CC",
  "P5"= "#612A79CC")

```

```
DefaultAssay(selected_obj)<-"go_CC_ssgSEA_normalized"

p1<- VlnPlot(selected_obj,
               features = "GOCC-LYSOSOMAL-LUMEN",
               group.by = "cell_states",
               pt.size = 0,
               cols= mycolor)+

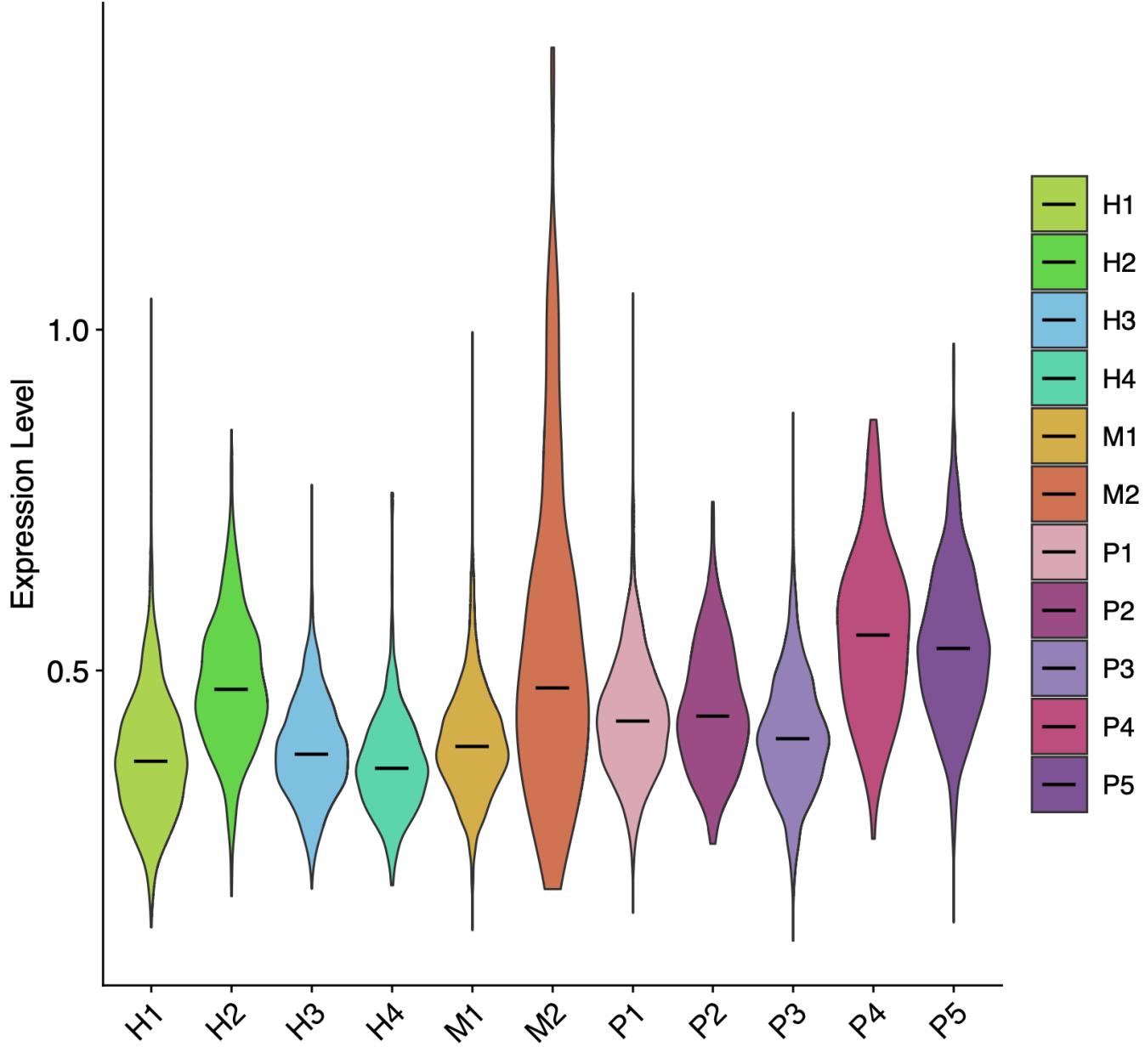
theme(axis.text = element_text(size = 15)) +
theme(text = element_text(size = 15))+

theme(legend.position = "right") + xlab("") + 

stat_summary(fun.y = median, geom='point',size= 10, colour =
"black", shape = 95)

ggsave("review_4_Vlnplot_GOCC-LYSOSOMAL-LUMEN.pdf",
       plot = p1, width = 8, height =8 )
```

GOCC–LYSOSOMAL–LUMEN



MF Plot for "Chromatin organization" and "Lysosome acidity"

