展示数据是来自 NCBI（national center for biotechnology information ）网站公开的 PBMC （Human peripheral blood mononuclear cells）数据集，PBMC3K 和 PBMC1K。「通常，我们的实验数据是来自湖南省儿童医院的病人样本。经过采样，预处理，高通量测序等一系列步骤后，生成的临床数据集」

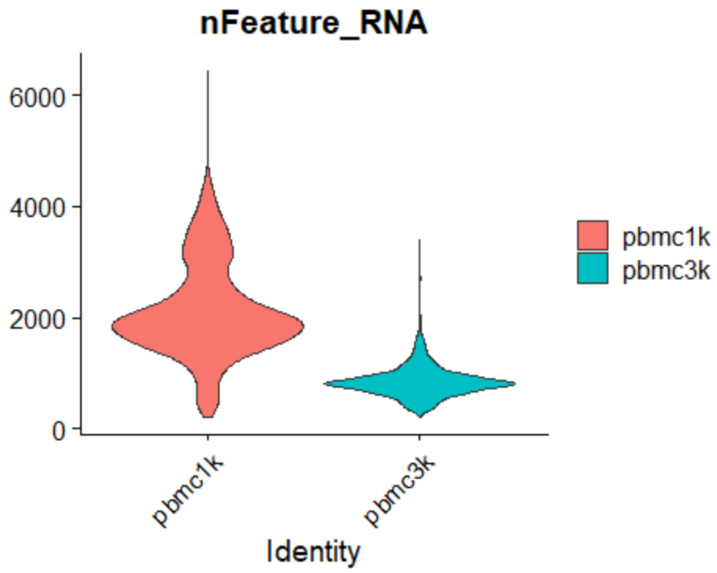
#为代码目的

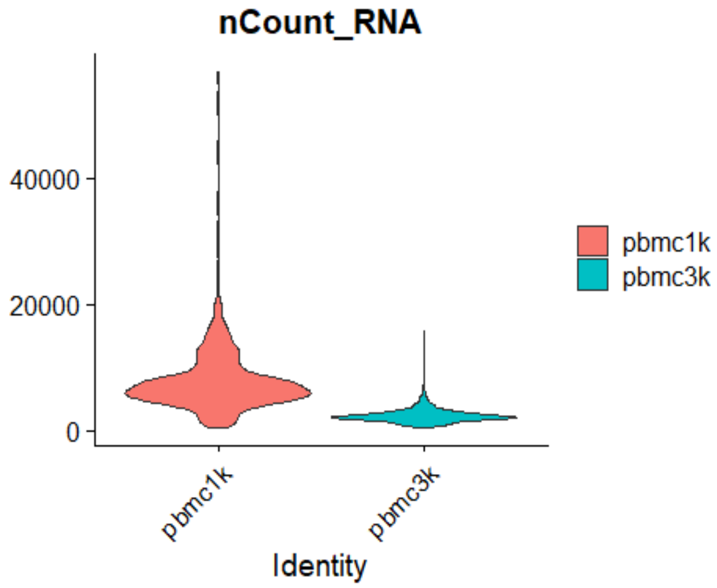
###为代码处理数据的步骤

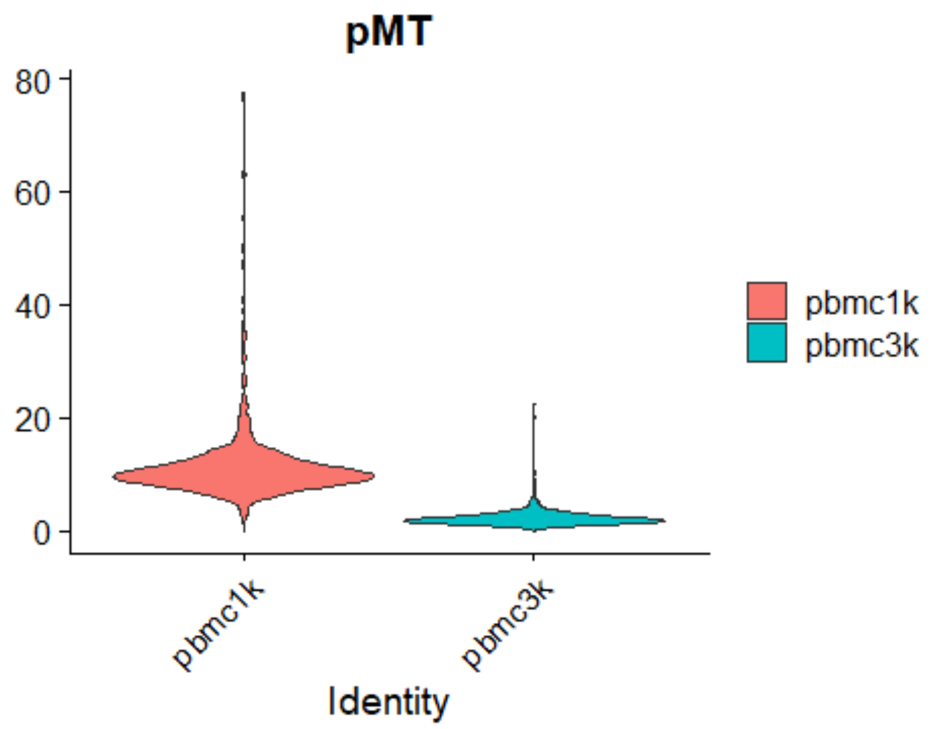
图片为可视化结果

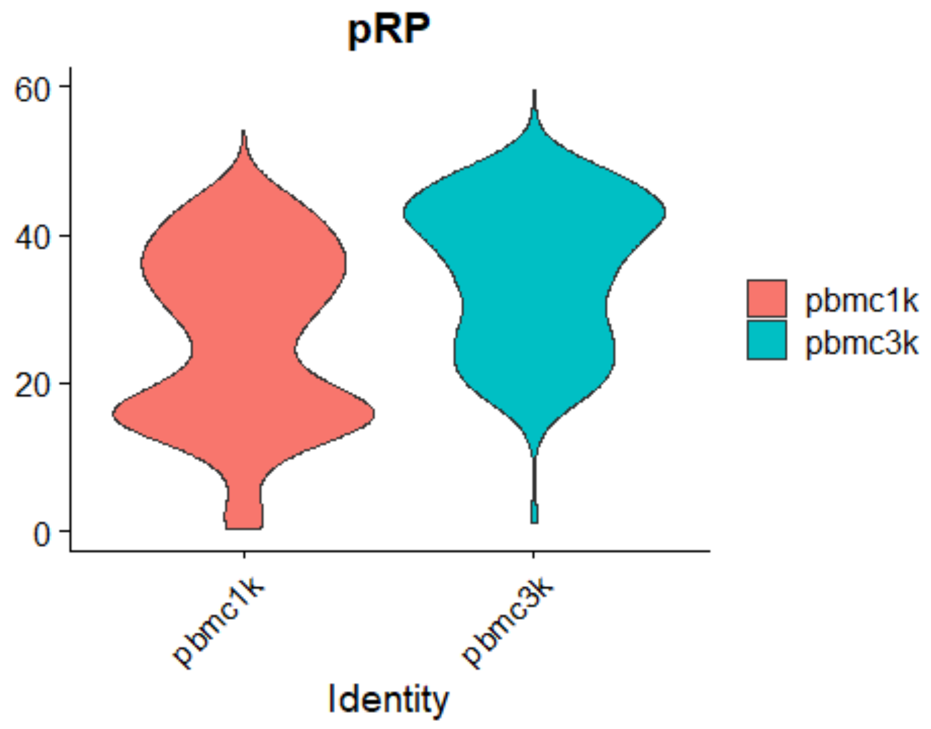
##为代码自动生成的解释

```
# Data Loading and QC

### Load the PBMC datasets
data("pbmc3k")
pbmc.data <- Read10X(data.dir = "E:/MangeXU/PBMC_Presentation/PBMC1K")

### Initialize the Seurat object with the raw (non-normalized data).
pbmc1k <- CreateSeuratObject(counts = pbmc.data, project = "pbmc1k", mi
n.cells = 3, min.features = 200)

### merge data sets
seu_obj <- merge(pbmc1k, y = pbmc3k)
seu_obj <- ScaleData(seu_obj)

### calculate mitochondrial, hemoglobin and ribosomal gene counts
seu_obj <- PercentageFeatureSet(seu_obj, pattern = "^MT-", col.name = "
pMT")
seu_obj <- PercentageFeatureSet(seu_obj, pattern = "^HBA|^HBB", col.nam
e = "pHB")
seu_obj <- PercentageFeatureSet(seu_obj, pattern = "^RPS|^RPL", col.nam
e = "pRP")

qcparams <- c("nFeature_RNA", "nCount_RNA", "pMT", "pRP")
for (i in seq_along(qcparams)){
  print(VlnPlot(object = seu_obj, features = qcparams[i], group.by = "o
rig.ident", pt.size = 0))
}
```


nFeature_RNA
6000
4000
2000
0
pbmc1k
pbmc3k
Identity
pbmc1k
pbmc3k

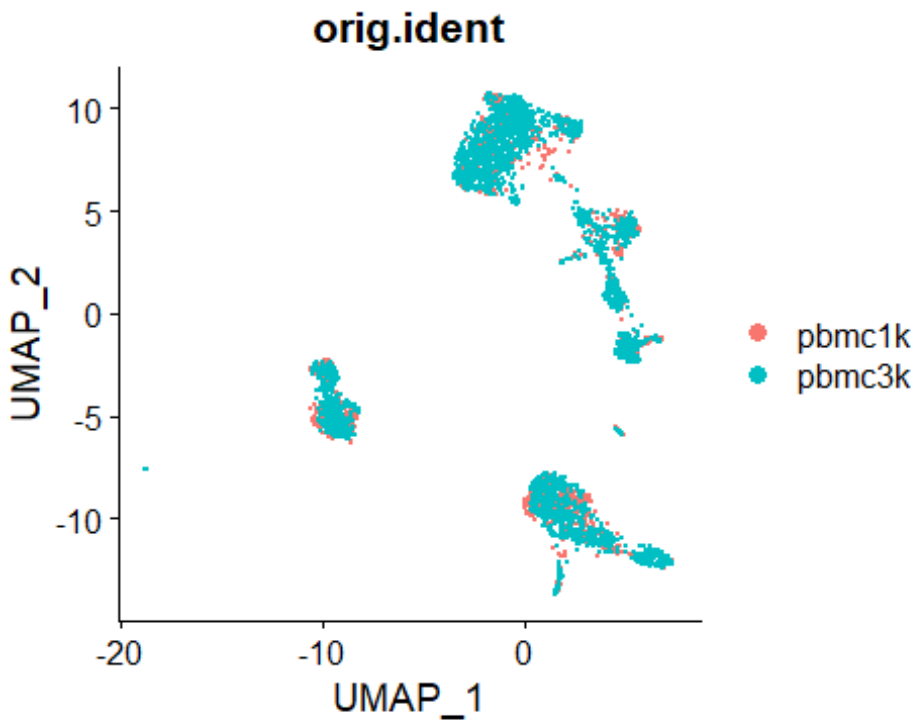

nCount_RNA
40000
20000
0
pbmc1k
pbmc3k
Identity
pbmc1k
pbmc3k



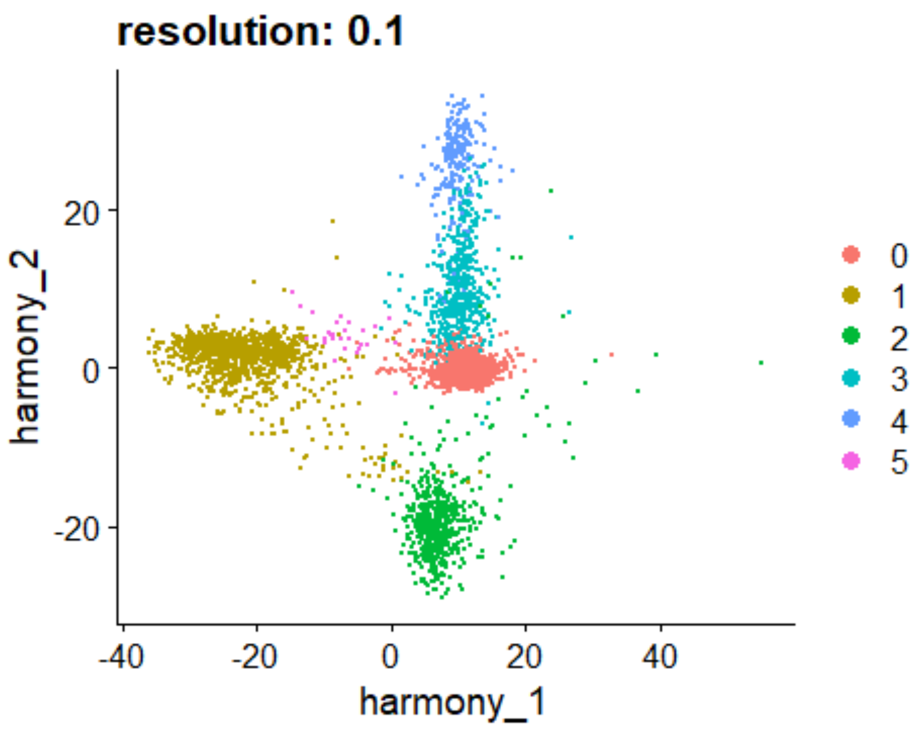


```
### clear environment
remove(pbmc1k)
remove(pbmc3k)
remove(pbmc.data)
```

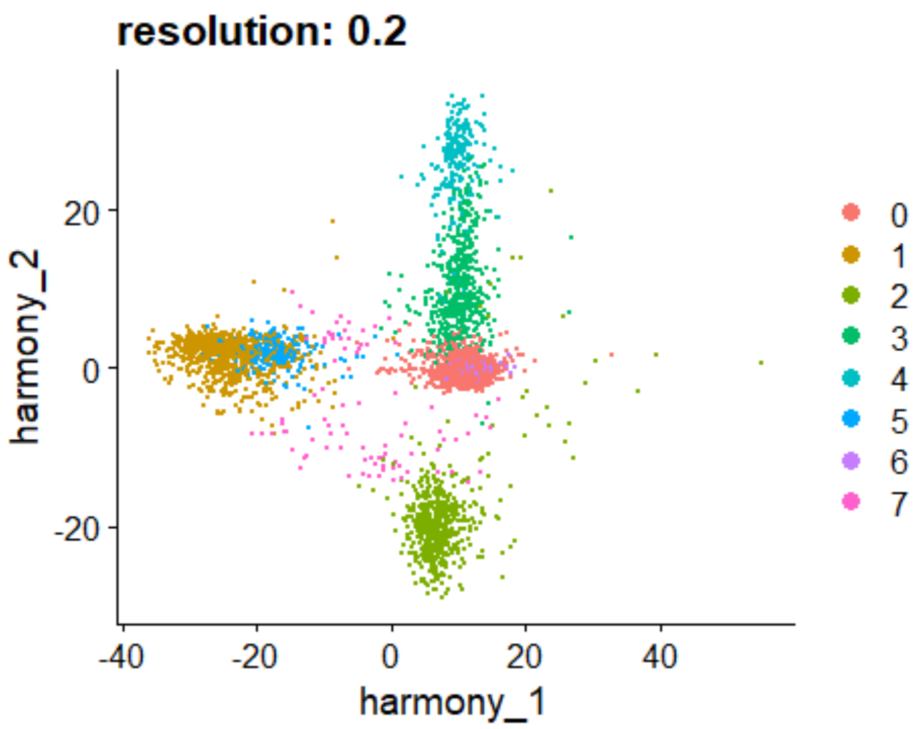
```
# Data Filtering
seu_obj <- subset(seu_obj, subset = nFeature_RNA > 200 & nFeature_RNA <
 5000 & pMT < 20)
seu_obj <- NormalizeData(seu_obj, normalization.method = "LogNormalize
", scale.factor = 10000)
seu_obj <- NormalizeData(seu_obj)


# Identify the 10 most highly variable genes
seu_obj <- FindVariableFeatures(seu_obj, selection.method = "vst", nfea
tures = 2000)


#dimensional reduction
seu_obj <- SCTransform(seu_obj, verbose = T, vars.to.regress = c("nCoun
t_RNA", "pMT"), conserve.memory = T)

seu_obj <- RunPCA(seu_obj)

seu_obj <- RunHarmony(seu_obj, group.by.vars="orig.ident", assay.use="S
CT", max.iter.harmony = 20)

seu_obj <- RunUMAP(seu_obj,reduction = "harmony", dims = 1:30)

seu_obj <- FindNeighbors(seu_obj, reduction = "harmony",dims = 1:30)

## Computing nearest neighbor graph
## Computing SNN

DimPlot(seu_obj, reduction = "umap", group.by = "orig.ident")
```



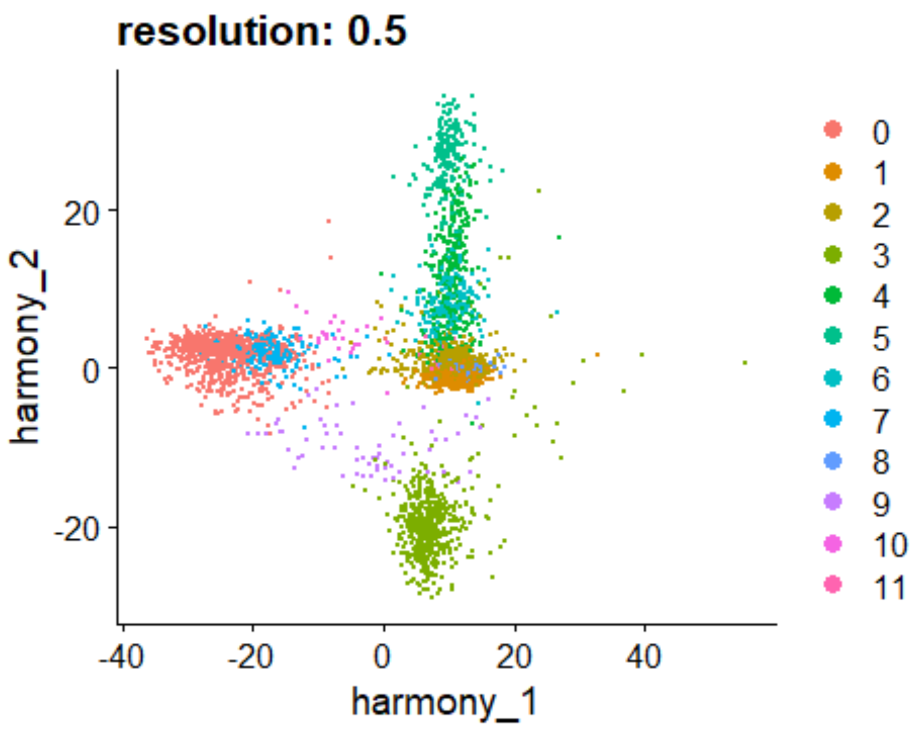
```
for(res in c(0.1, 0.2, 0.5, 0.7,1.0)){
seu_obj <- FindClusters(seu_obj, prefix = "SCT_snn_res.", resolution =r
es,verbose = FALSE)}

for (i in c(0.1, 0.2, 0.5, 0.7,1.0)) {
  seu_obj <- FindClusters(seu_obj, resolution = i)
  print(DimPlot(seu_obj, reduction = "harmony") + labs(title = paste0("
resolution: ", i)))
}

## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van
Eck
##
## Number of nodes: 3808
## Number of edges: 182202
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9681
## Number of communities: 6
## Elapsed time: 0 seconds
```



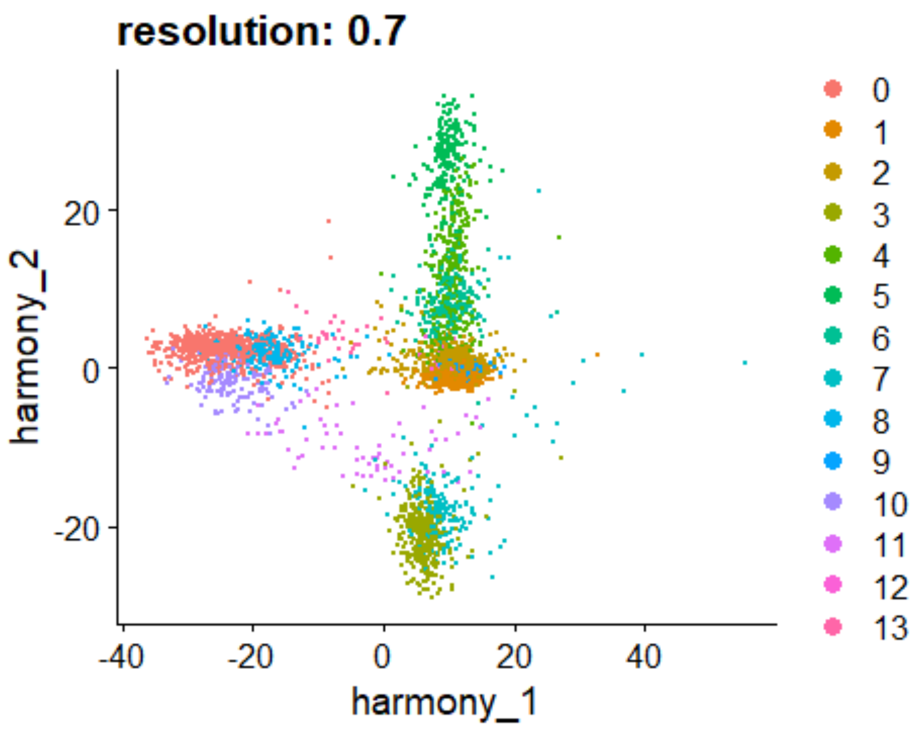
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van
Eck
##
## Number of nodes: 3808
## Number of edges: 182202
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9467
## Number of communities: 8
## Elapsed time: 0 seconds
```



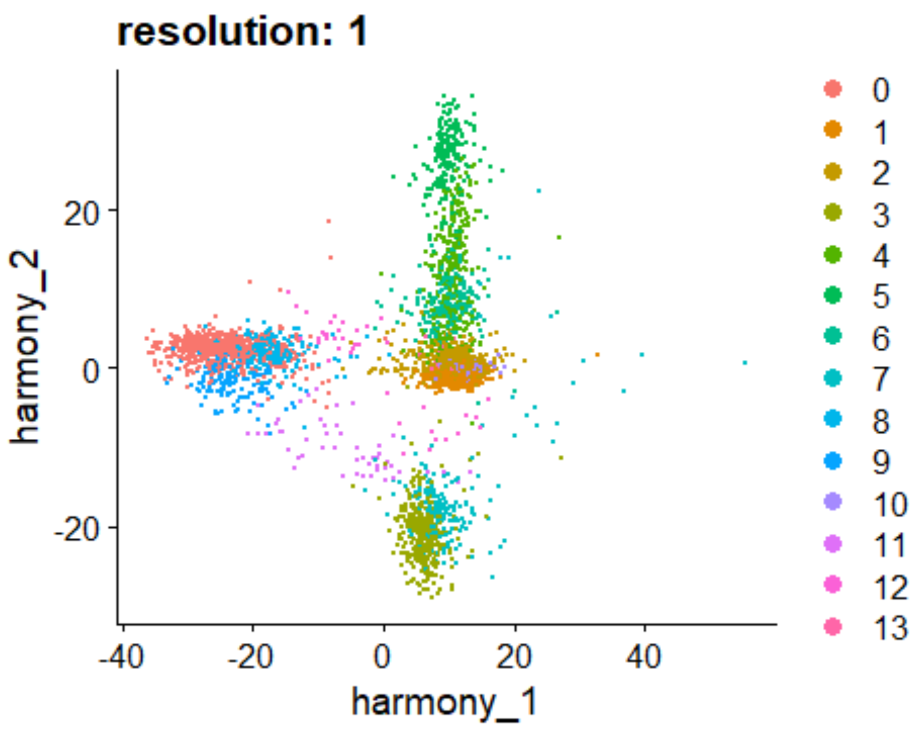
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van
Eck
##
## Number of nodes: 3808
## Number of edges: 182202
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8918
## Number of communities: 12
## Elapsed time: 0 seconds
```



```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van
Eck
##
## Number of nodes: 3808
## Number of edges: 182202
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8680
## Number of communities: 14
## Elapsed time: 0 seconds
```