```

## plot_MF
DefaultAssay(selected_obj) <- "go_MF_ssgSEA_normalized"

selected_genesets <- c("GOMF-ATP-DEPENDENT-CHROMATIN-REMODELER-
ACTIVITY",
                      "GOMF-CHROMATIN-BINDING",
                      "GOMF-CHROMATIN-DNA-BINDING",

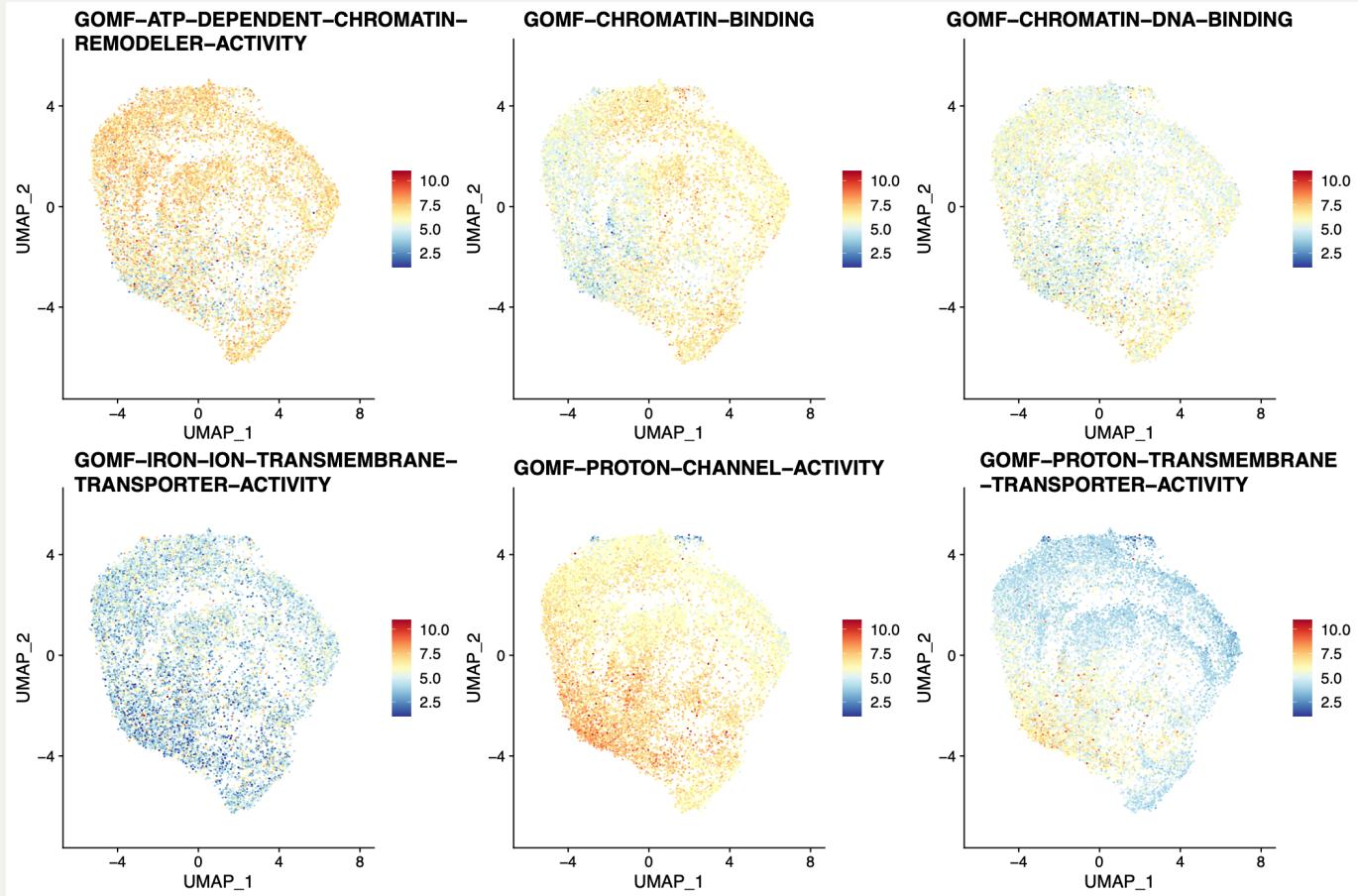
```

```
"GOMF-IRON-ION-TRANSMEMBRANE-TRANSPORTER-
ACTIVITY",
"GOMF-PROTON-CHANNEL-ACTIVITY",
"GOMF-PROTON-TRANSMEMBRANE-TRANSPORTER-
ACTIVITY")

library(RColorBrewer)
plot_color = rev(brewer.pal(11, "RdYlBu"))

p1<-FeaturePlot(selected_obj,
                  features = selected_genesets,
                  cols = plot_color,
                  # pt.size = 2,
                  ncol = 3)
# scale_color_gradientn(colors = plot_color)
# +theme(plot.title = element_text(color="red",
# size=10,face="bold.italic"))

ggsave("review_Extra_GOMF_geneset_plot.pdf",
       plot = p1, width = 15, height = 10)
```



BP Plot for "Chromatin organization" and "Lysosome acidity"

```

## plot_BP
DefaultAssay(selected_obj)<- "go_BP_ssgSEA_normalized"

selected_lysosome<-c("GOBP-LYSOSMAL-LUMEN-ACIDIFICATION",
                      "GOBP-LYSOSMAL-MEMBRANE-ORGANIZATION",
                      "GOBP-REGULATION-OF-LYSOSMAL-LUMEN-PH",
                      "GOBP-CELLULAR-RESPONSE-TO-IRON-ION",
                      "GOBP-IRON-ION-TRANSMEMBRANE-TRANSPORT",
                      "GOBP-IRON-ION-TRANSPORT",
                      "GOBP-REGULATION-OF-IRON-ION-TRANSMEMBRANE-
TRANSPORT",
                      "GOBP-RESPONSE-TO-IRON-ION" )

library(RColorBrewer)
plot_color = rev(brewer.pal(11, "RdYlBu"))

```

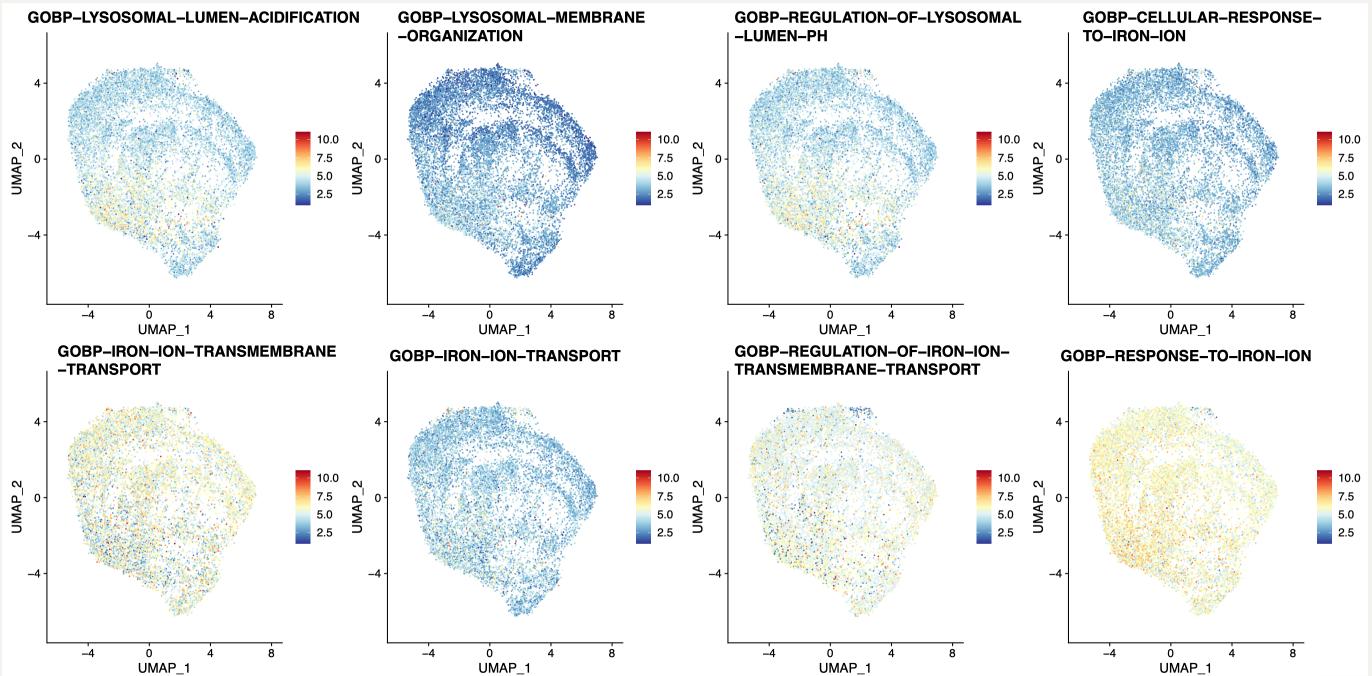
```