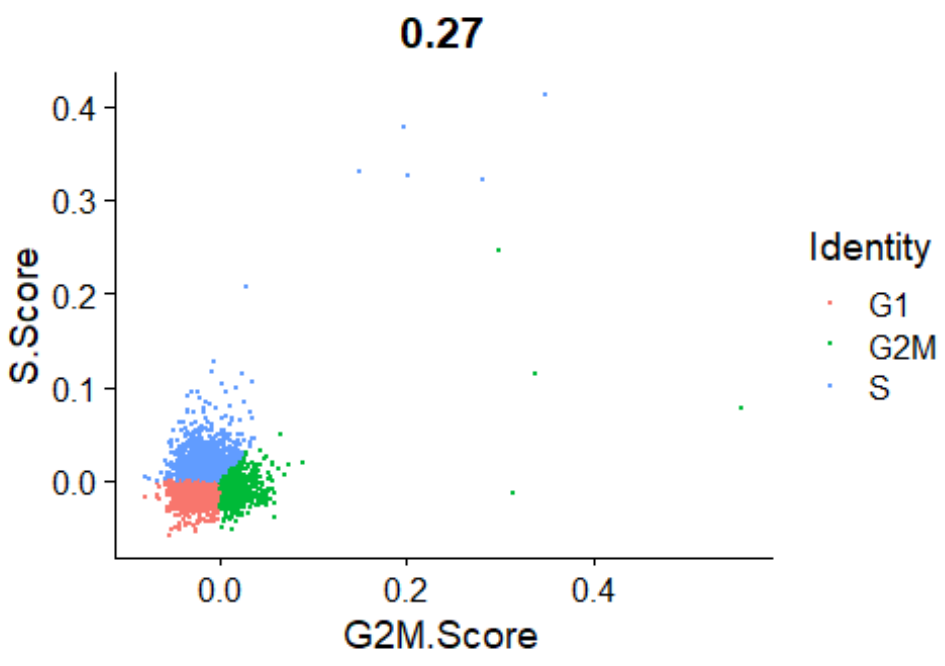


```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van
Eck
##
## Number of nodes: 3808
## Number of edges: 182202
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8339
## Number of communities: 14
## Elapsed time: 0 seconds
```



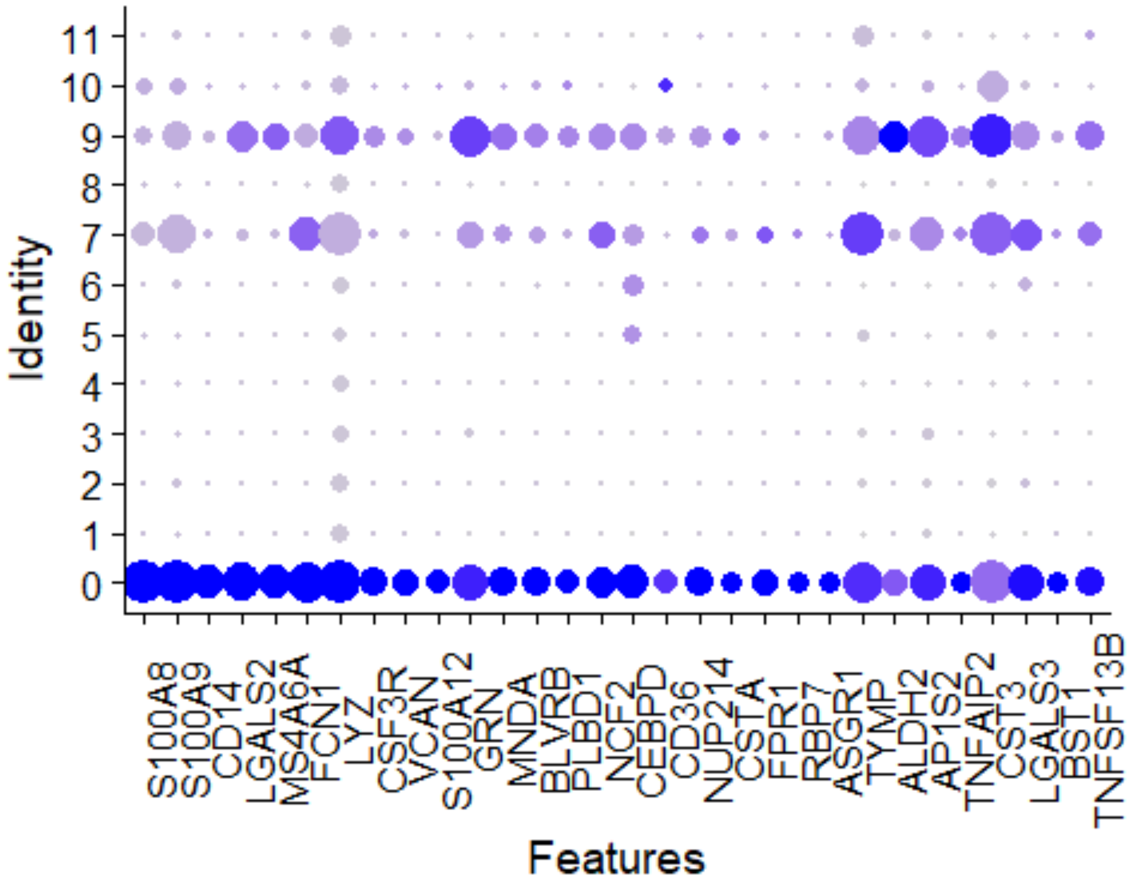
```
res_clustree<-clustree(seu_obj,prefix = "SCT_snn_res.")
ggsave("res_clustree.png",path = "PBMC_Presentation/annotation",height
= 10)

## Saving 5 x 10 in image

# Cell cycle scoring
### add cell cycle, cc.genes loaded with Seurat
s.genes <- cc.genes$s.genes
g2m.genes <- cc.genes$g2m.genes

score_cc <- function(seu_obj) {
  seu_obj <- CellCycleScoring(seu_obj, s.genes, g2m.genes)
  seu_obj@meta.data$CC.Diff <- seu_obj@meta.data$S.Score - seu_obj@met
a.data$G2M.Score
  return(seu_obj)
}

seu_obj <- score_cc(seu_obj)

## Warning: The following features are not present in the object: DTL,
MLF1IP,
## EXO1, E2F8, not searching for symbol synonyms

## Warning: The following features are not present in the object: FAM64
A, BUB1,
## HJURP, CDCA3, TTK, CDC25C, KIF2C, DLGAP5, CDCA2, ANLN, NEK2, not sea
```

```
rching for
## symbol synonyms

FeatureScatter(seu_obj, "G2M.Score", "S.Score", group.by = "Phase", pt.
size = .1) + coord_fixed(ratio = 1)
```

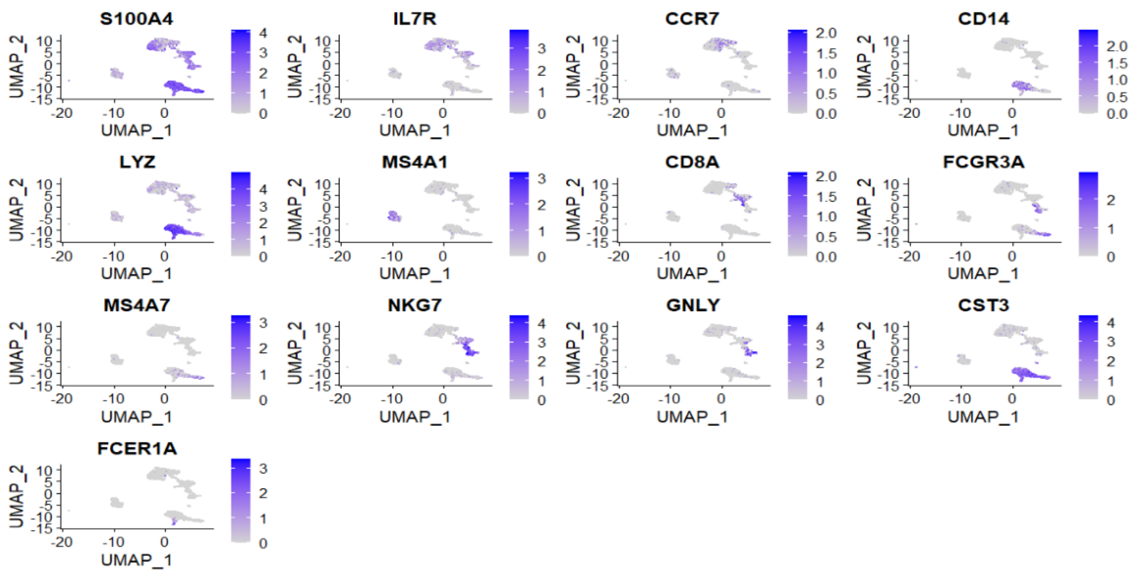


```
#save file
saveRDS(seu_obj, file = "E:/MangeXU/PBMC_Presentation/all_SCTransform.R
DS")

#SingleR

data_for_SingleR <- GetAssayData(seu_obj, slot="data")
clusters=seu_obj@meta.data$SCT_snn_res.0.5

humanImmu <- BlueprintEncodeData()

pred.humanImmu <- SingleR(test = data_for_SingleR, ref = humanImmu, lab
els = humanImmu$label.main, clusters = clusters, assay.type.test = "log
counts", assay.type.ref = "logcounts")

humangene <- HumanPrimaryCellAtlasData()

pred.humangene <- SingleR(test = data_for_SingleR, ref = humangene, lab
els = humangene$label.fine,
                          clusters = clusters, assay.type.test = "logco
unts", assay.type.ref = "logcounts")
```

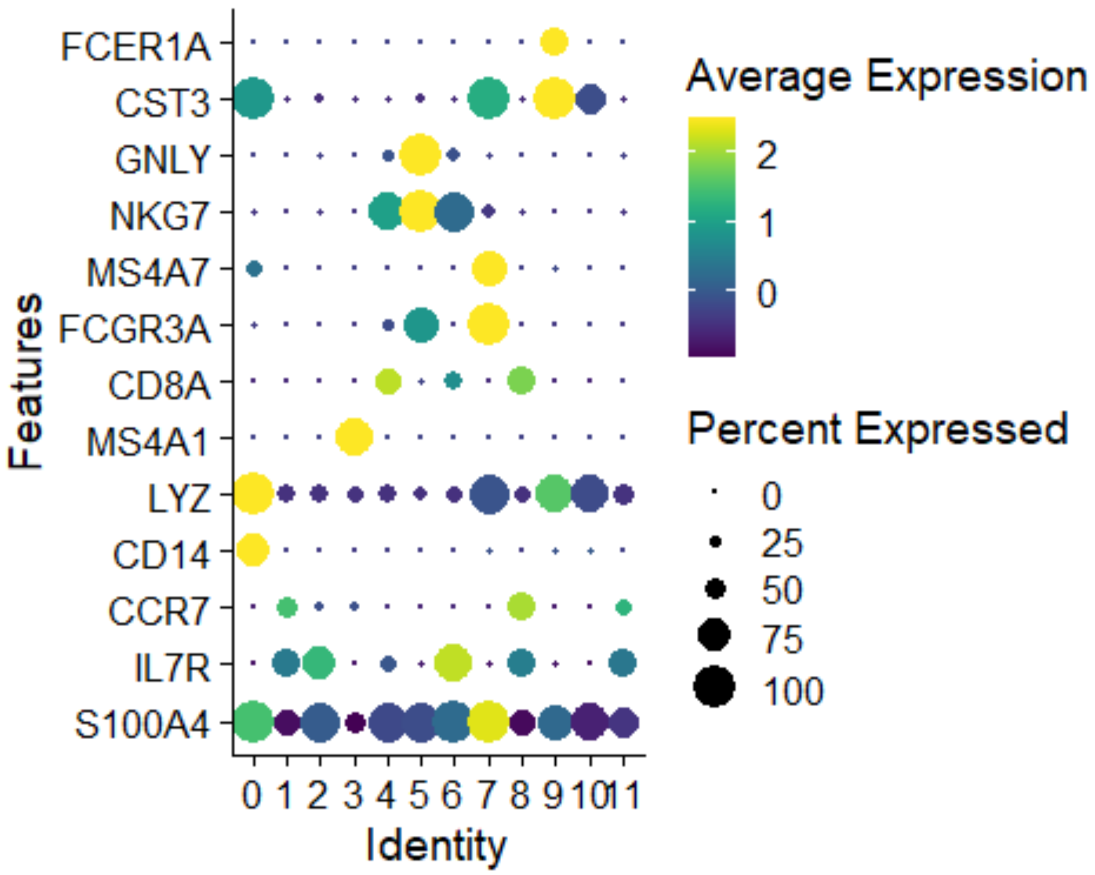
```
humancell <- NovershternHematopoieticData()

pred.humancell <- SingleR(test = data_for_SingleR, ref = humancell, lab
els = humancell$label.fine,
                          clusters = clusters, assay.type.test = "logco
unts", assay.type.ref = "logcounts")

cellType=data.frame(ClusterID=levels(seu_obj@meta.data$SCT_snn_res.0.
5),
                    humanImmu=pred.humanImmu$labels,
                    humangene=pred.humangene$labels,
                    humancell=pred.humancell$labels)

cellType

##    ClusterID    humanImmu                   humangene
## 1          0    Monocytes              Monocyte:CD16-
## 2          1 CD4+ T-cells  T_cell:CD4+_central_memory
## 3          2 CD4+ T-cells  T_cell:CD4+_central_memory
## 4          3      B-cells                B_cell:Naive
## 5          4 CD8+ T-cells                 T_cell:CD8+
## 6          5     NK cells                     NK_cell
## 7          6 CD8+ T-cells T_cell:CD4+_effector_memory
## 8          7    Monocytes              Monocyte:CD16+
## 9          8 CD8+ T-cells           T_cell:CD4+_Naive
## 10         9    Monocytes              Monocyte:CD16-
## 11        10    Monocytes              Monocyte:CD16-
## 12        11 CD4+ T-cells  T_cell:CD4+_central_memory
##                                    humancell
## 1                                  Monocytes
## 2                         CD4+ Central Memory
## 3                        CD4+ Effector Memory
## 4                              Naive B cells
## 5                        CD8+ Effector Memory
## 6                     CD8+ Effector Memory RA
## 7                        CD8+ Effector Memory
## 8             Mature NK cells_CD56- CD16+ CD3-
## 9                          Naive CD8+ T cells
## 10                    Myeloid Dendritic Cells
## 11 Granulocytes (Neutrophilic Metamyelocytes)
## 12                       CD4+ Effector Memory


#COSG

Idents(seu_obj)<-"SCT_snn_res.0.5"
library(COSG)
COSG_markers <- cosg(
  seu_obj,
  groups='all',
  assay='RNA',
  slot='data',
  mu=1,
  n_genes_user=100
)

m1<- head(COSG_markers$names$`0`,30)
th= theme(axis.text.x = element_text(angle = 90))
DotPlot(seu_obj, features = m1,assay='RNA') + th +NoLegend()
```



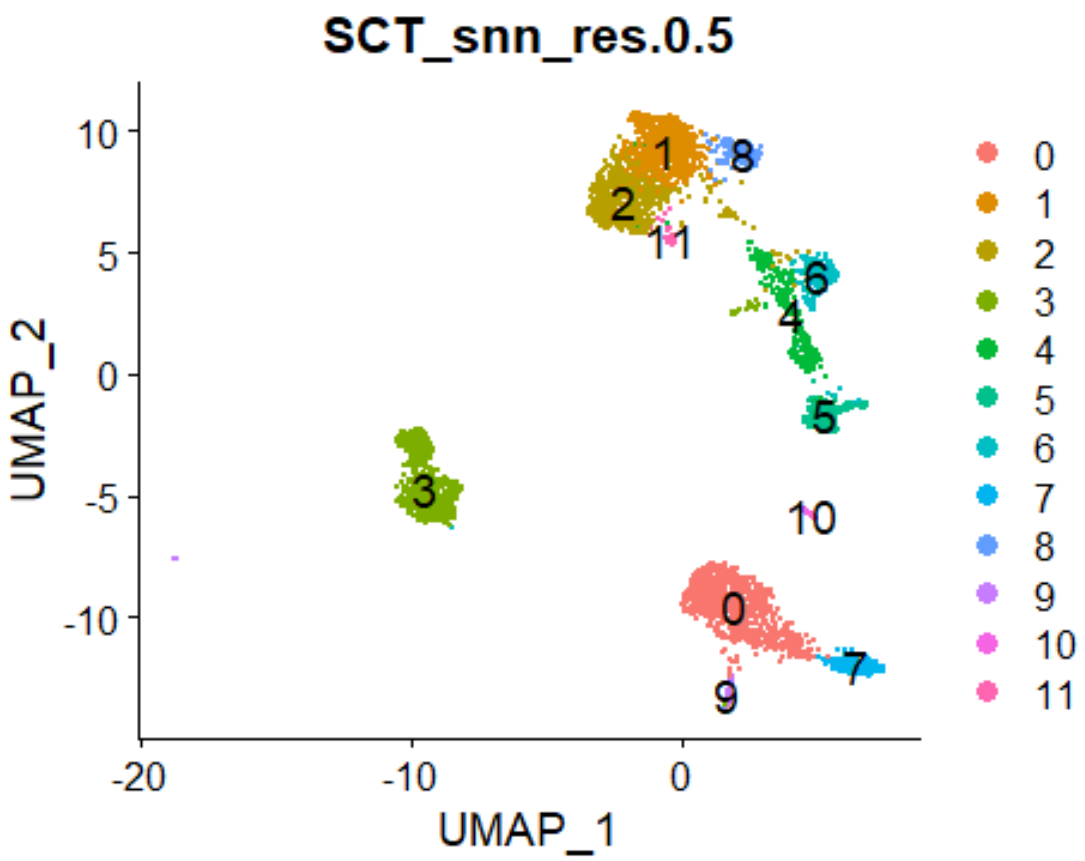
```
# Main cell type annotation

mainmarkers <- c("S100A4","IL7R","CCR7","CD14","LYZ","MS4A1","CD8A",  "
FCGR3A", "MS4A7","NKG7","GNLY", "CST3",  "FCER1A")

FeaturePlot(seu_obj, features = mainmarkers)
```



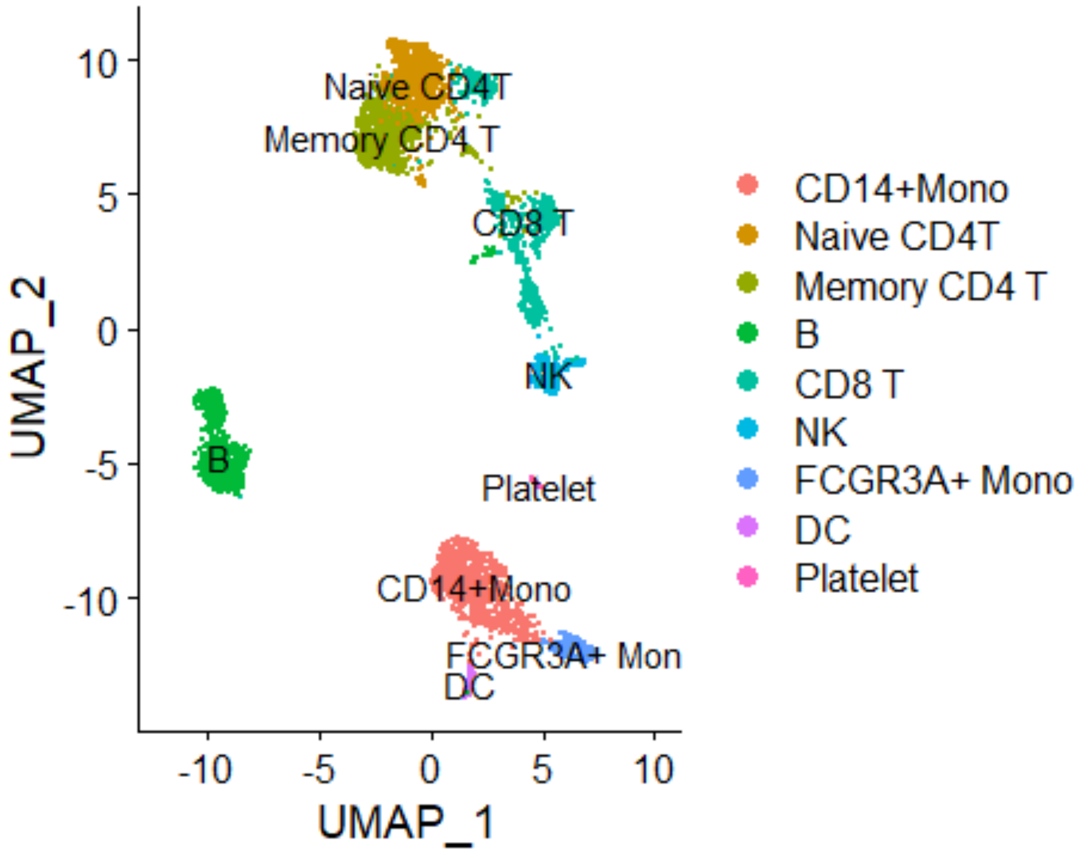
```
DotPlot(seu_obj, features = mainmarkers, group.by = "SCT_snn_res.0.5")
+ coord_flip() + scale_color_viridis()

## Scale for 'colour' is already present. Adding another scale for 'col
our',
## which will replace the existing scale.
```