p1<-FeaturePlot(selected_obj,
                  features = selected_lysosome,
                  cols = plot_color,
                  # pt.size = 2,
                  ncol = 4)

# scale_color_gradientn(colors = plot_color)
# +theme(plot.title = element_text(color="red",
size=10,face="bold.italic"))

ggsave("review_Extra_GOBP_Lysosome_geneset_plot.pdf",
       plot = p1, width = 20, height = 10)

```



```

selected_chromatin<- c("GOBP-CHROMATIN-LOOPING", # no found
                      "GOBP-CHROMATIN-REMODELING",
                      "GOBP-CONSTITUTIVE-HETEROCHROMATIN-
FORMATION", # no found
                      "GOBP-DNA-METHYLATION-DEPENDENT-
HETEROCHROMATIN-FORMATION", # not found
                      "GOBP-DNA-REPAIR-DEPENDENT-CHROMATIN-
REMODELING", # no founf

```

```

"GOBP-FACULTATIVE-HETEROCHROMATIN-FORMATION",
# no found

"GOBP-HETEROCHROMATIN-FORMATION", # no found
"GOBP-HETEROCHROMATIN-ORGANIZATION",
"GOBP-NEGATIVE-REGULATION-OF-CHROMATIN-
ORGANIZATION",
"GOBP-NUCLEOLAR-CHROMATIN-ORGANIZATION",
"GOBP-POSITIVE-REGULATION-OF-CHROMATIN-
ORGANIZATION",
"GOBP-POSITIVE-REGULATION-OF-DNA-METHYLATION-
DEPENDENT-HETEROCHROMATIN-FORMATION", # no found
"GOBP-REGULATION-OF-CHROMATIN-ORGANIZATION",
"GOBP-REGULATION-OF-DNA-METHYLATION-
DEPENDENT-HETEROCHROMATIN-FORMATION", # no found
"GOBP-TRANSCRIPTION-ELONGATION-COUPLED-
CHROMATIN-REMODELING") # no found