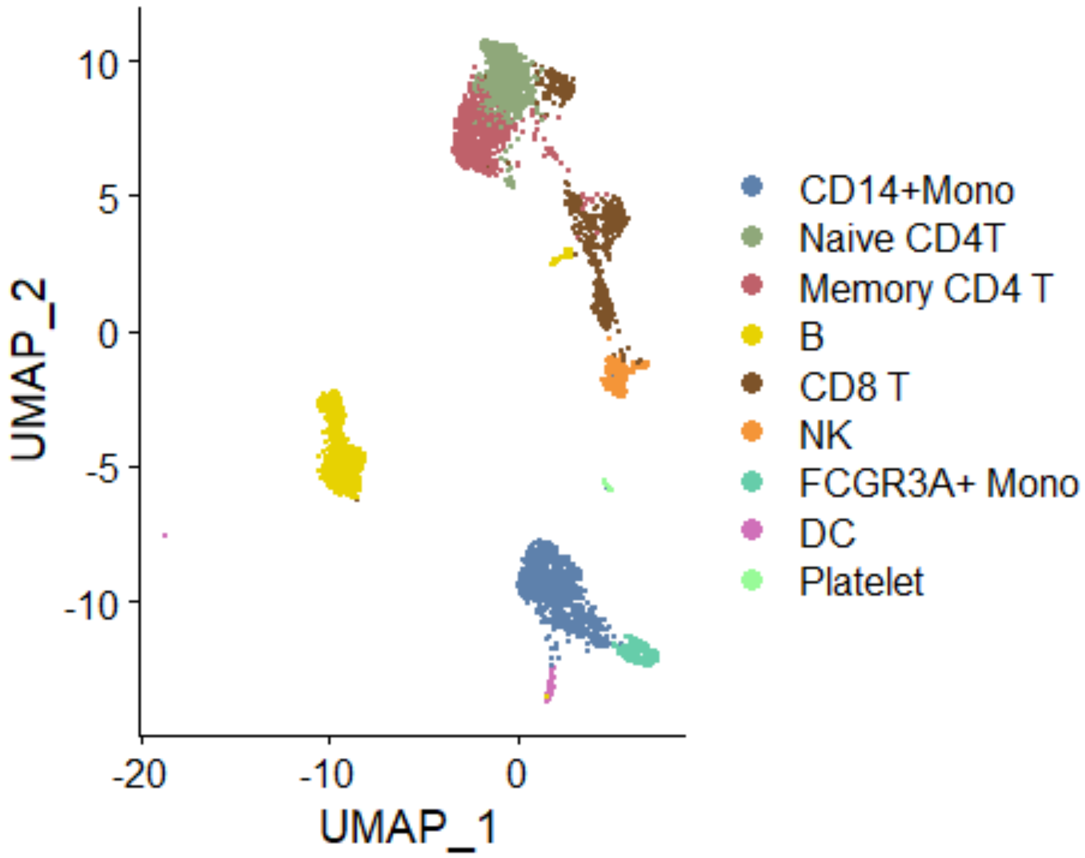
```
#ggsave2("DotPlot_mainmarkers.png", path = "PBMC_Presentation/annotatio
n", width = 30, height = 8, units = "cm")

DimPlot(seu_obj, group.by = "SCT_snn_res.0.5", label = T, label.size =
5)
```



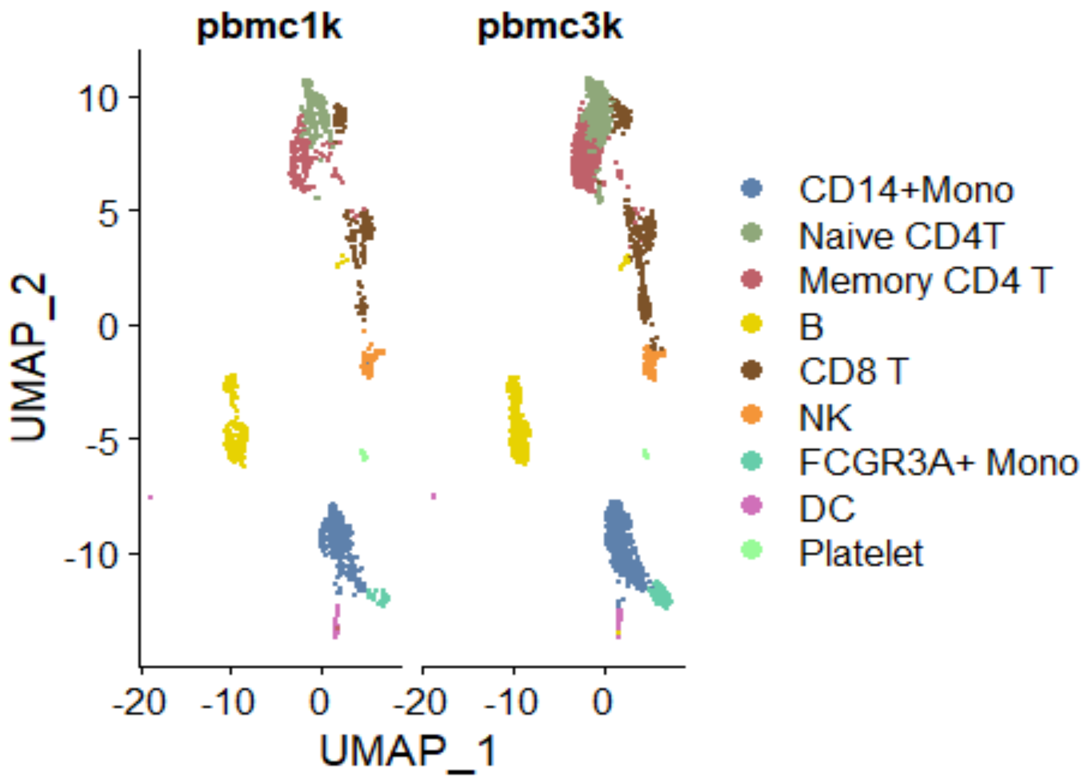
```
#ggsave2("DimPlot_all_clusters.png", path = "PBMC_Presentation/annotati
on", width = 20, height = 20, units = "cm")


new.cluster.ids <- c("CD14+Mono","Naive CD4T","Memory CD4 T","B","CD8 T
","NK","CD8 T","FCGR3A+ Mono","CD8 T","DC","Platelet","Naive CD4T")
names(new.cluster.ids) <- levels(seu_obj)
seu_obj <- RenameIdents(seu_obj, new.cluster.ids)
seu_obj@meta.data$cell_type <- Idents(seu_obj)
DimPlot(seu_obj, reduction = "umap", label = TRUE, pt.size = 0.5)+xlim
(-12,10)
```



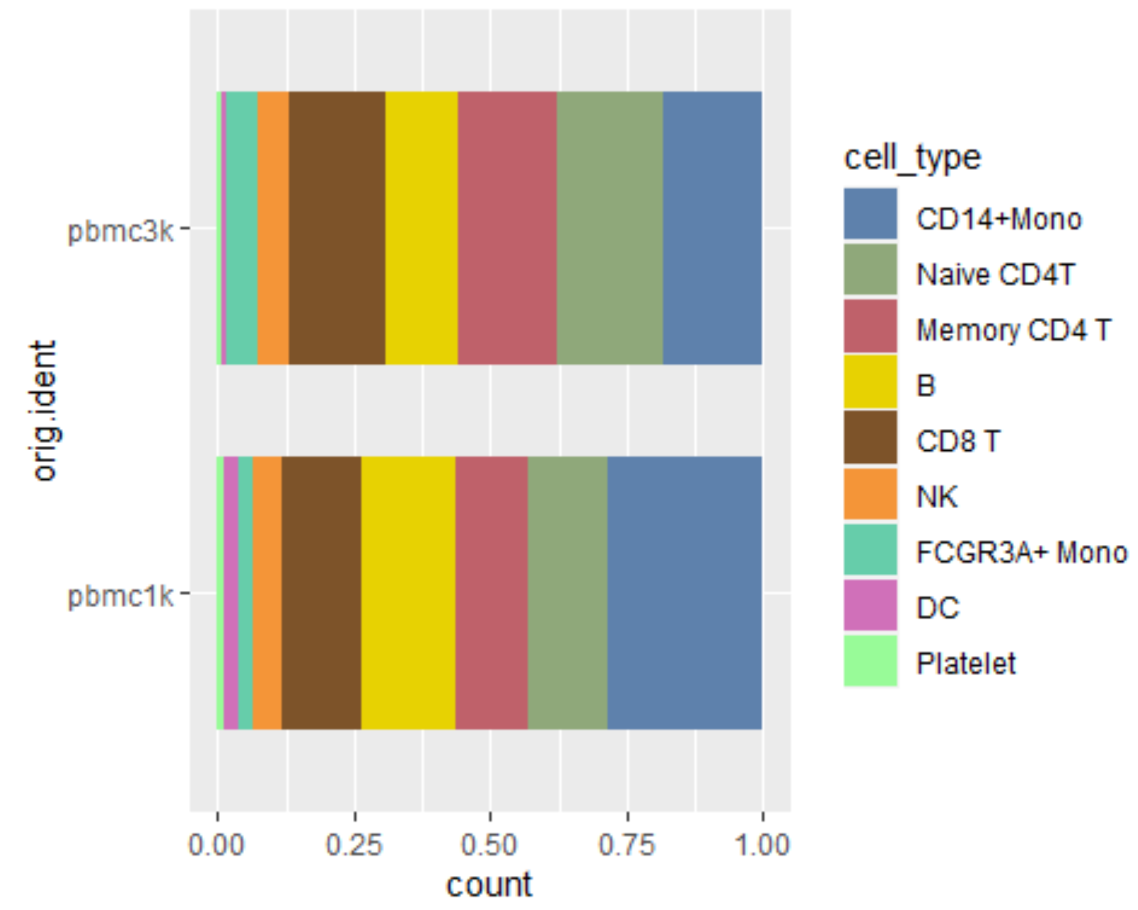
```
#ggsave2("DimPlot_anno_clusters.png", path = "PBMC_Presentation/annotat
ion", width = 20, height = 20, units = "cm")

###save file
saveRDS(seu_obj, file = "PBMC_Presentation/pbmc_final.rds")

#annotation plots
seu_obj$cell_type<- factor(seu_obj$cell_type)
seu_obj$orig.ident<- factor(seu_obj$orig.ident)

DimPlot(seu_obj, pt.size = 0.1,label = F, label.size = 3)+scale_color_p
aletteer_d("basetheme::brutal")
```



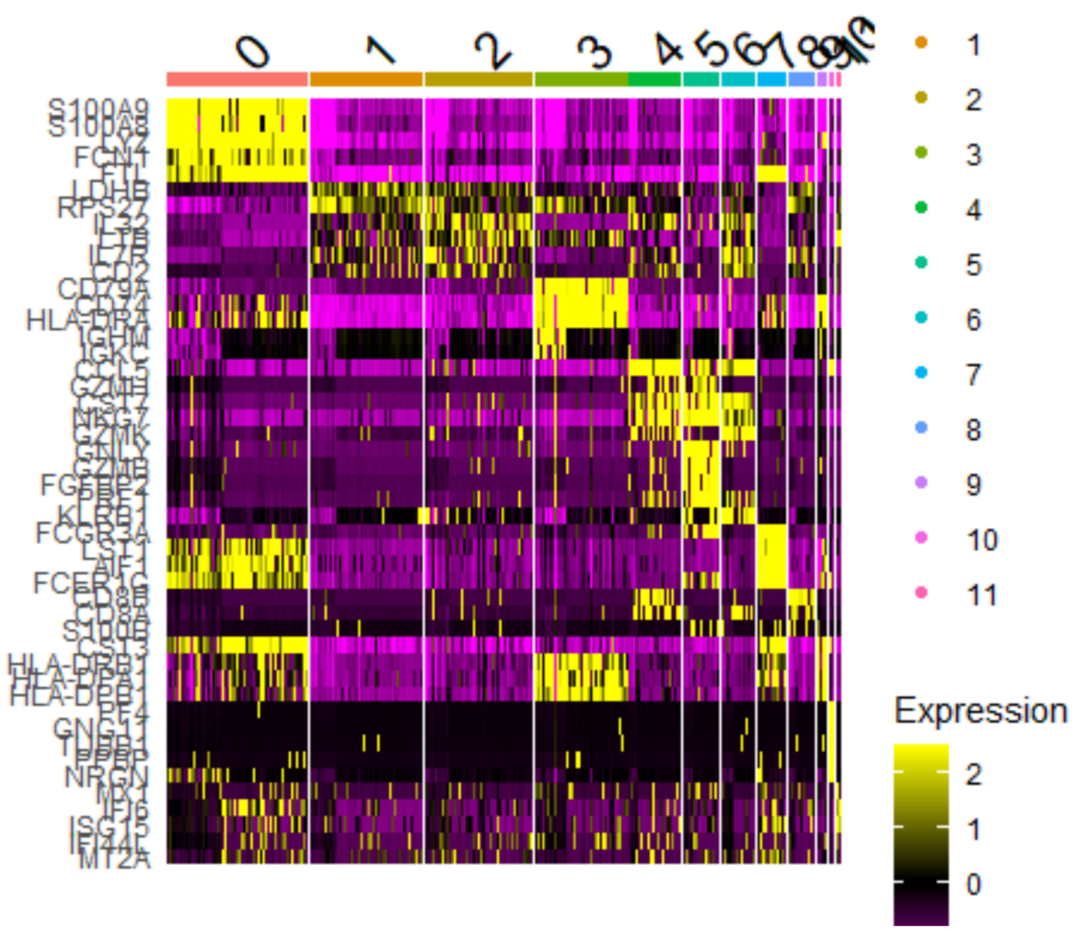
```
#ggsave2("seu_obj.png", path = "PBMC_Presentation/annotation", width =
20, height = 15, units = "cm")

DimPlot(seu_obj, split.by = "orig.ident",pt.size = 0.1)+scale_color_pal
etteer_d("basetheme::brutal")
```



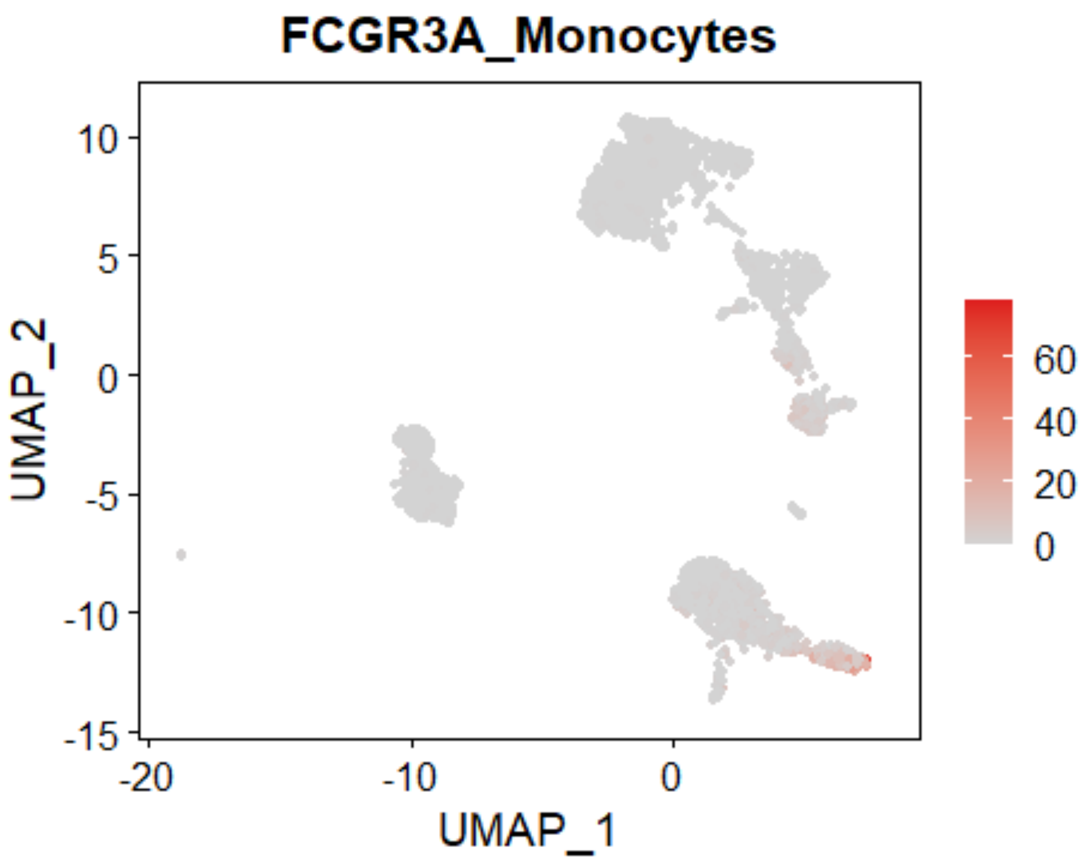
```
#ggsave2("seu_obj_samp.png", path = "PBMC_Presentation/annotation", wid
th = 39, height = 10, units = "cm")


type <- FetchData(seu_obj, vars = c("cell_type", "orig.ident")) %>%
  mutate(cell_type = factor(cell_type)) %>%
  mutate(orig.ident = factor(orig.ident))

ggplot(data = type) +
  geom_bar(mapping = aes(x = orig.ident,fill = cell_type), position = "
fill", width = 0.75) +
  scale_fill_manual(values =paletteer_d("basetheme::brutal")) +
  coord_flip()
```



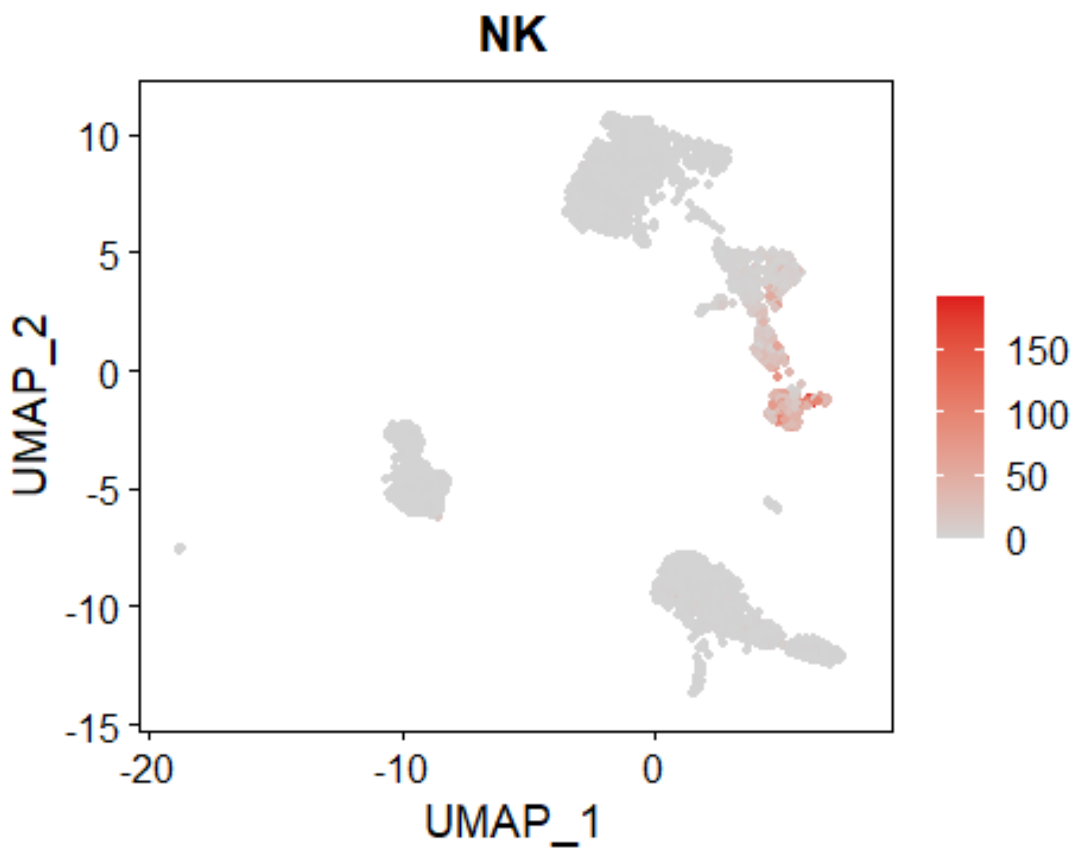
```r
#ggsave2("celltype_samp_barplot.png", path = "PBMC_Presentation/annotat
ion", width = 20, height = 10, units = "cm")

Idents(seu_obj) <- seu_obj$SCT_snn_res.0.5


#Find all markers

cluster.markers <- FindAllMarkers(object = seu_obj, only.pos = TRUE, mi
n.pct = 0.25, thresh.use = 0.25)

#write_csv(cluster.markers, file ="../clusterMarkers.csv")
top5.sub <- cluster.markers %>% group_by(cluster) %>% top_n(5, avg_log2
FC)
DoHeatmap(seu_obj,features = top5.sub$gene, label = TRUE)
```



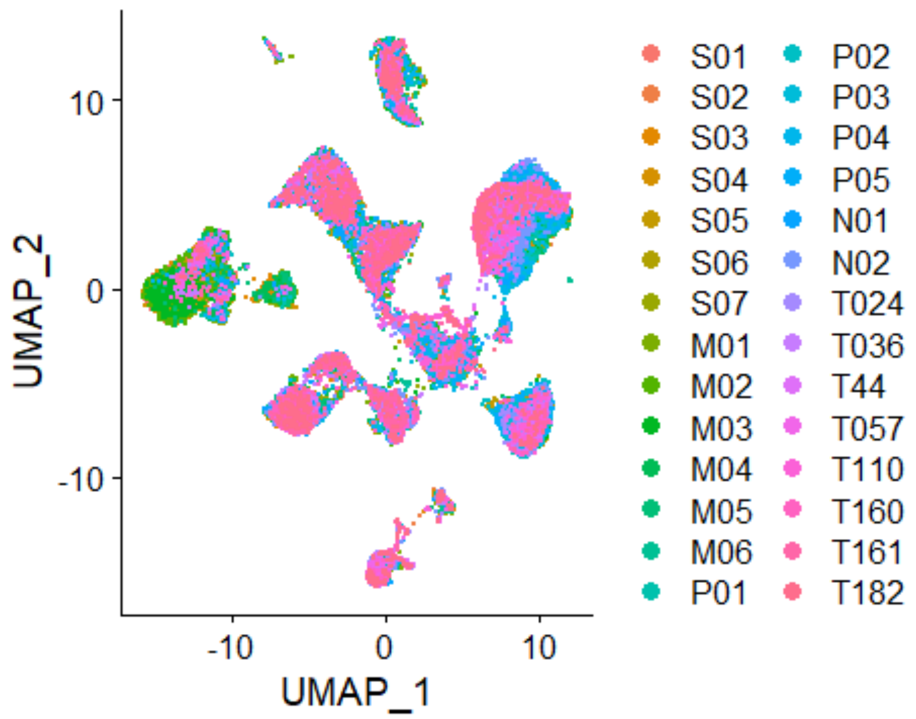
```
#ggsave2("Heatmap_top5_sub.png", path = "PBMC_Presentation/annotation",
 width = 80, height = 40, units = "cm")

GL = list(FCGR3A_Monocytes = mainmarkers[8:9],
          NK = mainmarkers[10:11]
)

FCGR3A_Monocytes=seu_obj@assays$RNA@counts[GL[[1]],]
NK=seu_obj@assays$RNA@counts[GL[[2]],]

seu_obj$FCGR3A_Monocytes = colSums(FCGR3A_Monocytes)
seu_obj$NK = colSums(NK)


FeaturePlot(seu_obj,"FCGR3A_Monocytes",cols = c("lightgrey" ,"#DE1F1F
"),slot = "data",label.size = 6,pt.size = 1.2)+annotate(geom = 'segment
', y = Inf, yend = Inf, color = 'black', x = -Inf, xend = Inf, size =
1)+annotate(geom = 'segment', x = Inf, xend = Inf, color = 'black', y =
 -Inf, yend = Inf, size = 0.5)
```



```
#ggsave2("FCGR3A+Monocytes_FeaturePlot.png", path = "E:/MangeXU/PBMC_Pr
esentation/annotation", width = 20, height = 20, units = "cm")

FeaturePlot(seu_obj,"NK",cols = c("lightgrey" ,"#DE1F1F"),slot = "data
",label.size = 6,pt.size = 1.2)+annotate(geom = 'segment', y = Inf, yen
d = Inf, color = 'black', x = -Inf, xend = Inf, size = 1)+annotate(geom
 = 'segment', x = Inf, xend = Inf, color = 'black', y = -Inf, yend = In
f, size = 0.5)
```



```
#ggsave2("NK_FeaturePlot.png", path = "E:/MangeXU/PBMC_Presentation/ann
otation", width = 20, height = 20, units = "cm")


#Save file

saveRDS(seu_obj, file = "E:/MangeXU/PBMC_Presentation/pbmc_final.RDS")

save.image(file = "PBMC_Pre_annotation.RData")


#亚群比例热图和箱型图

set.seed(123)
seu_obj_anno_10k <- readRDS("E:/MangeXU/seu_obj_anno_10k.RDS")
DimPlot(seu_obj_anno_10k)
```

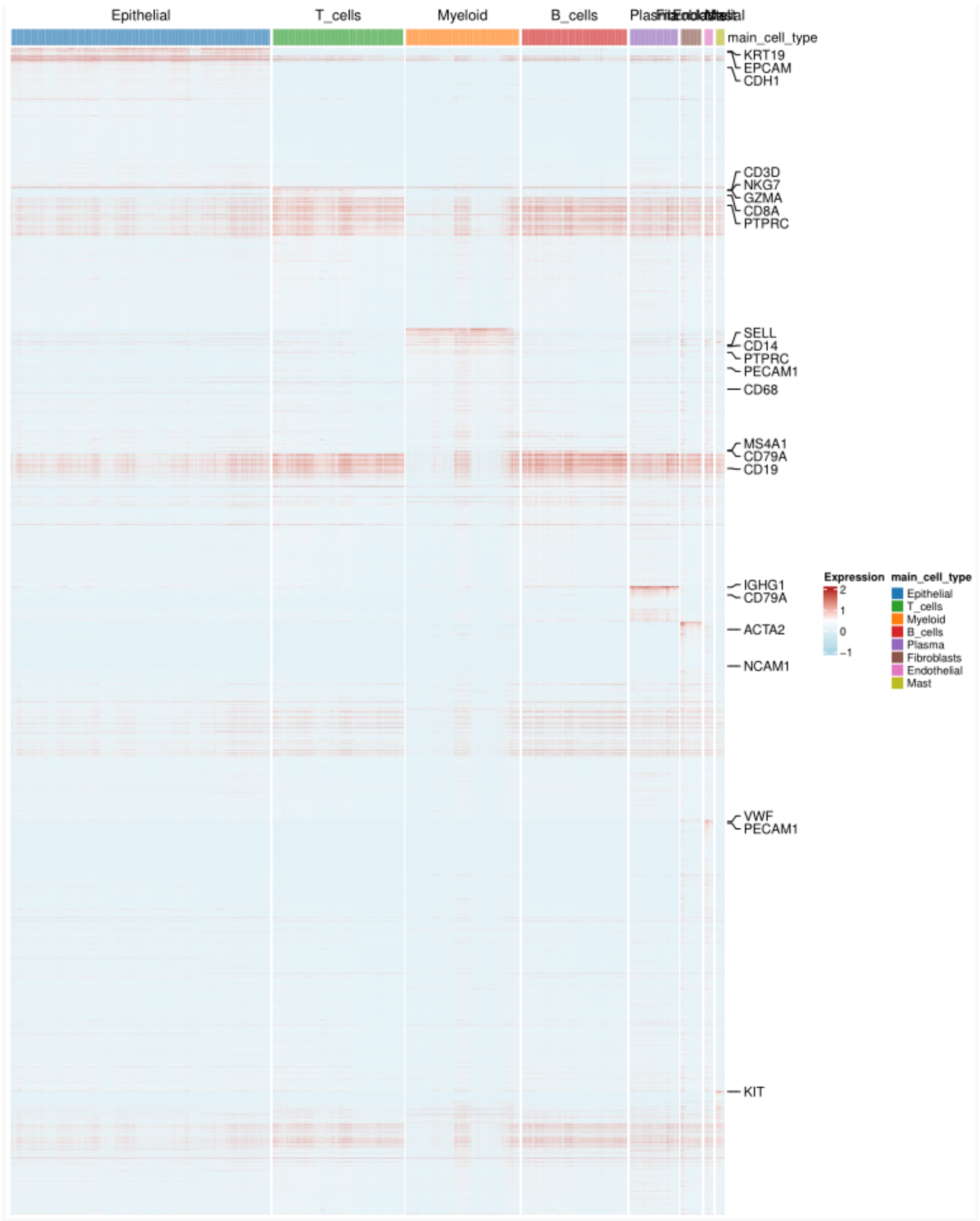


```
cluster_colors<- c("Epithelial"="#1F77B4FF","Myeloid"="#FF7F0EFF","T_ce
lls"="#2CA02CFF","B_cells"="#D62728FF",
               "Plasma"="#9467BDFF",
               "Fibroblasts"="#8C564BFF","Endothelial"="#E377C2FF",
               "Mast"="#BCBD22FF")
Idents(seu_obj_anno_10k)<- seu_obj_anno_10k@meta.data$main_cell_type

###随机取样函数
Sample_seob <- function(obj,group.by="seurat_clusters",sp.size=NULL,die
t="true",sp.total=1000) {
all <- obj
if (diet=="true") {
all <- DietSeurat(all,dimreducs = c('pca','umap'))
}

if (is.null(sp.size)) {
nlen <- length(unique(all@meta.data[,group.by]))
sp.size <- ceiling(sp.total/nlen)

}
ncellist <- c()
for (sc in unique(all@meta.data[,group.by])){
cellist <- colnames(all)[which(all@meta.data[,group.by] == sc)]
if (length(cellist) > sp.size) {
cellist=sample(cellist, sp.size)
}
ncellist <- c(ncellist,cellist)
}
all <- subset(all,cells=ncellist)
return(all)
}

table(seu_obj_anno_10k$main_cell_type)

##
##  Epithelial     T_cells     Myeloid     B_cells      Plasma Fibrobla
sts
##       10482        5266        4567        4246        1945
815
## Endothelial        Mast
##         349         330

sample_marker <- Sample_seob(seu_obj_anno_10k,sp.size = 327,group.by='m
ain_cell_type')

cluster.markers_samp <- FindAllMarkers(object = sample_marker, only.pos
 = TRUE, min.pct = 0.25, thresh.use = 0.25)

top10.samp <- cluster.markers_samp %>% group_by(cluster) %>% top_n(10,
avg_log2FC)

cluster.markers <- FindAllMarkers(object = seu_obj_anno_10k, only.pos =
 TRUE, min.pct = 0.25, thresh.use = 0.25)

top5.sub <- cluster.markers %>% group_by(cluster) %>% top_n(5, avg_log2
FC)

seu_obj_anno_10k<-ScaleData(seu_obj_anno_10k)

## Centering and scaling data matrix

cts<-seu_obj_anno_10k@assays$RNA@counts

###数据转换
cts<- GetAssayData(seu_obj_anno_10k, slot = "counts")
str(cts)

## Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
##   ..@ i       : int [1:42257490] 71 196 207 243 316 337 341 368 380
393 ...
##   ..@ p       : int [1:28001] 0 623 1209 3315 3960 4942 5345 6685 84
14 11511 ...
##   ..@ Dim     : int [1:2] 28004 28000
##   ..@ Dimnames:List of 2
##   .. ..$ : chr [1:28004] "AL627309.1" "AL627309.5" "AL627309.4" "AL6
69831.2" ...
##   .. ..$ : chr [1:28000] "S01_AAAGAACCATAGGTTC-1" "S01_AAAGGATGTGGCT
```

```
AGA-1" "S01_AAAGGATTCATTCGGA-1" "S01_AAAGGGCAGGTAACTA-1" ...
##   ..@ x        : num [1:42257490] 1 3 1 1 1 1 1 3 1 1 ...
##   ..@ factors : list()

cts <- log10(cts + 1)
head(seu_obj_anno_10k$main_cell_type)


## 8 Levels: Epithelial T_cells Myeloid B_cells Plasma ... Mast

cts <- as.matrix(cts[cluster.markers$gene, names(new_cluster)])

###注释及配色
ha <- HeatmapAnnotation(
  main_cell_type = new_cluster,
  col = list(main_cell_type=cluster_colors)
  )

###主图颜色
f1 = colorRamp2(seq(-1, 2, length = 3), c("lightblue","white", "firebri
ck"))

###展示目标基因
gene_anno <- read.table('E:/gene_anno.txt', header = T, check.names = F
ALSE)
head(gene_anno)

##     gene
## 1   CD14
## 2   CD1C
## 3 FCGR3A
## 4   CD3D
## 5   CD8A
## 6  NCAM1

genelist <- gene_anno$gene
#rownames(cts) == "Cd69"
index <- which(rownames(cts) %in% genelist)
#得到对应的文本标签;
labs <- rownames(cts)[index]
lab2 = rowAnnotation(foo = anno_mark(at = index,
                                     labels = labs,
                                     labels_gp = gpar(fontsize = 12),
                                     lines_gp = gpar()))


###作图

pdf("E:/heatmap1.pdf",height = 15,width = 12)
```

```r
ht <- Heatmap(
  cts,
  name = "Expression",
  top_annotation = ha,
  show_column_names = F,
  show_row_names = F ,
  cluster_columns  = F,
  cluster_rows = F ,
  row_names_gp = gpar(fontsize = 8),
  right_annotation = lab2,
  row_names_side = "left",
  col = f1,
  use_raster = T,
  raster_quality = 2,
  column_split = new_cluster

)

draw(ht,adjust_annotation_extension = TRUE)
dev.off()
```

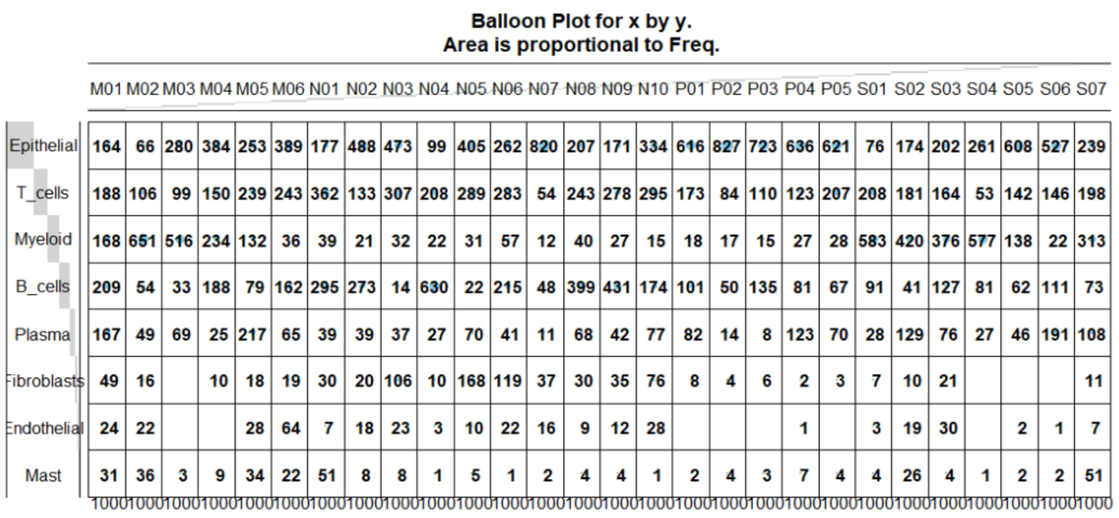
```
## png
##   2
```



```
########################balloonplot
as.data.frame(table(seu_obj_anno_10k$main_cell_type))
```

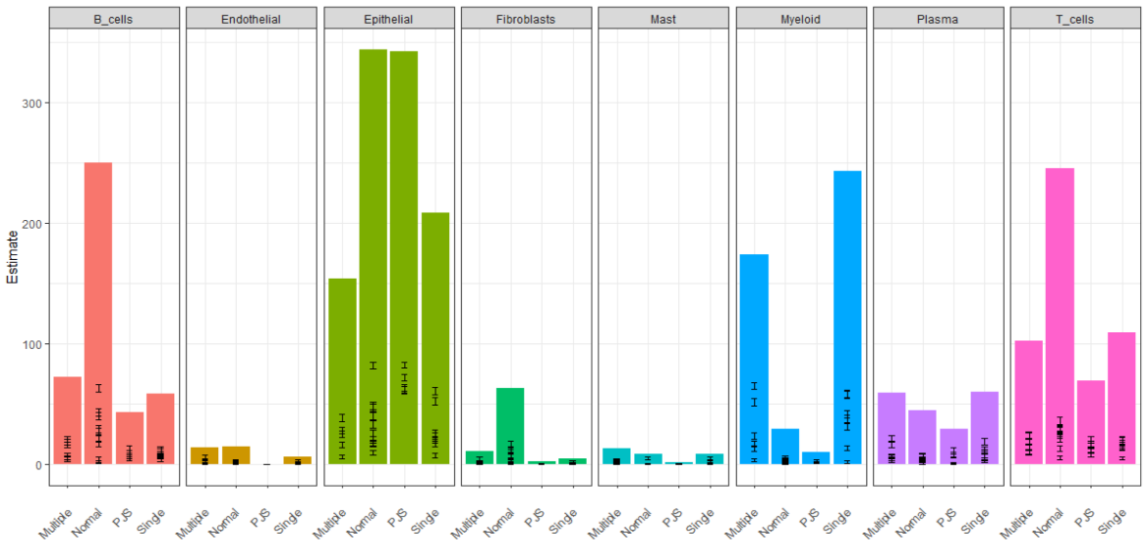
```
##           Var1  Freq
## 1  Epithelial 10482
## 2      T_cells  5266
## 3      Myeloid  4567
## 4      B_cells  4246
## 5       Plasma  1945
## 6 Fibroblasts   815
## 7 Endothelial   349
## 8         Mast   330
```

```
as.data.frame(table(seu_obj_anno_10k$orig.ident))

library(gplots)

tab.1=table(seu_obj_anno_10k$orig.ident,seu_obj_anno_10k$main_cell_typ
e)
balloonplot(tab.1)
```

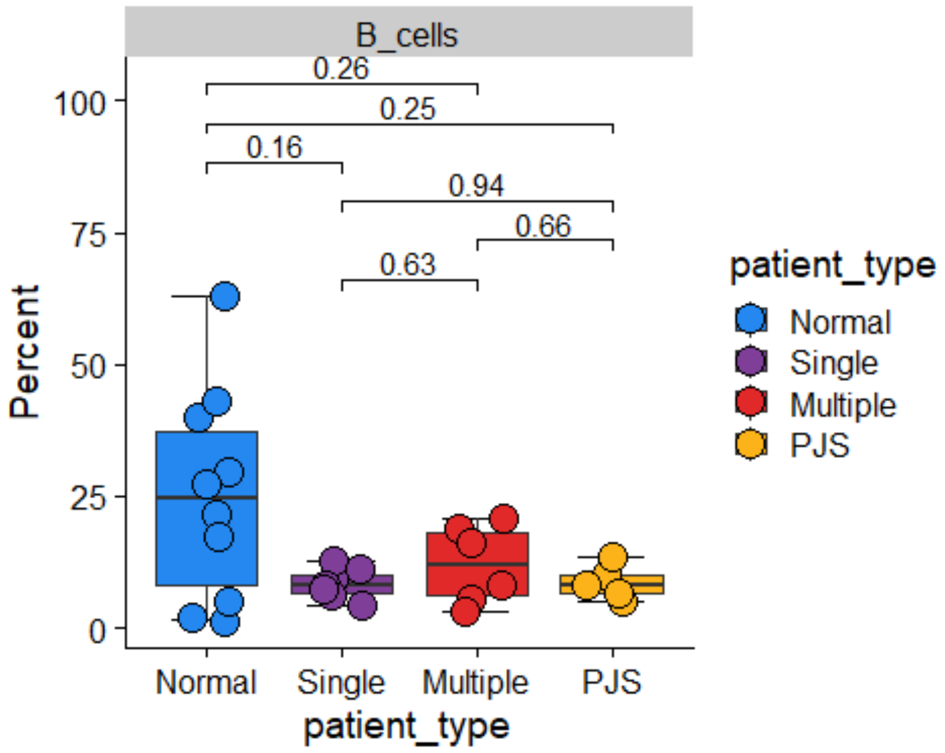


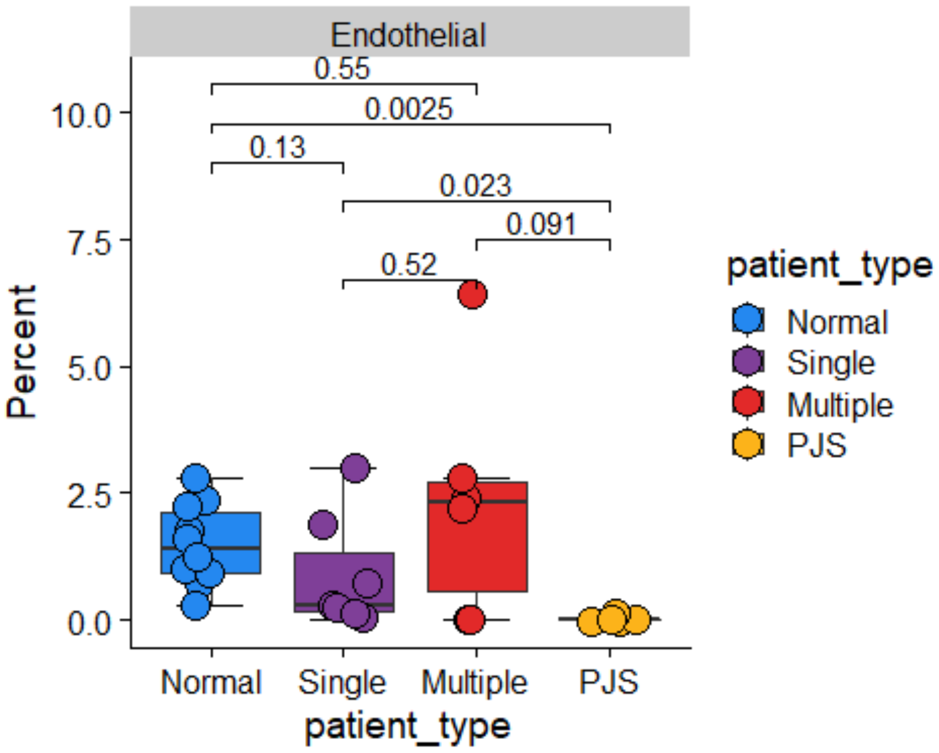
	M01	M02	M03	M04	M05	M06	N01	N02	N03	N04	N05	N06	N07	N08	N09	N10	P01	P02	P03	P04	P05	S01	S02	S03	S04	S05	S06	S07
Epithelial	164	66	280	384	253	389	177	488	473	99	405	262	820	207	171	334	616	827	723	636	621	76	174	202	261	608	527	239
T_cells	188	106	99	150	239	243	362	133	307	208	289	283	54	243	278	295	173	84	110	123	207	208	181	164	53	142	146	198
Myeloid	168	651	516	234	132	36	39	21	32	22	31	57	12	40	27	15	18	17	15	27	28	583	420	376	577	138	22	313
B_cells	209	54	33	188	79	162	295	273	14	630	22	215	48	399	431	174	101	50	135	81	67	91	41	127	81	62	111	73
Plasma	167	49	69	25	217	65	39	39	37	27	70	41	11	68	42	77	82	14	8	123	70	28	129	76	27	46	191	108
Fibroblasts	49	16		10	18	19	30	20	106	10	168	119	37	30	35	76	8	4	6	2	3	7	10	21				11
Endothelial	24	22			28	64	7	18	23	3	10	22	16	9	12	28				1		3	19	30		2	1	7
Mast	31	36	3	9	34	22	51	8	8	1	5	1	2	4	4	1	2	4	3	7	4	4	26	4	1	2	2	51
	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