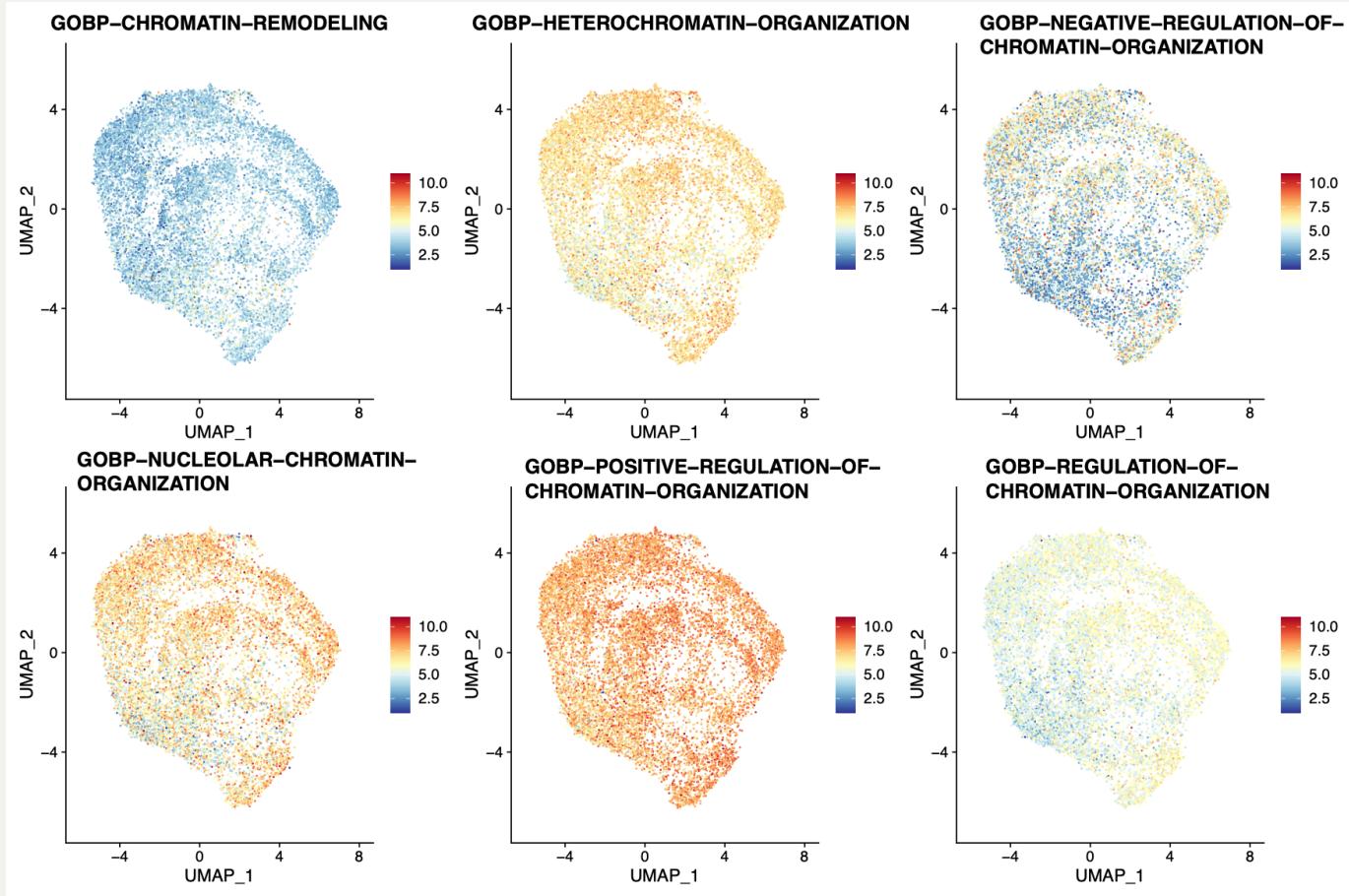
library(RColorBrewer)
plot_color = rev(brewer.pal(11, "RdYlBu"))

p1<-FeaturePlot(selected_obj,
                  features = selected_chromatin,
                  cols = plot_color,
                  # pt.size = 2,
                  ncol = 3)

# scale_color_gradientn(colors = plot_color)
# +theme(plot.title = element_text(color="red",
# size=10,face="bold.italic"))

ggsave("review_Extra_GOBP_Chromatin_geneset_plot.pdf",
       plot = p1, width = 15, height = 10)

```



Genes for "Chromatin organization"

```

selected_genes<-c("H2AFZ", "CENPA", # Histones and Histone Variants

"SMARCA4", "SMARCA2", "ARID1A", "ARID1B", "SMARCA1", "SMARCA5", "CHD3",
"CHD4", # Chromatin Remodelers

"EP300", "CREBBP", "KAT2A", "HDAC1",
"HDAC2", "SIRT1", "EZH2", "SUV39H1", "KDM1A", "KDM6A", # Histone
Modifiers

"DNMT1", "DNMT3A", "DNMT3B", "TET1", "TET2", "TET3", # DNA
Modifiers

"CTCF", "RAD21", "SMC1A", "SMC3", "STAG1", "STAG2", # Chromatin
Structural Proteins