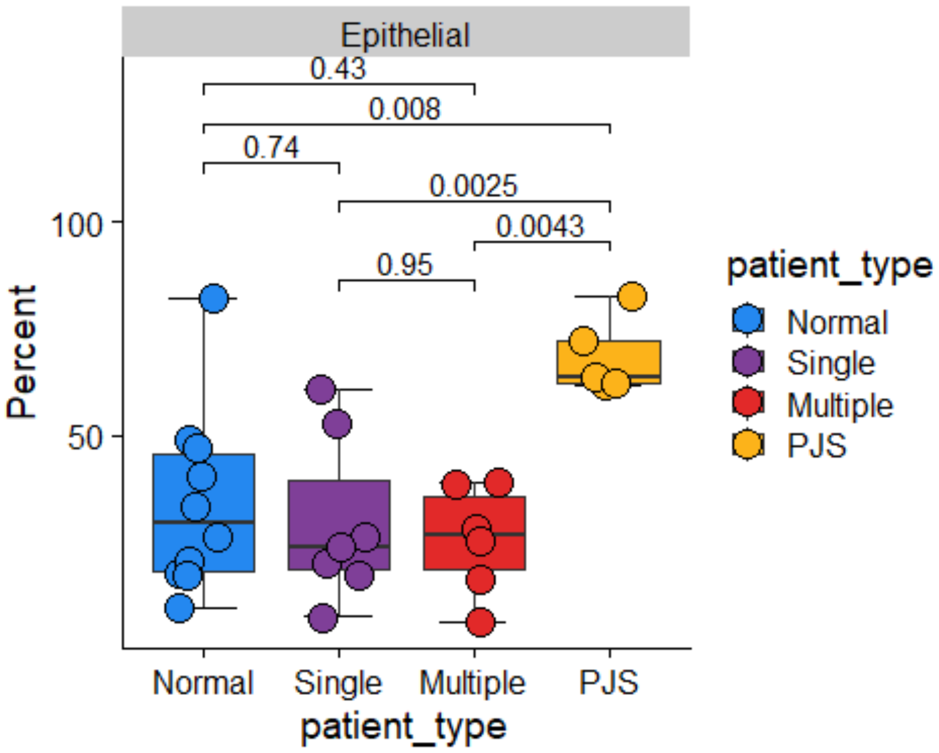
```
########################barplot
library(ggrepel)
require(qdapTools)

require(REdaS)

meta.temp <- seu_obj_anno_10k@meta.data[,c("main_cell_type", "orig.iden
t")]

#按celltype 计算百分比存入 prop
prop.table <- list()
for(i in 1:length(unique(meta.temp$orig.ident))){
  vec.temp <- meta.temp[meta.temp$orig.ident==unique(meta.temp$orig.ide
nt)[i],"main_cell_type"]
  # Convert to counts and calculate 95% CI
  table.temp <- freqCI(vec.temp, level = c(.95))
  prop.table[[i]] <- print(table.temp, percent = TRUE, digits = 3)
  #
}

# Name list
names(prop.table) <- unique(meta.temp$orig.ident)

# Convert to data frame
tab.2 <- as.data.frame.array(do.call(rbind, prop.table))

# Add orig.ident column
b <- c()
a <- c()
for(i in names(prop.table)){
  a <- rep(i,nrow(prop.table[[i]]))
  b <- c(b,a)
}
tab.2$orig.ident <- b

# Add common celltype names
aa <- gsub("\\.[0-9]+","",row.names(tab.2))
tab.2$celltype <- aa

# Resort factor orig.ident (celltype*orig.ident)，需要查看 table2 按 orig
的顺序
tab.2$patient_type <- c(rep("Single",56),rep("Multiple",48),rep("PJS",4
0),rep("Normal",80))

# Rename percentile columns
colnames(tab.2)[1] <- "lower"
colnames(tab.2)[3] <- "upper"

# Plots
p<- ggplot(tab.2, aes(x=patient_type, y=Estimate, group=celltype)) +
  geom_line(aes(color=celltype))+
  geom_point(aes(color=celltype)) + facet_grid(cols =  vars(celltype))
+
  theme(axis.text.x = element_text(angle = 45, hjust=1, vjust=0.5), leg
end.position="bottom") +
  xlab("") +
  geom_errorbar(aes(ymin=lower, ymax=upper), width=.2,position=position
_dodge(0.05))

p1<- ggplot(tab.2, aes(x=patient_type, y=Estimate, group=celltype)) +
  geom_bar(stat = "identity", aes(fill=celltype)) + facet_grid(cols =
vars(celltype)) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust=1, vjust=0.5), leg
end.position= "none") +
  xlab("") +
  geom_errorbar(aes(ymin=lower, ymax=upper), width=.2,position=position
_dodge(0.05))

print(p1)
```

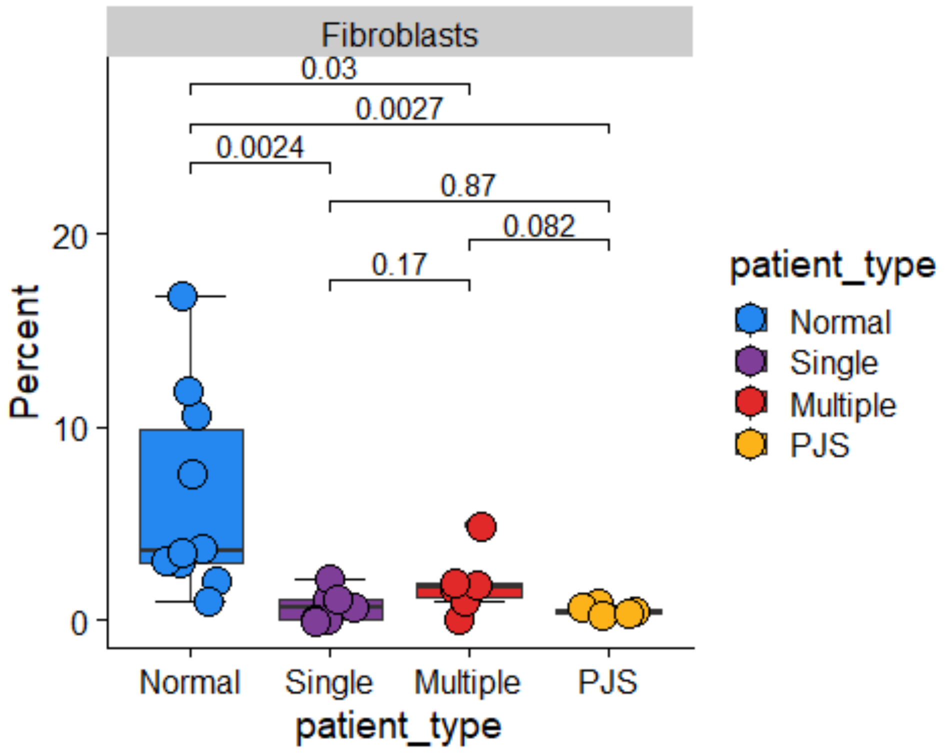


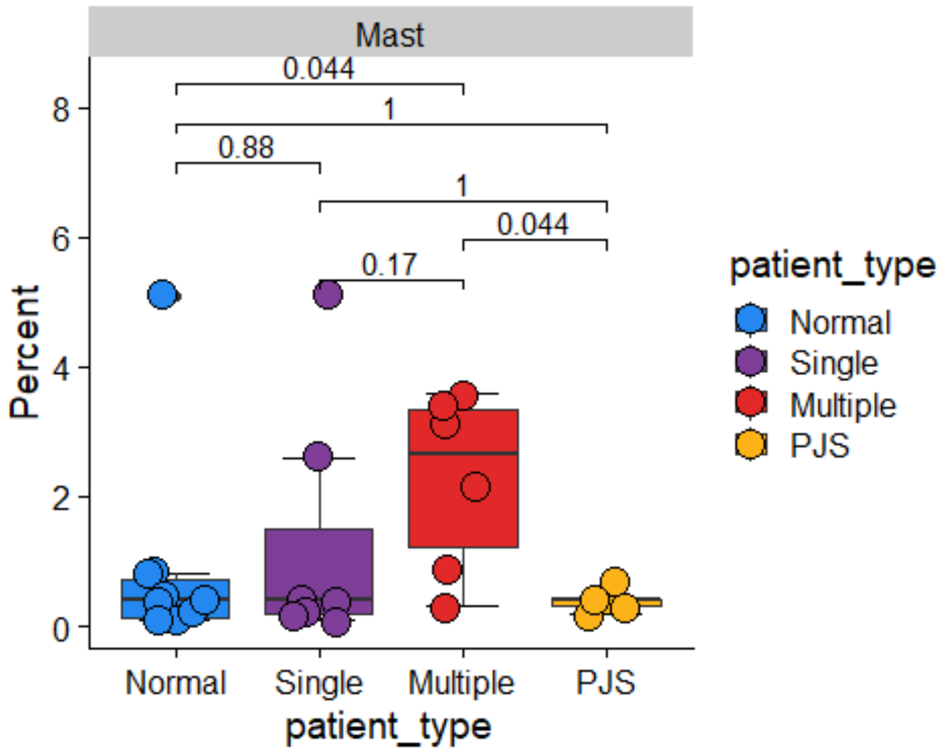
```
############################boxplot cell count

setwd("E:/")
plot_list = list()

#counting P-value
my_comparisons=list(c("Single","Multiple"),c("Multiple","PJS"),c("PJS",
"Single"),c("Normal","Single"),c("Normal", "PJS"),c("Normal","Multiple
"))

#create boxplot's table
tab.3 <- as.data.frame(table(seu_obj_anno_10k$main_cell_type,seu_obj_an
no_10k$orig.ident))
colnames(tab.3) <- c("celltype","sample","count")
#add pateient_type coloumn
tab.3$patient_type <- c(rep("Multiple",48),rep("Normal",80),rep("PJS",4
0),rep("Single",56))

# for (i in levels(as.factor(tab.3$celltype))) {
  p2 <- ggplot(tab.3 %>% filter(celltype == i ),aes(x=patient_type,y=co
unt,fill=patient_type))+
    stat_boxplot(geom = "errorbar",width=0.5)+
    geom_boxplot()+
    facet_wrap(~celltype,scales = "free")+
    scale_x_discrete(limits=c("Normal","Single","Multiple","PJS"))+
    scale_fill_manual(values =c(Normal="#2488F0",Single = "#7F3F98",Mul
tiple = "#E22929",PJS="#FCB31A"))+
    theme(panel.background = element_blank(),axis.line = element_line
())+
    geom_jitter(aes(fill=patient_type),width =0.2,shape = 21,size=5)+
    ylab("Cells Count")+
    stat_compare_means(comparisons=my_comparisons)

for (i in levels(as.factor(tab.2$celltype))) {
  p3 <- ggplot(tab.2 %>% filter(celltype == i),aes(x=patient_type,y=Est
imate,fill=patient_type))+
    stat_boxplot(geom = "errorbar",width=0.5)+
    geom_boxplot()+
    facet_wrap(~celltype,scales = "free")+
    scale_x_discrete(limits=c("Normal","Single","Multiple","PJS"))+
    scale_fill_manual(values =c(Normal="#2488F0",Single = "#7F3F98",Mul
tiple = "#E22929",PJS="#FCB31A"))+
    theme(panel.background = element_blank(),axis.line = element_line
())+
    geom_jitter(aes(fill=patient_type),width =0.2,shape = 21,size=5)+
    ylab("Percent")+
    stat_compare_means(comparisons =my_comparisons)

  #p4=p2+p3
  plot_list[[i]] = p3
  print(p3)
}
```

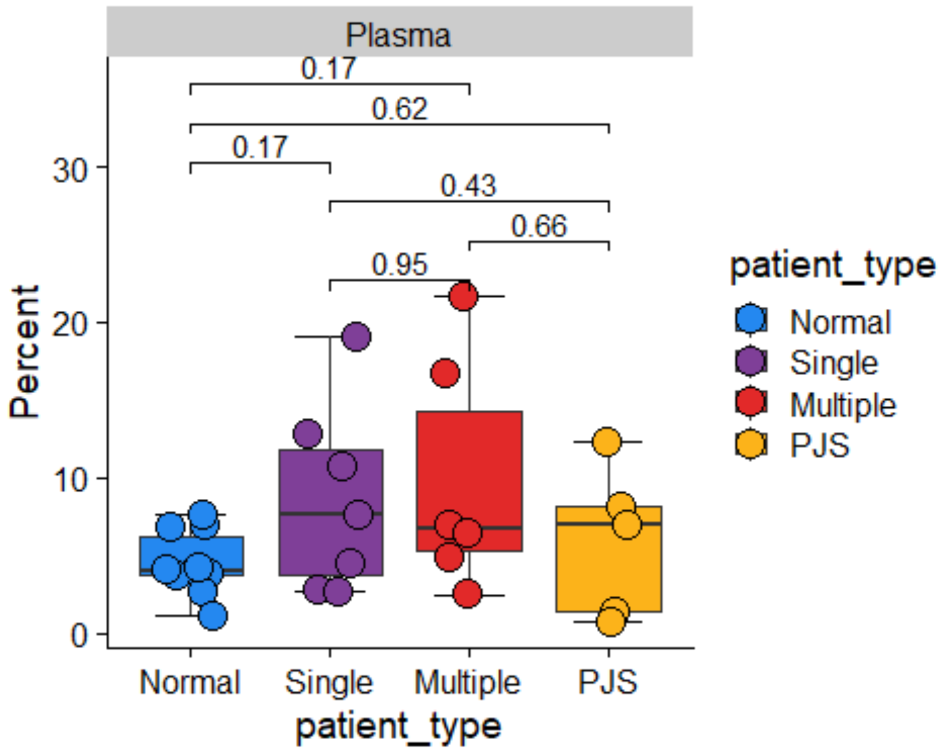

B_cells
0.26
0.25
0.16
0.94
0.66
0.63
Percent
100
75
50
25
0
Normal
Single
Multiple
PJS
patient_type
patient_type
Normal
Single
Multiple
PJS

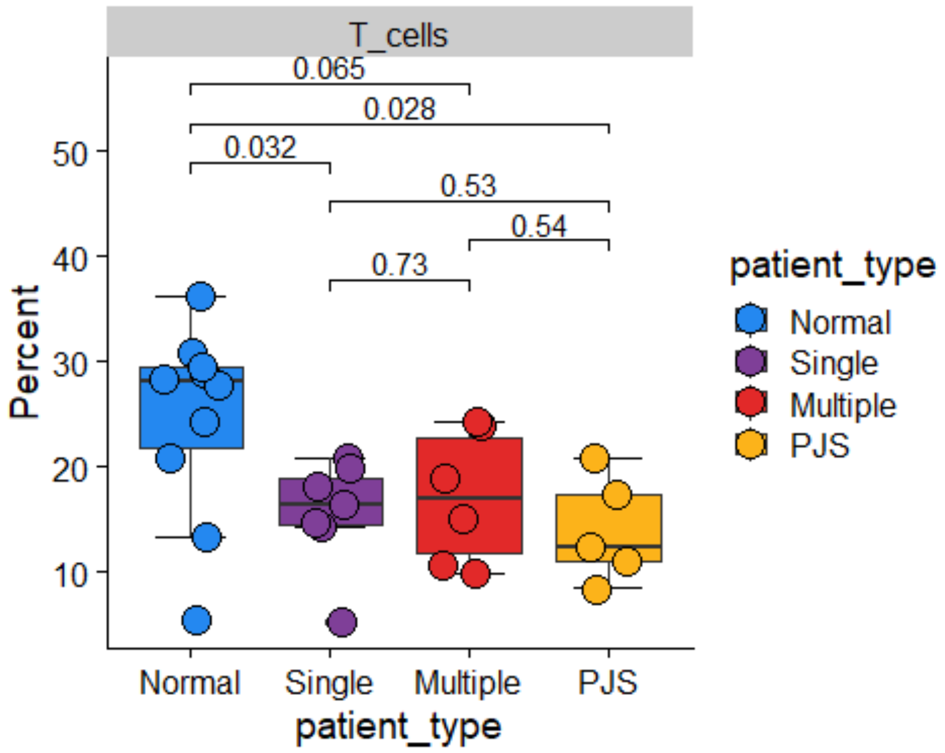

Endothelial
0.55
0.0025
0.13
0.023
0.091
0.52
10.0
7.5
5.0
2.5
0.0
Percent
Normal
Single
Multiple
PJS
patient_type
patient_type
Normal
Single
Multiple
PJS


Epithelial
0.43
0.008
0.74
0.0025
0.0043
0.95
100
50
Percent
Normal
Single
Multiple
PJS
patient_type
patient_type
Normal
Single
Multiple
PJS


Fibroblasts
0.03
0.0027
0.0024
0.87
0.082
0.17
Percent
0
10
20
Normal
Single
Multiple
PJS
patient_type
patient_type
Normal
Single
Multiple
PJS


Mast
0.044
1
0.88
1
0.044
0.17
Percent
0
2
4
6
8
Normal
Single
Multiple
PJS
patient_type
patient_type
Normal
Single
Multiple
PJS



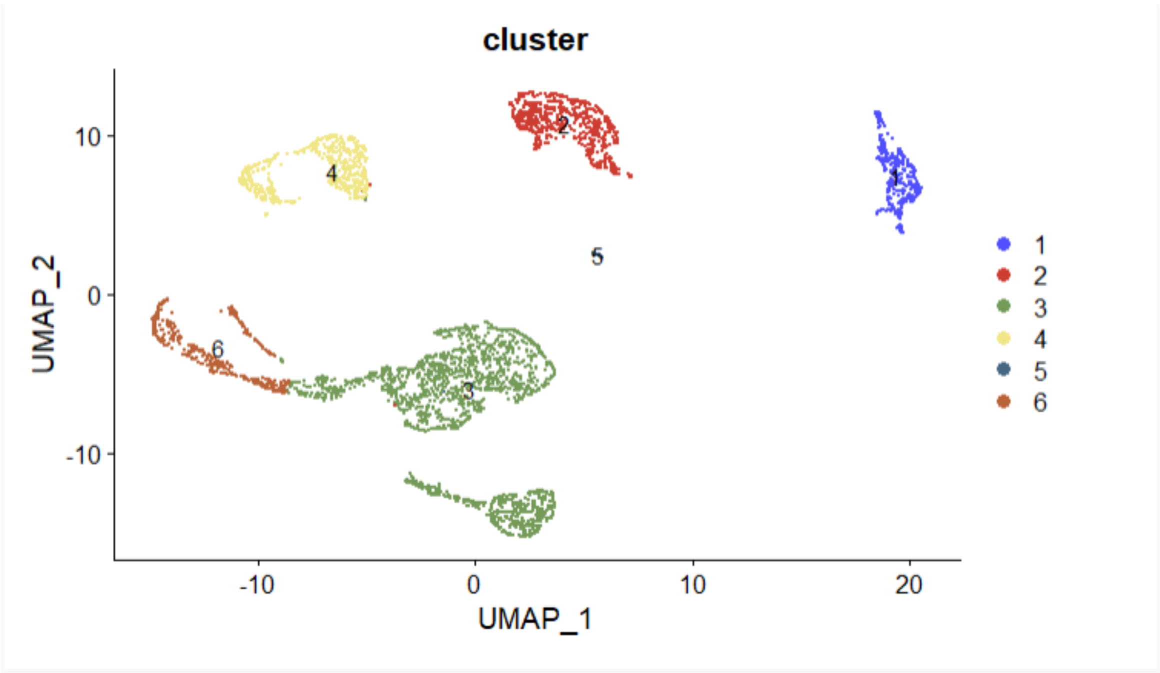


```
# Save plots to pdf. Makes a separate file for each plot.
for (i in levels(as.factor(tab.2$celltype))) {
  file_name = paste("Boxplot_", i, ".pdf", sep="")
```

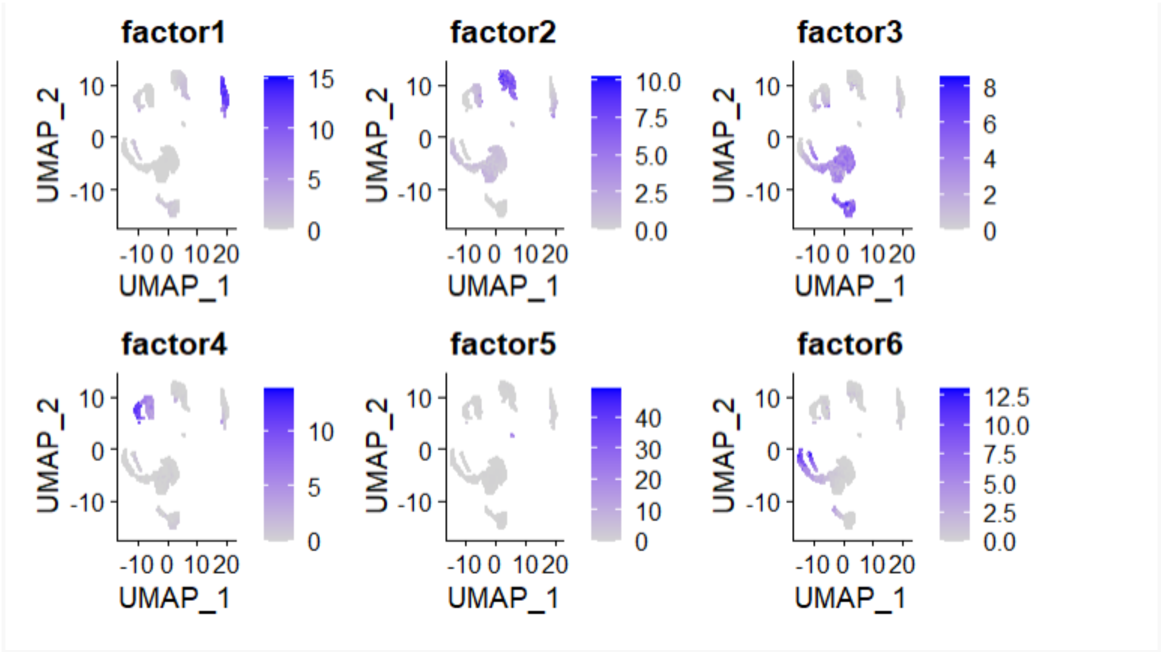
```r
  pdf(file = file_name,width = 8, height = 9)
  print(plot_list[[i]])
  dev.off()
}

# create pdf where each page is a separate plot.
pdf("Celltype_Boxplots.pdf",width = 8,height = 9)
for (i in levels(as.factor(tab.2$celltype))) {
  print(plot_list[[i]])
}

dev.off()


#NMF and Pagoda2 analyst

### load file

pbmc_final <- readRDS("E:/MangeXU/PBMC_Presentation/pbmc_final.RDS")


###首先对单核单细胞亚群矩阵进行归一化

pbmc_final=CreateSeuratObject(

  counts = pbmc_final@assays$RNA@counts,

  meta.data = pbmc_final@meta.data

)

pbmc_final = NormalizeData(pbmc_final) %>% FindVariableFeatures() %>% S
caleData(do.center = F)



###非负矩阵分解分析

suppressPackageStartupMessages(library(NMF))

vm <- pbmc_final@assays$RNA@scale.data

###保存文件到服务器上运行

#saveRDS(vm, file = "../pbmc_final/vm.RDS")



#vm <- readRDS("/home/mgxu/vm.RDS")
```

```
### 参数rank=6，是期望的细胞亚群数量

# 默认交替最小二乘法(Alternating Least Squares(ALS))——snmf/r

res <- nmf(vm,rank=6,method = "snmf/r",seed = 'nndsvd')

#runtime (res)

#save.image(file="pbmc_final_NMF.RData")


###读取服务器运行完的数据

#load("../pbmc_final/pbmc_final_NMF.RData")

###查看得到的 NMFfit 的对象

head(basis(res))

               [,1]        [,2]       [,3]        [,4]         [,5]
    [,6]

HES4    0.008345689 0.145699660 0.00000000 0.000000000 0.0035335303 0.0
00000000

ISG15   0.002073496 0.194012117 0.06801100 0.012683413 0.0000000000 0.0
55989413

TNFRSF4 0.000000000 0.007344792 0.09429587 0.000000000 0.0000000000 0.0
06084677

ATAD3C  0.005264382 0.000000000 0.01674113 0.001234889 0.0000000000 0.0
07884858

RER1    0.053077950 0.088103373 0.07583872 0.027842831 0.0007800666 0.0
62308748

LRRC47  0.042234113 0.019689802 0.04892135 0.024810879 0.0034610191 0.0
36841062


###前面的非负矩阵分解相当于是替代了 PCA 操作，将结果导入 seurat 对象里面

pbmc_final <- RunPCA(pbmc_final)

pbmc_final@reductions$nmf <- pbmc_final@reductions$pca

pbmc_final@reductions$nmf@cell.embeddings <- t(coef(res) )

pbmc_final@reductions$nmf@feature.loadings <- basis(res)
```

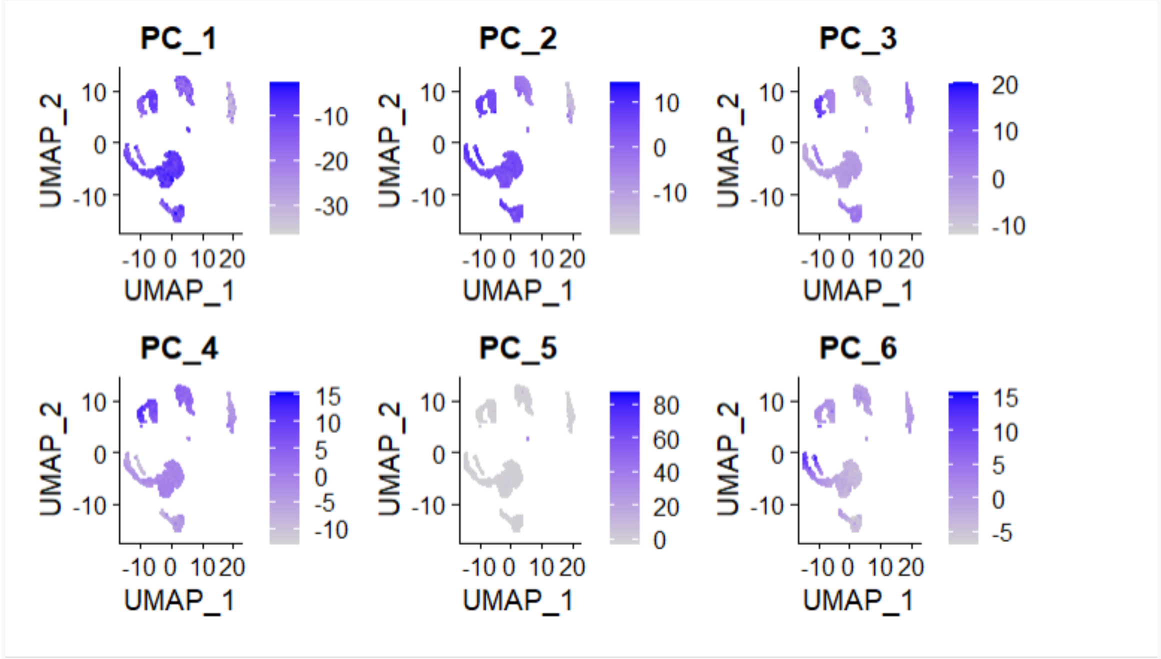
```
###使用 NMF 运行的结果进行降维和聚类, dim 最大值为rank 的设置值

set.seed(219)

pbmc_final.nmf <- RunUMAP(pbmc_final,reduction = "nmf",dims = 1:6) %>%
FindNeighbors(reduction = "nmf",dims = 1:6) %>% FindClusters(resolution
 = 0.2)

pbmc_final.nmf$cluster <- apply(NMF::coef(res)[1:6,],2,which.max)

table(Idents(pbmc_final.nmf) ,pbmc_final.nmf$cluster)


       1    2    3    4    5    6

  0    0    1 1171    0    0    0

  1    0  681    0    0    2    0

  2  369    0    0    0    0    0

  3    0    0    4  352    0    1

  4    0    0  350    0    1    0

  5    0    0  133    0    1  180

  6    0    0    0  192    0    0

  7    0    0   97    0    0   88

  8    0    0    0    0    0  157

  9    0    0    0    0   28    0


### 结果可视化

DimPlot(pbmc_final.nmf, label = T,group.by = "cluster") + ggsci::scale_
color_igv()
```



```
###提取每个细胞亚群的权重排名靠前的特征基因
fs <- extractFeatures(res,10L)
fs <- lapply(fs,function(x)rownames(res)[x])
fs <- do.call("rbind", fs)
DT::datatable(t(fs))

### 查看细胞因子上的荷载
tmp <- data.frame(t(coef(res)), check.names = F)
colnames(tmp) <- paste0("factor", 1:6)
pbmc_final.nmf <- AddMetaData(pbmc_final.nmf, metadata = tmp)

FeaturePlot(pbmc_final.nmf, features = paste0("factor", 1:6), ncol = 3)
```



```
### 查看细胞主成分上的荷载
FeaturePlot(pbmc_final.nmf, features = paste0("PC_", 1:6), ncol = 3)
```

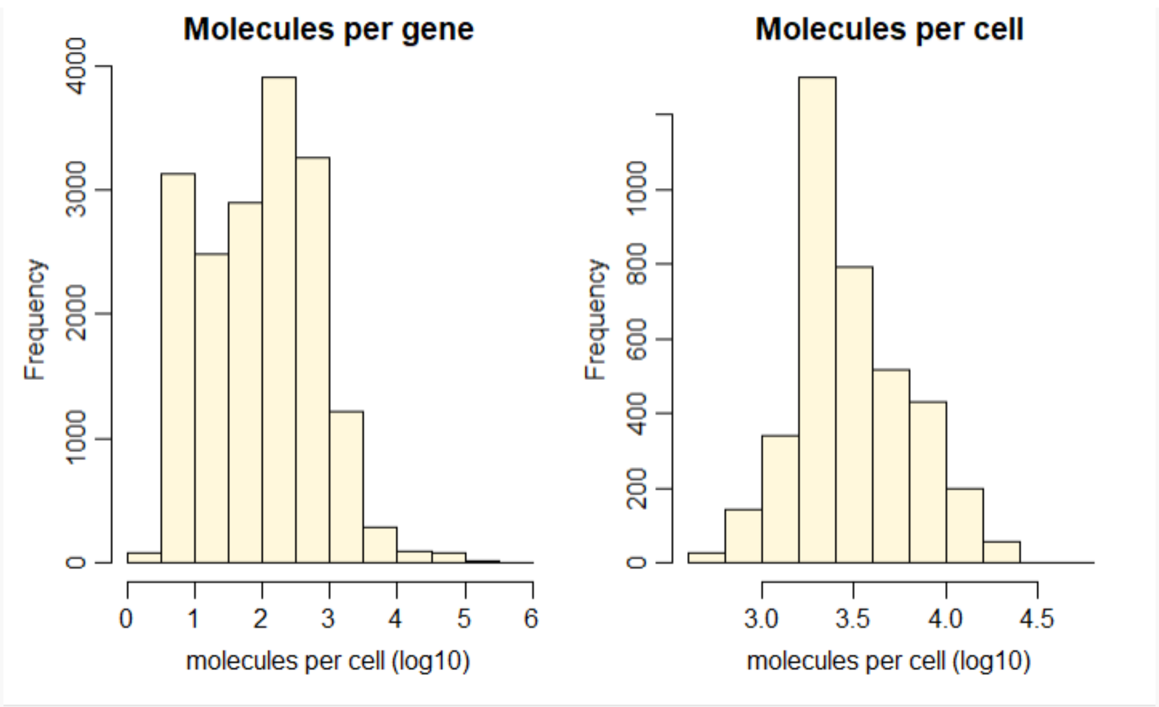


Pagoda2

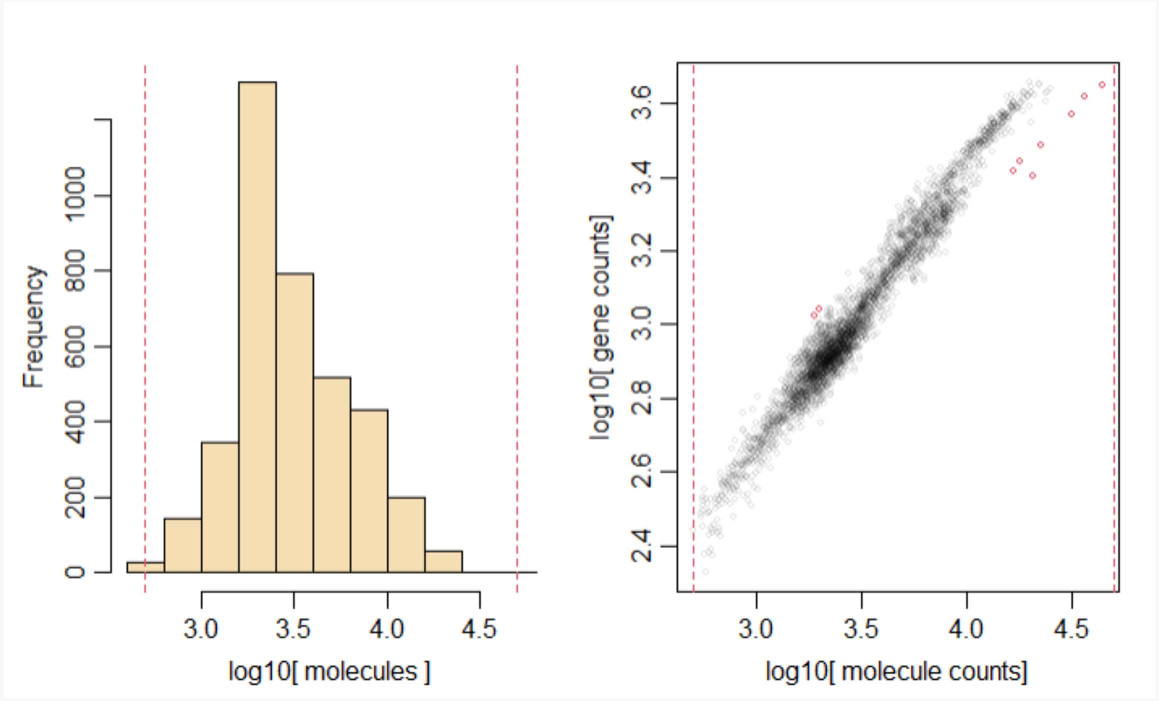
```
### 质控，pagoda2 只需要一个表达量矩阵
```

```
cm=pbmc_final@assays$RNA@counts

dim(cm)

[1] 17424  3808

cm[1:3,1:3]

3 x 3 sparse Matrix of class "dgCMatrix"

           AAACCCAAGGAGAGTA-1 AAACGCTTCAGCCCAG-1 AAAGAACAGACGACTG-1

AL627309.1                  .                  .                  .

AL669831.5                  .                  .                  .

LINC00115                   .                  .                  .


str(cm)

Formal class 'dgCMatrix' [package "Matrix"] with 6 slots

  ..@ i       : int [1:4701615] 21 26 28 29 30 39 45 48 49 54 ...

  ..@ p       : int [1:3809] 0 2618 4423 5982 7207 9035 11081 12668 160
88 19835 ...

  ..@ Dim     : int [1:2] 17424 3808

  ..@ Dimnames:List of 2

  .. ..$ : chr [1:17424] "AL627309.1" "AL669831.5" "LINC00115" "FAM41C"
 ...

  .. ..$ : chr [1:3808] "AAACCCAAGGAGAGTA-1" "AAACGCTTCAGCCCAG-1" "AAAG
AACAGACGACTG-1" "AAAGAACCAATGGCAG-1" ...

  ..@ x       : num [1:4701615] 1 1 1 1 2 1 1 2 2 1 ...

  ..@ factors : list()


dta<-as.matrix(GetAssayData(pbmc_final,slot = "counts"))


par(mfrow=c(1,2), mar = c(3.5,3.5,2.0,0.5), mgp = c(2,0.65,0), cex = 1.
0)
```

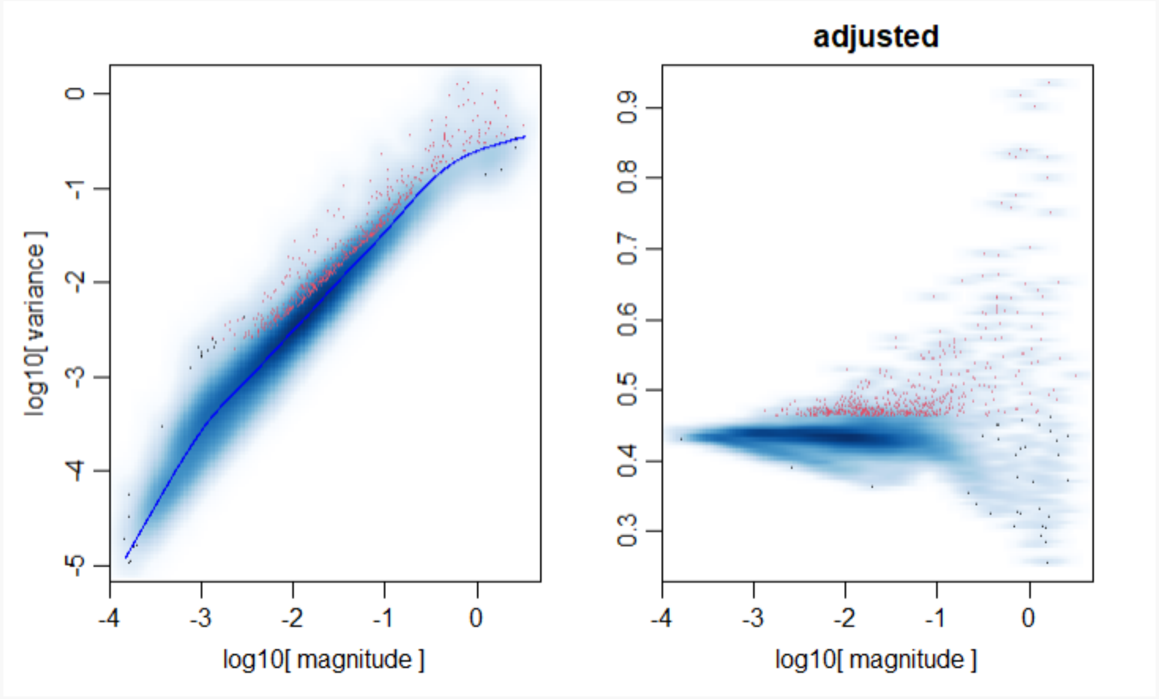
```
hist(log10(rowSums(dta)+1),main='Molecules per gene',xlab='molecules pe
r cell (log10)',col='cornsilk')

hist(log10(colSums(dta)+1),main='Molecules per cell',xlab='molecules pe
r cell (log10)',col='cornsilk')


counts <- gene.vs.molecule.cell.filter(cm, min.cell.size=500)


dta <- dta[rowSums(dta)>=10,]

dim(dta)

[1] 14219  3808


###构建对象

rownames(dta) <- make.unique(rownames(dta))

r <- Pagoda2$new(dta,log.scale=TRUE, n.cores=2)


### 一切的输入数据，都是 dta 这样纯粹的表达量矩阵

r <- Pagoda2$new(dta,log.scale=TRUE, n.cores=2)


### 对表达量差异很大的基因对下游分析所占比重进行调整

r$adjustVariance(plot=T,gam.k=10)


###PCA reduction.

r$calculatePcaReduction(nPcs=50,n.odgenes=3e3)
```



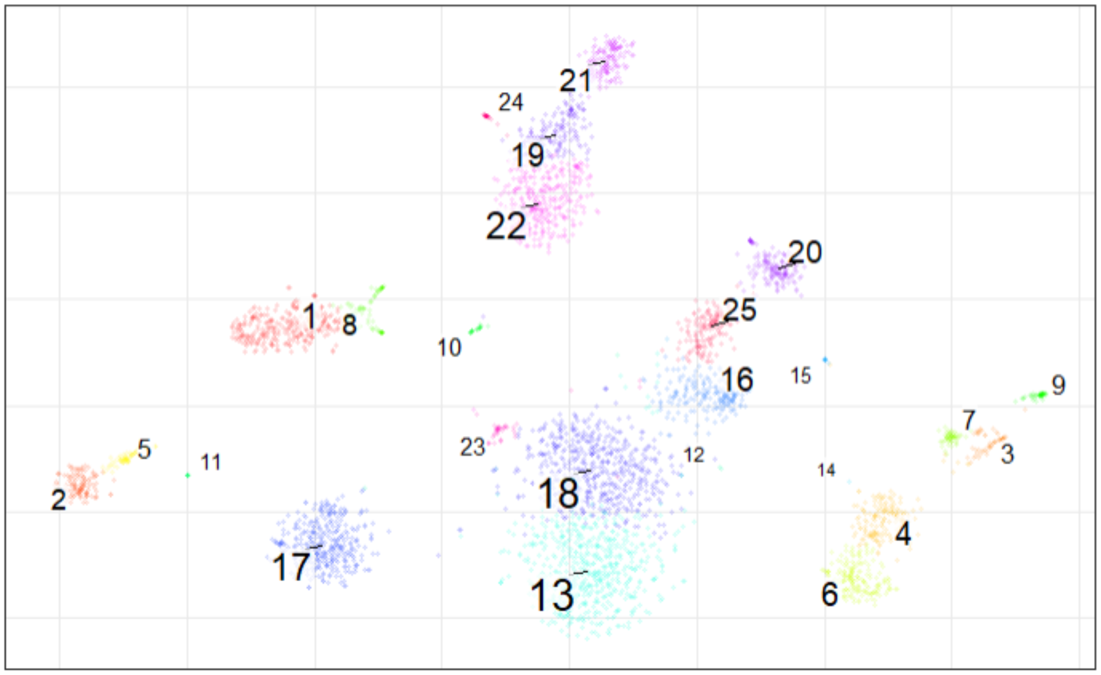
```
### generate a KNN graph
r$makeKnnGraph(k=40,type='PCA',center=T,distance='cosine')
```