"EZH2", "EED", "SUZ12", "CBX2", "RING1", "KMT2A", "KMT2B", # Polycomb
and Trithorax Complexes

"LMNB1", "ATRX", "BRD4") # Additional Factors

```

```
length(selected_genes) # 42

## define the default assay
DefaultAssay(selected_obj)<- "RNA"

## for original obj
p1<- FeaturePlot(selected_obj,
                    pt.size = 0.5,
                    reduction= "umap",
                    features= unique(selected_genes),
                    ncol = 7,
                    cols = c('grey90','#CC0033')) 

ggsave("Review_selected_chromatin_genes_plot.pdf",
       plot = p1, width = 40, height =35)
```



5- Prepare the cell expression values for analysis

- HALLMARK-OXIDATIVE-PHOSPHORYLATION
- HALLMARK-REACTIVE-OXYGEN-SPECIES-PATHWAY
- HALLMARK-FATTY-ACID-METABOLISM
- HALLMARK-ADIPOGENESIS
- GOBP-REGULATION-OF-CHROMATIN-ORGANIZATION
- GOBP-NEGATIVE-REGULATION-OF-CHROMATIN-ORGANIZATION
- GOBP-REGULATION-OF-LYSOSOMAL-LUMEN-PH

- GOBP-LYSOSOMAL-LUMEN-ACIDIFICATION
- Expression values for the genes in the stress genes set 2 (also keratin 14):
 - CREBRF
 - DDIR3
 - XBP1
 - MAFF
 - PPP1R15A
 - HSPB8
 - KRT14

```
load("R11_4_genset_enrich_normalizaed_seurat.Rdata")
# selected_obj
```

5.1 selected_Hallmark matrix

```
## plot_hallmark
DefaultAssay(selected_obj)<- "hallmark_ssGSEA_normalized"

selected_hallmark<-c("HALLMARK-OXIDATIVE-PHOSPHORYLATION",
" HALLMARK-REACTIVE-OXYGEN-SPECIES-PATHWAY",
" HALLMARK-FATTY-ACID-METABOLISM",
" HALLMARK-ADIPOGENESIS")

library(RColorBrewer)
plot_color = rev(brewer.pal(11, "RdYlBu"))

p1<-FeaturePlot(selected_obj,
                  features = selected_hallmark,
                  cols = plot_color,
                  # pt.size = 2,
                  ncol = 4)

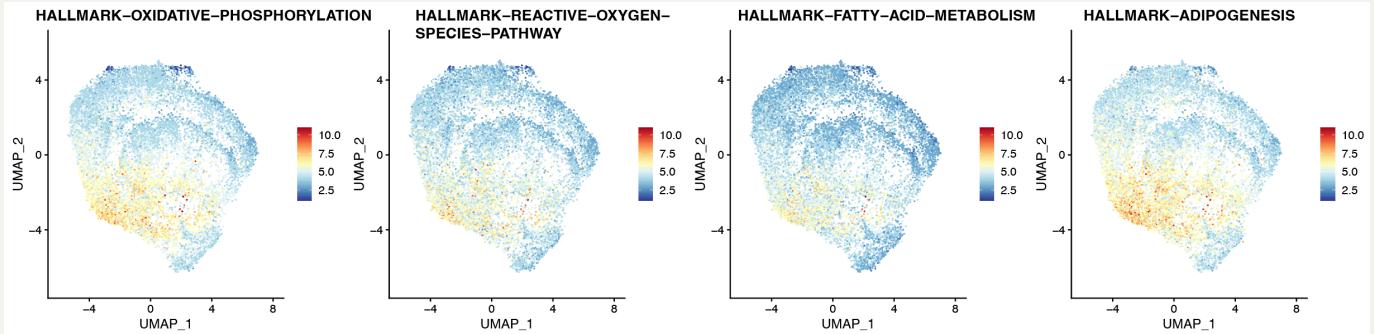
# scale_color_gradientn(colors = plot_color)
```

```

# +theme(plot.title = element_text(color="red",
size=10,face="bold.italic"))

ggsave("review_final_hallmark_plot.pdf",
plot = pl, width = 20, height = 5)

```



```

library(Matrix)

hallmark_matrix=as.data.frame(selected_obj@assays$hallmark_ssGSEA_n
ormalized@data)

## transpose the row and column
transposed_hallmark <- as.data.frame(t(hallmark_matrix))

## select the intested columns.
final_hallmart <- transposed_hallmark[, selected_hallmark]

## save
library(openxlsx)
write.xlsx(final_hallmart,
           rowNames=T,
           "review_selected_hallmark_matrix.xlsx")

```

5.2 selected_BP matrix

```

## plot_hallmark
DefaultAssay(selected_obj)<-"go_BP_ssGSEA_normalized"

```

```

selected_BP<-c( "GOBP-REGULATION-OF-CHROMATIN-ORGANIZATION",
"GOBP-NEGATIVE-REGULATION-OF-CHROMATIN-ORGANIZATION",
"GOBP-REGULATION-OF-LYSOSOMAL-LUMEN-PH",
"GOBP-LYSOSOMAL-LUMEN-ACIDIFICATION" )

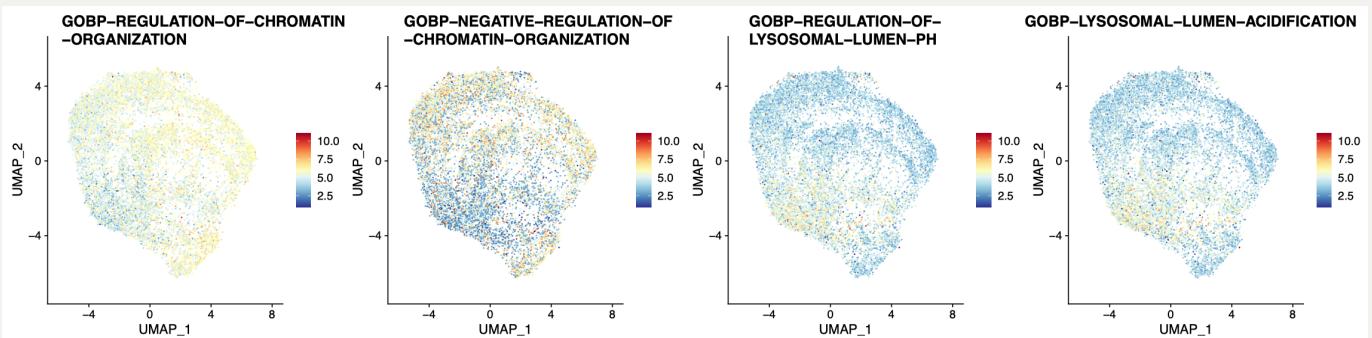
library(RColorBrewer)
plot_color = rev(brewer.pal(11, "RdYlBu"))

p1<-FeaturePlot(selected_obj,
                  features = selected_BP,
                  cols = plot_color,
                  # pt.size = 2,
                  ncol = 4)

# scale_color_gradientn(colors = plot_color)
# +theme(plot.title = element_text(color="red",
# size=10,face="bold.italic"))

ggsave("review_final_BP_plot.pdf",
       plot = p1, width = 20, height = 5)

```



```

library(Matrix)

BP_matrix=as.data.frame(selected_obj@assays$go_BP_ssgSEA_normalized
@data)

## transpose the row and column

```

```

transposed_BP <- as.data.frame(t(BP_matrix))

## select the intested columns.
final_BP <- transposed_BP[, selected_BP]

## save
library(openxlsx)
write.xlsx(final_BP,
           rowNames=T,
           "review_selected_BP_matrix.xlsx")

```

5.3 selected_gene matrix

```

## plot_hallmark
DefaultAssay(selected_obj)<- "RNA"

selected_gene<-c( "CREBRF", "DDIT3", "XBP1", "MAFF",
                 "PPP1R15A", "HSPB8", "KRT14")

library(Matrix)
gene_matrix=as.data.frame(selected_obj@assays$RNA@data)

## transpose the row and column
transposed_gene <- as.data.frame(t(gene_matrix))

## select the intested columns.
final_gene <- transposed_gene[, selected_gene]

## save
library(openxlsx)
write.xlsx(final_gene,
           rowNames=T,
           "review_selected_Gene_matrix.xlsx")

```