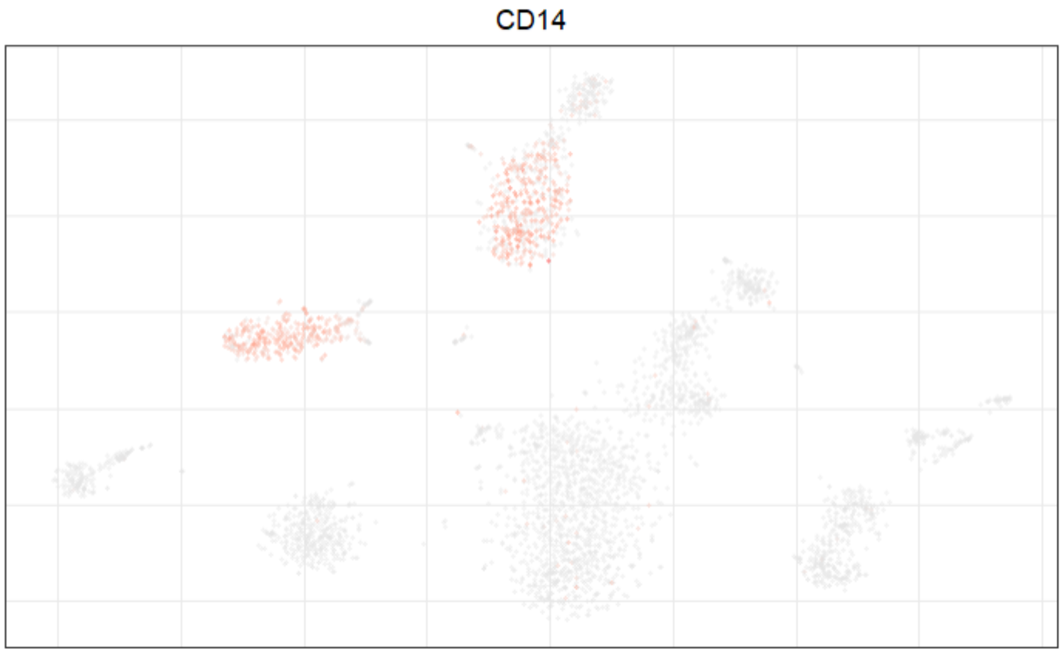
```
###call clusters

r$getKnnClusters(method=infomap.community,type='PCA')


M <- 30

r$getEmbedding(type='PCA', embeddingType = 'largeVis', M=M, perplexity=
30, gamma=1/M)
```



```
###tsne

r$getEmbedding(type='PCA',embeddingType='tSNE',perplexity=50,verbose=F,
n.cores=30)

r$plotEmbedding(type='PCA',embeddingType='tSNE',show.legend=F,min.grou
p.size=1,shuffle.colors=F,mark.cluster.cex=1,alpha=0.1,main='clusters
(tSNE)')
```

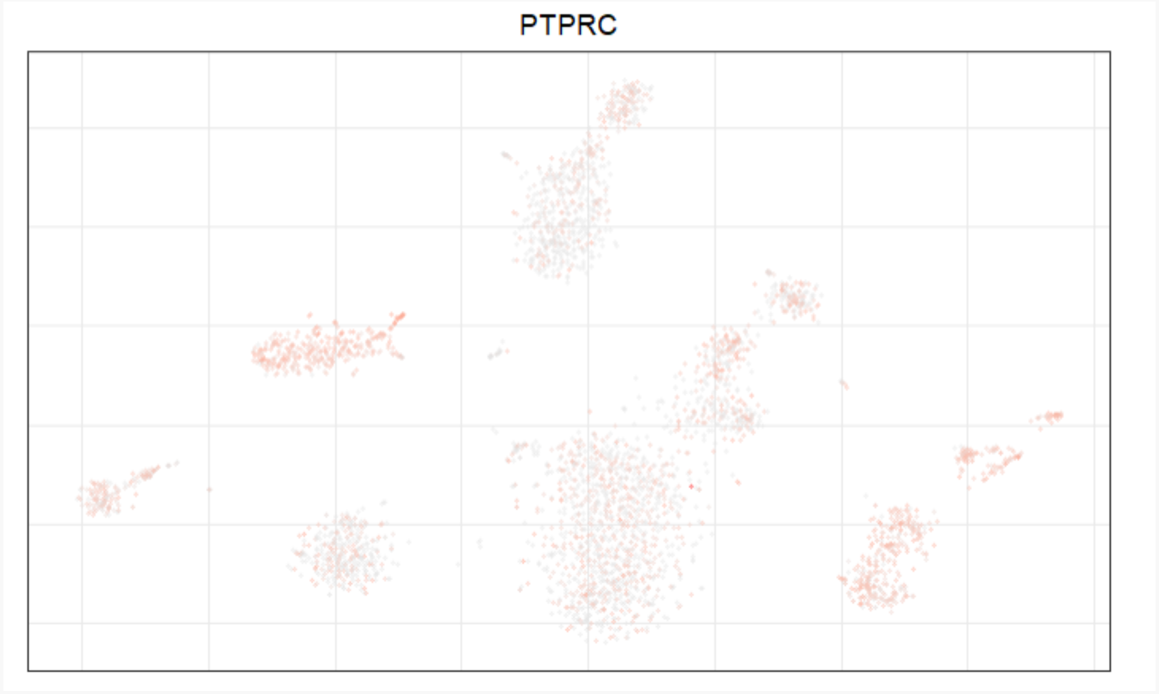


```
gene <-"CD14"

r$plotEmbedding(type='PCA', embeddingType='tSNE', colors=r$counts[,gen
e], shuffle.colors=FALSE,font.size=3, alpha=0.3, title=gene, plot.theme
=theme_bw() + theme(plot.title = element_text(hjust = 0.5)))
```

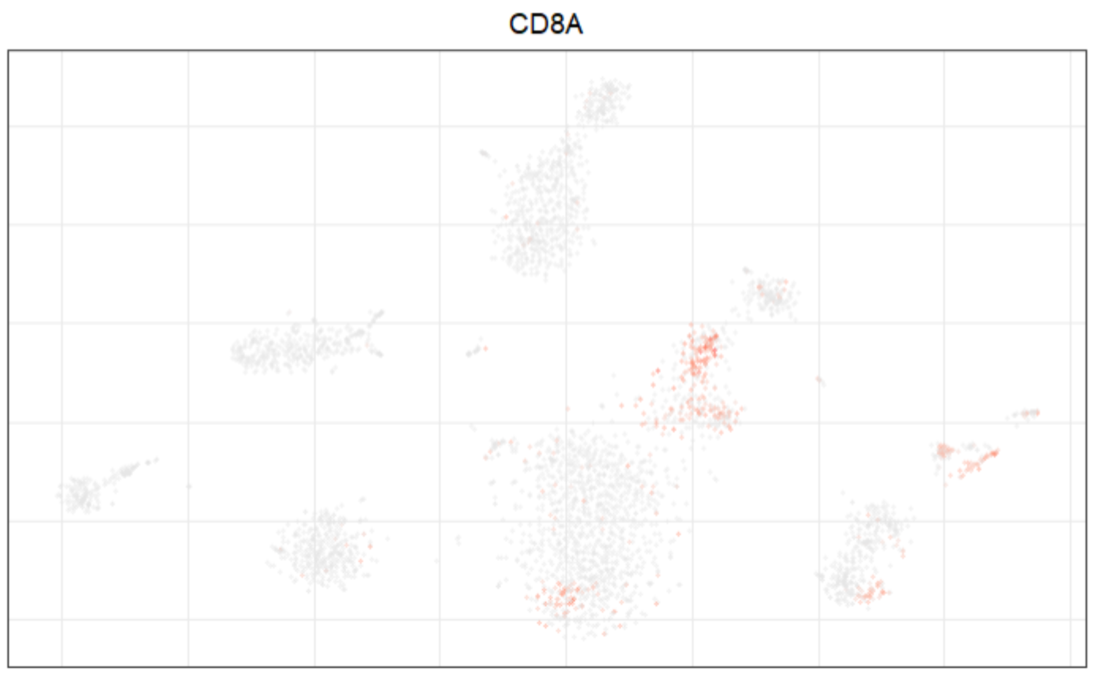


```
gene <-"PTPRC"
```

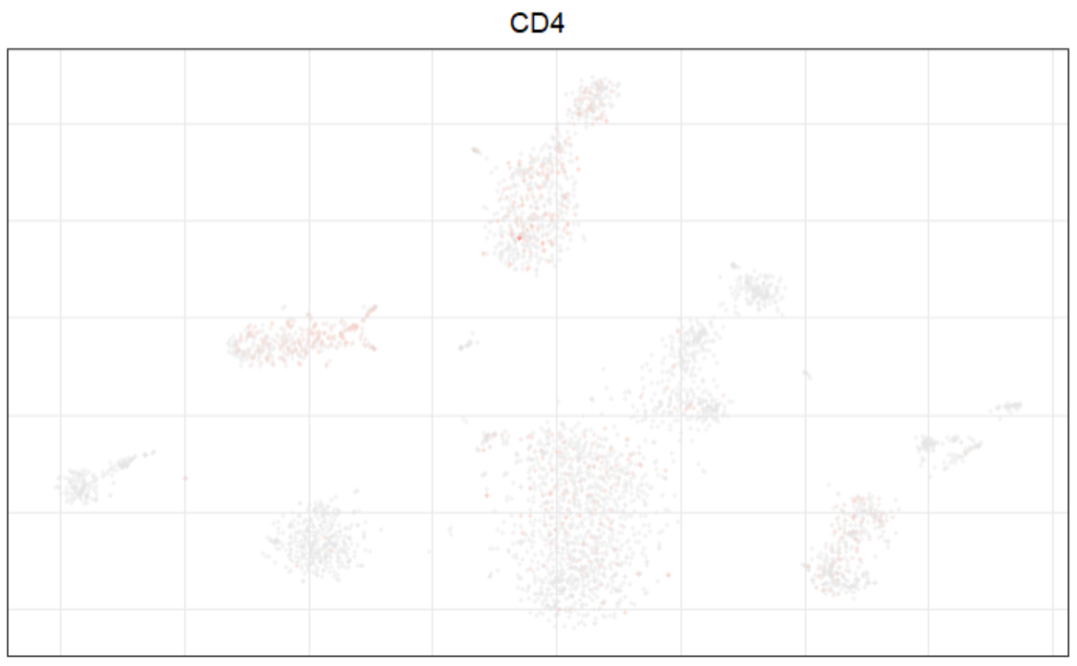
```
r$plotEmbedding(type='PCA', embeddingType='tSNE', colors=r$counts[,gen
e], shuffle.colors=FALSE,font.size=3, alpha=0.3, title=gene, plot.theme
=theme_bw() + theme(plot.title = element_text(hjust = 0.5)))
```



```
gene <-"CD8A"

r$plotEmbedding(type='PCA', embeddingType='tSNE', colors=r$counts[,gen
e], shuffle.colors=FALSE,font.size=3, alpha=0.3, title=gene, plot.theme
=theme_bw() + theme(plot.title = element_text(hjust = 0.5)))
```

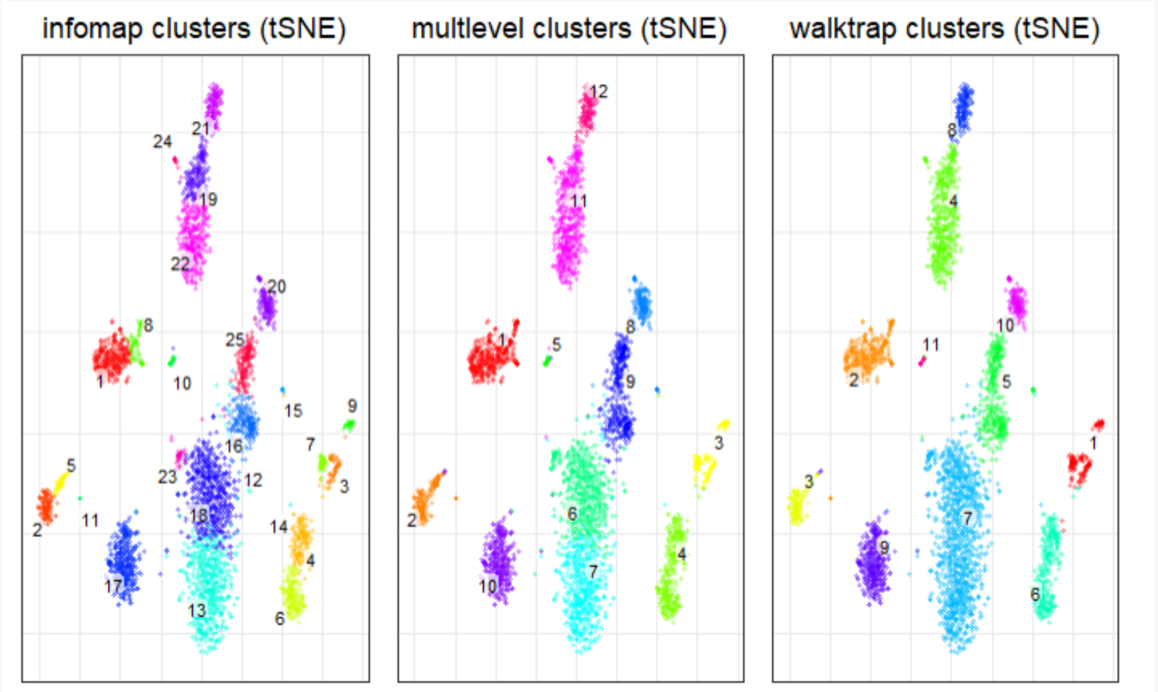


```
gene <-"CD4"

r$plotEmbedding(type='PCA', embeddingType='tSNE', colors=r$counts[,gen
e], shuffle.colors=FALSE,font.size=3, alpha=0.3, title=gene, plot.theme
=theme_bw() + theme(plot.title = element_text(hjust = 0.5)))
```



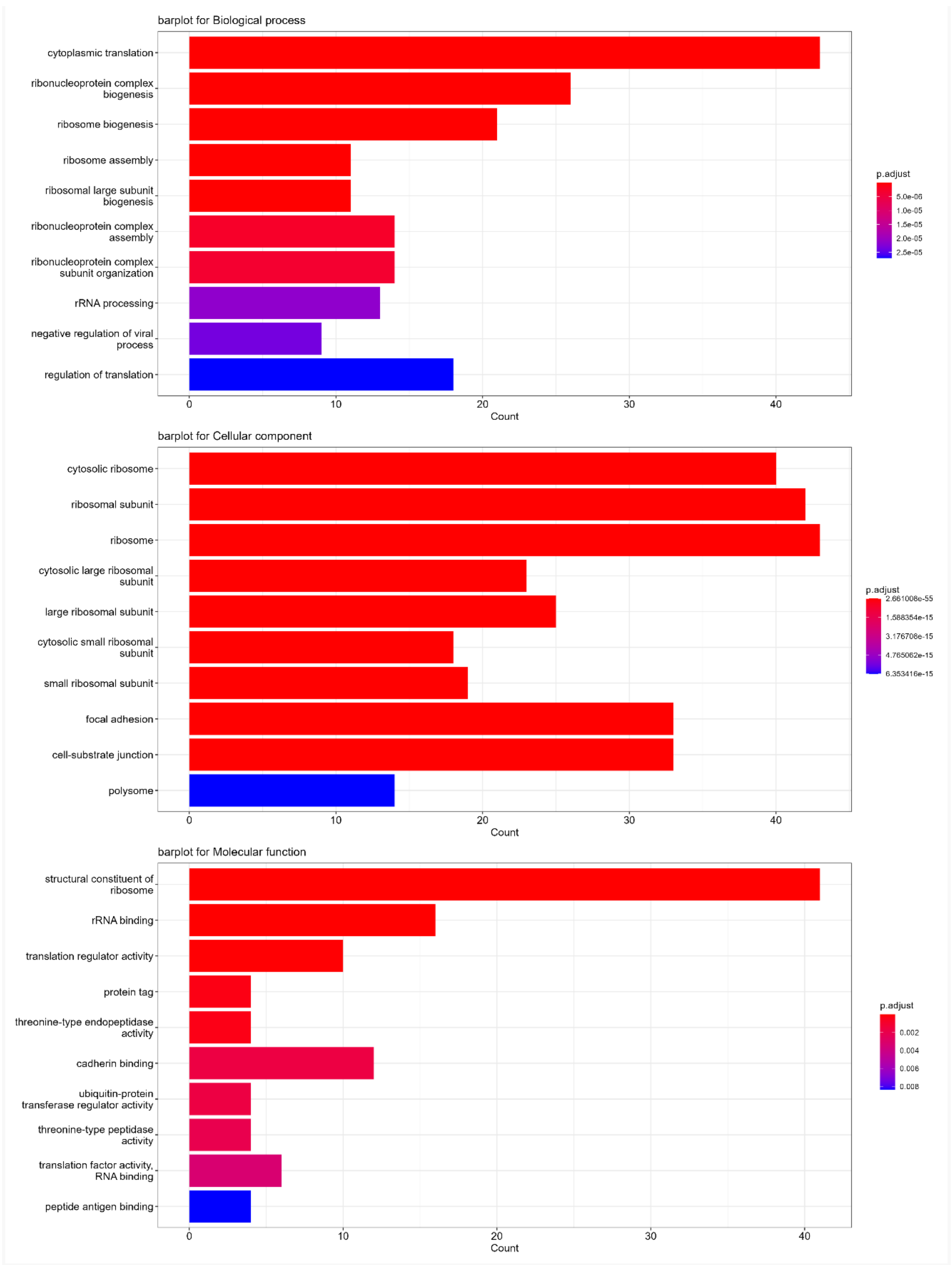
```
r$getKnnClusters(method=multilevel.community, type='PCA', name='multile
vel')
```

```
r$getKnnClusters(method=walktrap.community, type='PCA', name='walktrap
')

str(r$clusters)



plt1 = r$plotEmbedding(type='PCA', embeddingType='tSNE', groups=r$clust
ers$PCA$community, show.legend=FALSE, mark.groups=TRUE, min.cluster.siz
e=1, shuffle.colors=FALSE, font.size=3, alpha=0.3, title='infomap clust
ers (tSNE)', plot.theme=theme_bw() + theme(plot.title = element_text(hj
ust = 0.5)))

plt2 = r$plotEmbedding(type='PCA', embeddingType='tSNE', clusterType='m
ultilevel', show.legend=FALSE, mark.groups=TRUE, min.cluster.size=1, sh
uffle.colors=FALSE, font.size=3, alpha=0.3, title=' multlevel clusters
(tSNE)', plot.theme=theme_bw() + theme(plot.title = element_text(hjust
= 0.5)))

plt3 = r$plotEmbedding(type='PCA', embeddingType='tSNE', clusterType='w
alktrap', show.legend=FALSE, mark.groups=TRUE, min.cluster.size=1, shuf
fle.colors=FALSE, font.size=3, alpha=0.3, title='walktrap clusters (tSN
E)', plot.theme=theme_bw() + theme(plot.title = element_text(hjust = 0.
5)))

gridExtra::grid.arrange(plt1, plt2, plt3, ncol=3)
```

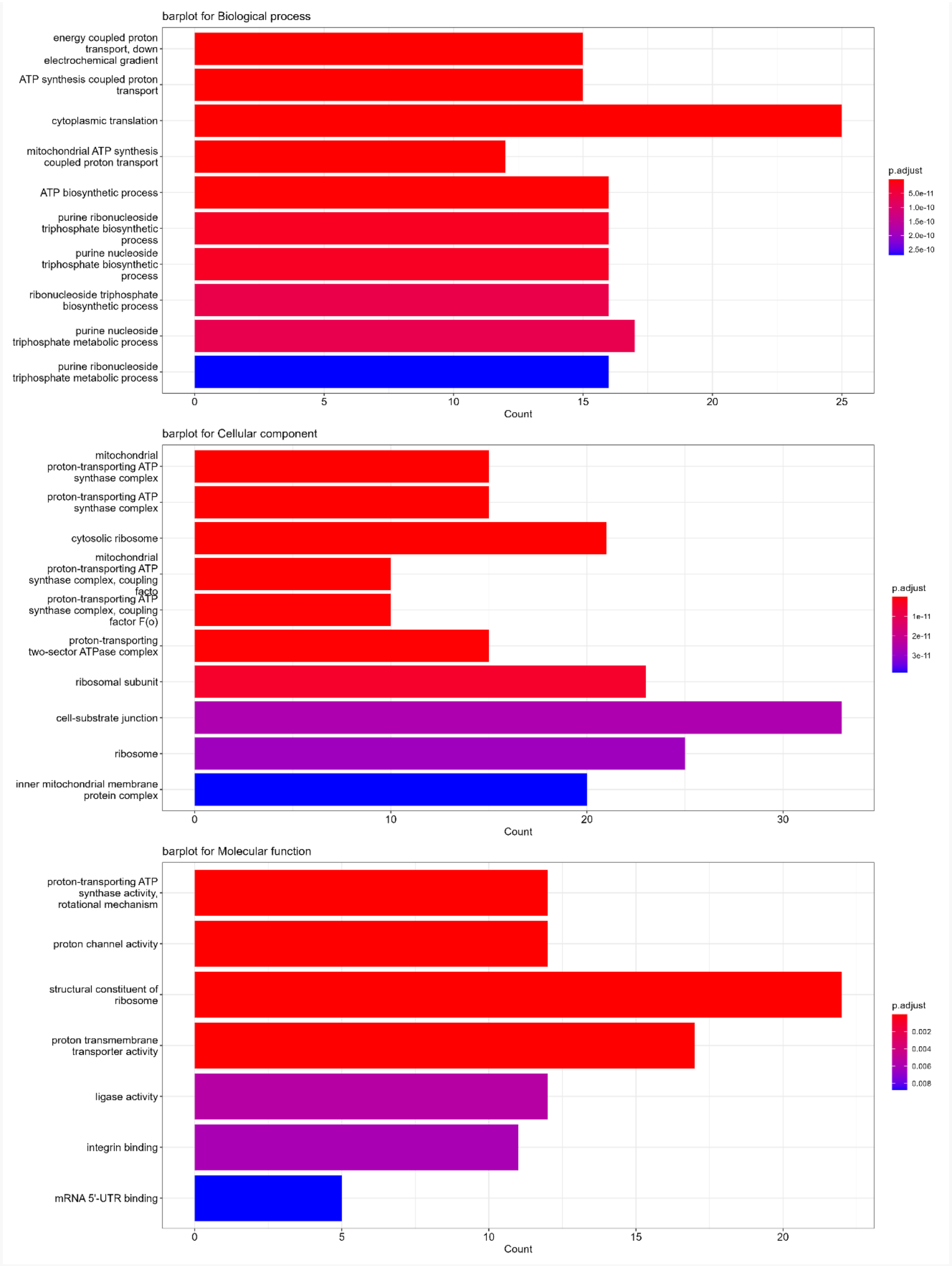


```
###save.image

save.image(file="E:/MangeXU/PBMC_Presentation/2.NMF+Pagoda2/PBMC_NMF+Pa
goda2.RData")
```

```
#lKegg and GO enrich analysis
pbmc_final <- readRDS("E:/MangeXU/PBMC_Presentation/pbmc_final.RDS")

Idents(pbmc_final) <- pbmc_final@meta.data$orig.ident

dge.celltype <- FindMarkers(pbmc_final, ident.1 = 'pbmc1k', ident.2 = '
pbmc3k', group.by = 'orig.ident')
sig_dge.celltype <- subset(dge.celltype, p_val_adj<0.05&abs(avg_log2F
C)>0.25)

gene_up <- subset(sig_dge.celltype, avg_log2FC>0)
gene_down <- subset(sig_dge.celltype, avg_log2FC<0)
gene_diff<- unique(c(gene_up,gene_down ))

#Go Down
ego_ALL <- enrichGO(gene          = row.names(gene_down),
                   OrgDb         = 'org.Hs.eg.db',
                   keyType       = 'SYMBOL',
                   ont           = "ALL",
                   pAdjustMethod = "BH",
                   pvalueCutoff  = 0.01,
                   qvalueCutoff  = 0.05)
ego_all <- data.frame(ego_ALL)
write.csv(ego_all,'E:/MangeXU/PBMC_Presentation/Enrich/enrichGO_down.cs
v')

ego_CC <- enrichGO(gene          = row.names(gene_down),
                   OrgDb         = 'org.Hs.eg.db',
                   keyType       = 'SYMBOL',
                   ont           = "CC",
                   pAdjustMethod = "BH",
                   pvalueCutoff  = 0.01,
                   qvalueCutoff  = 0.05)

ego_MF <- enrichGO(gene          = row.names(gene_down),
                   OrgDb         = 'org.Hs.eg.db',
                   keyType       = 'SYMBOL',
                   ont           = "MF",
                   pAdjustMethod = "BH",
                   pvalueCutoff  = 0.01,
                   qvalueCutoff  = 0.05)

ego_BP <- enrichGO(gene          = row.names(gene_down),
                   OrgDb         = 'org.Hs.eg.db',
                   keyType       = 'SYMBOL',
                   ont           = "BP",
                   pAdjustMethod = "BH",
                   pvalueCutoff  = 0.01,
                   qvalueCutoff  = 0.05)

ego_CC@result$Description <- substring(ego_CC@result$Description,1,70)
ego_MF@result$Description <- substring(ego_MF@result$Description,1,70)
ego_BP@result$Description <- substring(ego_BP@result$Description,1,70)

p_BP <- barplot(ego_BP,showCategory = 10) + ggtitle("barplot for Biolog
ical process")
p_CC <- barplot(ego_CC,showCategory = 10) + ggtitle("barplot for Cellul
ar component")
p_MF <- barplot(ego_MF,showCategory = 10) + ggtitle("barplot for Molecu
lar function")
plotc <- p_BP/p_CC/p_MF
ggsave('enrichGO_down.png',path = "E:/MangeXU/PBMC_Presentation/Enrich
", plotc, width = 15,height = 20)
```

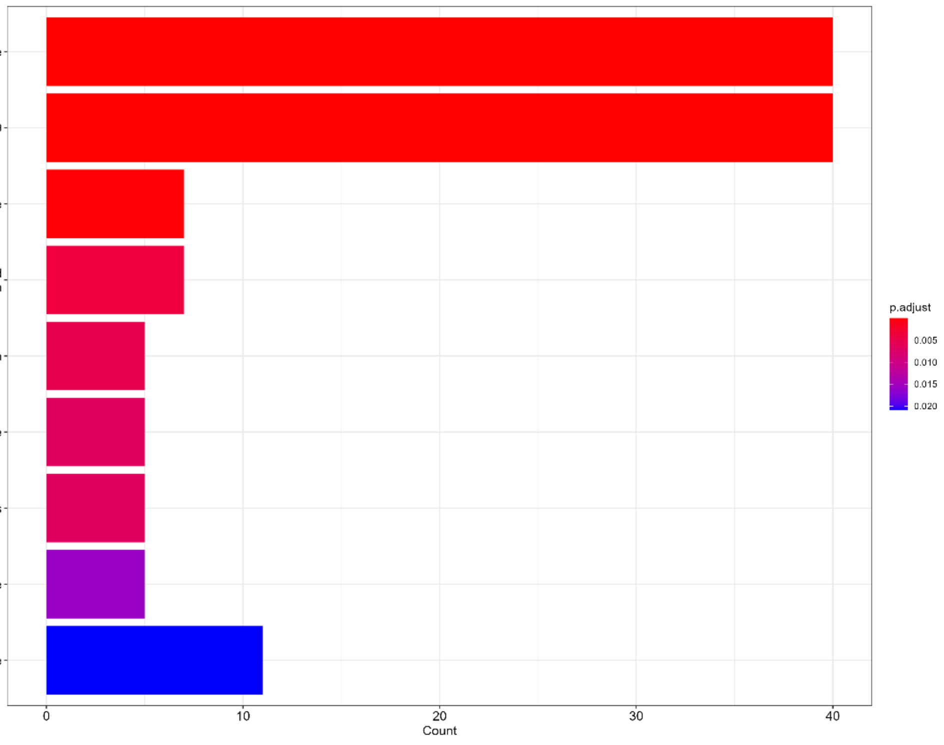


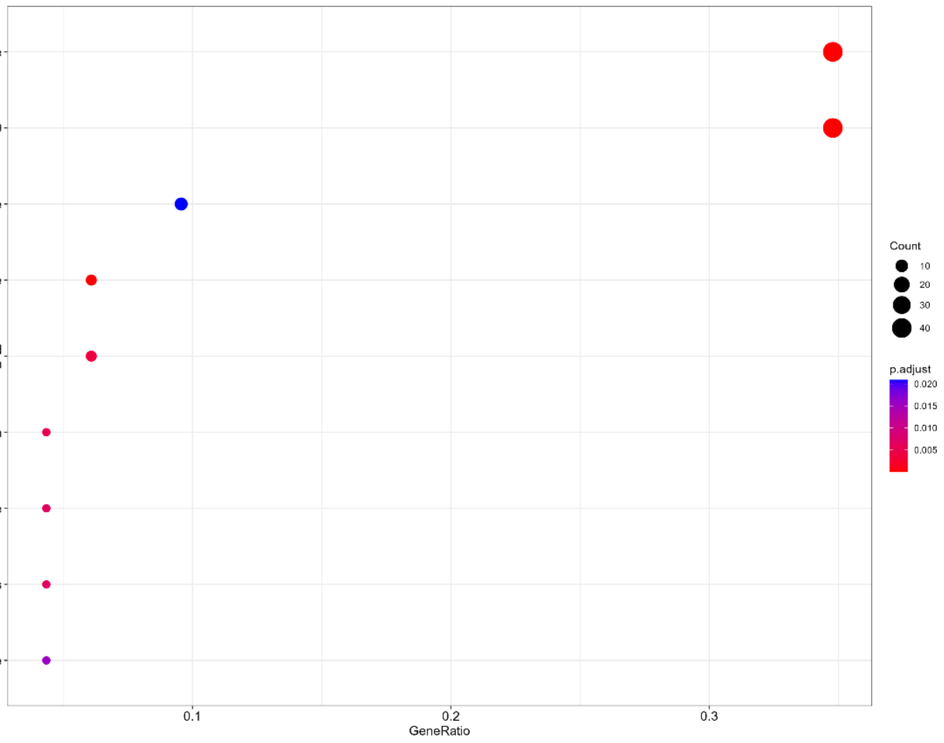
```
#Go Up
ego_ALL <- enrichGO(gene          = row.names(gene_up),
                    OrgDb         = 'org.Hs.eg.db',
                    keyType       = 'SYMBOL',
                    ont           = "ALL",
                    pAdjustMethod = "BH",
                    pvalueCutoff  = 0.01,
                    qvalueCutoff  = 0.05)
ego_all <- data.frame(ego_ALL)
write.csv(ego_all,'E:/MangeXU/PBMC_Presentation/Enrich/enrichGO_UP.csv
')

ego_CC <- enrichGO(gene          = row.names(gene_up),
                    OrgDb         = 'org.Hs.eg.db',
                    keyType       = 'SYMBOL',
                    ont           = "CC",
                    pAdjustMethod = "BH",
                    pvalueCutoff  = 0.01,
                    qvalueCutoff  = 0.05)

ego_MF <- enrichGO(gene          = row.names(gene_up),
                    OrgDb         = 'org.Hs.eg.db',
                    keyType       = 'SYMBOL',
                    ont           = "MF",
                    pAdjustMethod = "BH",
                    pvalueCutoff  = 0.01,
                    qvalueCutoff  = 0.05)

ego_BP <- enrichGO(gene          = row.names(gene_up),
                    OrgDb         = 'org.Hs.eg.db',
                    keyType       = 'SYMBOL',
                    ont           = "BP",
                    pAdjustMethod = "BH",
                    pvalueCutoff  = 0.01,
                    qvalueCutoff  = 0.05)

ego_CC@result$Description <- substring(ego_CC@result$Description,1,70)
ego_MF@result$Description <- substring(ego_MF@result$Description,1,70)
ego_BP@result$Description <- substring(ego_BP@result$Description,1,70)

p_BP <- barplot(ego_BP,showCategory = 10) + ggtitle("barplot for Biolog
ical process")
p_CC <- barplot(ego_CC,showCategory = 10) + ggtitle("barplot for Cellul
ar component")
p_MF <- barplot(ego_MF,showCategory = 10) + ggtitle("barplot for Molecu
lar function")
plotc <- p_BP/p_CC/p_MF
ggsave('enrichGO_up.png',path = "E:/MangeXU/PBMC_Presentation/Enrich",
plotc, width = 15,height = 20)
```



```r
#KEGG DOWN

genelist <- bitr(row.names(gene_down), fromType="SYMBOL",
                 toType="ENTREZID", OrgDb="org.Hs.eg.db")
```

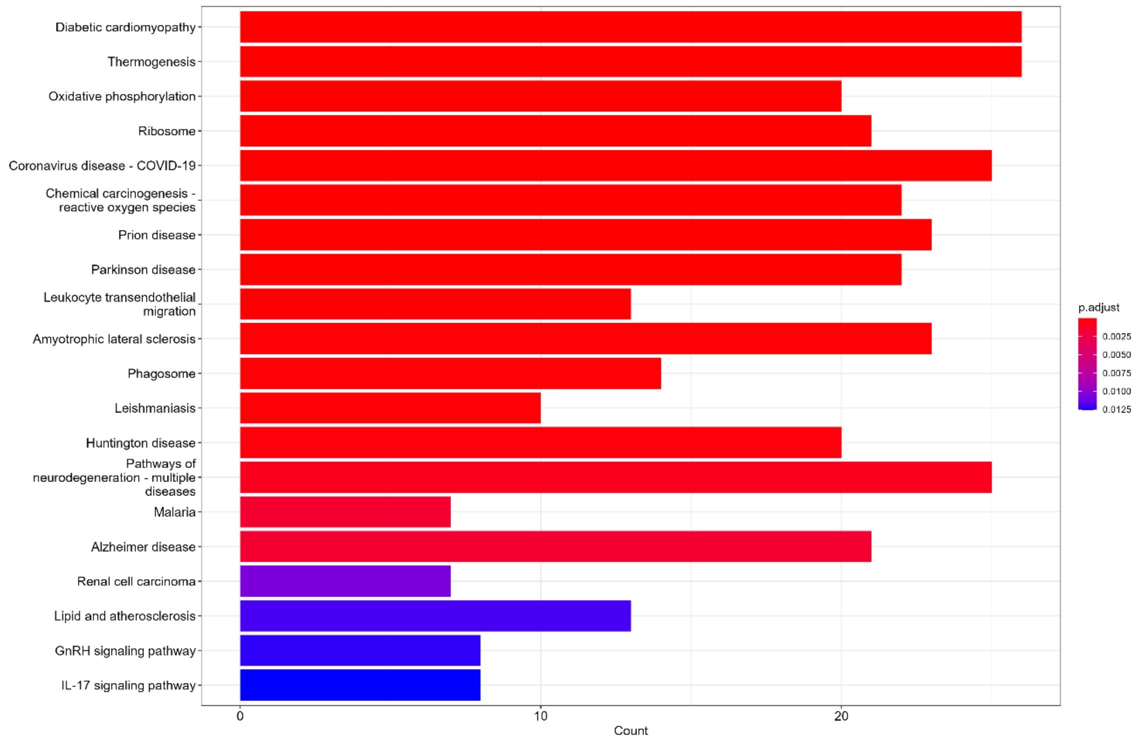
```
## 'select()' returned 1:1 mapping between keys and columns

## Warning in bitr(row.names(gene_down), fromType = "SYMBOL", toType =
## "ENTREZID", : 29.41% of input gene IDs are fail to map...

genelist <- pull(genelist,ENTREZID)
R.utils::setOption("clusterProfiler.download.method","auto")
kegg <- enrichKEGG(genelist, organism = "hsa",keyType = "kegg",pvalueCu
toff = 0.05)

## Reading KEGG annotation online:

## Reading KEGG annotation online:

p1 <- barplot(kegg, showCategory=20)
p2 <- dotplot(kegg, showCategory=20)
plotc = p1/p2
ggsave("enrichKEGG_down.png",path = "E:/MangeXU/PBMC_Presentation/Enric
h", plot = plotc, width = 15, height = 20)
```

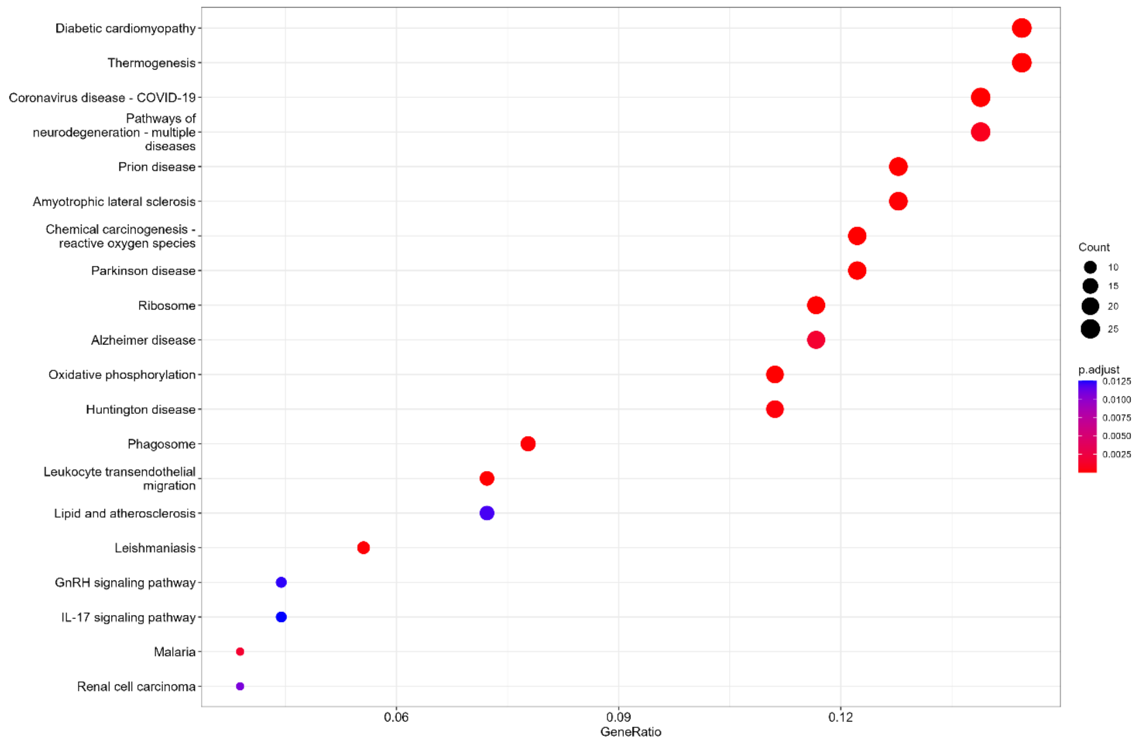




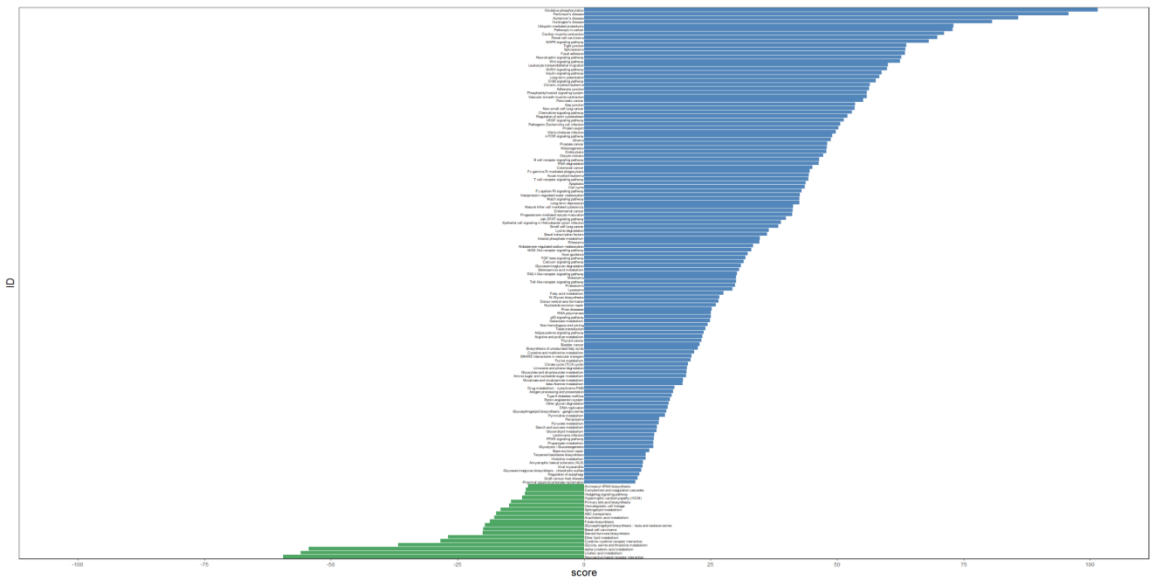
```
#KEGG UP
genelist <- bitr(row.names(gene_up), fromType="SYMBOL",
                 toType="ENTREZID", OrgDb="org.Hs.eg.db")

## 'select()' returned 1:1 mapping between keys and columns
## Warning in bitr(row.names(gene_up), fromType = "SYMBOL", toType = "E
NTREZID", :
## 7.37% of input gene IDs are fail to map...
```

```
genelist <- pull(genelist,ENTREZID)
kegg <- enrichKEGG(genelist, organism = "hsa", pvalueCutoff = 0.05)
p1 <- barplot(kegg, showCategory=20)
p2 <- dotplot(kegg, showCategory=20)
plotc = p1/p2
ggsave("enrichKEGG_up.png",path = "E:/MangeXU/PBMC_Presentation/Enrich
", plot = plotc, width = 15, height = 20)
```

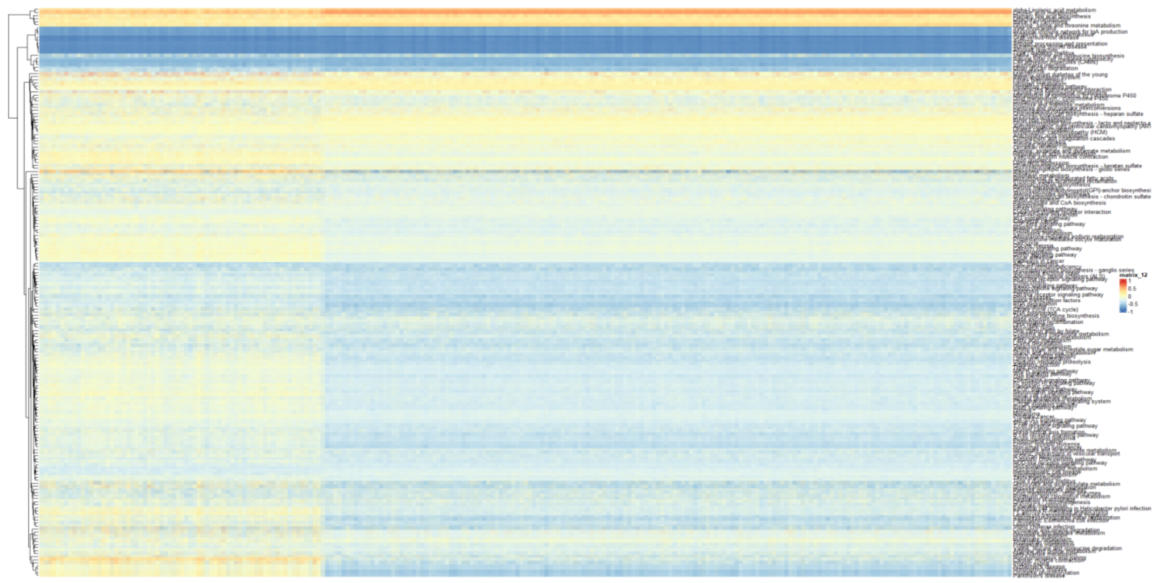

Diabetic cardiomyopathy
Thermogenesis
Oxidative phosphorylation
Ribosome
Coronavirus disease - COVID-19
Chemical carcinogenesis - reactive oxygen species
Prion disease
Parkinson disease
Leukocyte transendothelial migration
Amyotrophic lateral sclerosis
Phagosome
Leishmaniasis
Huntington disease
Pathways of neurodegeneration - multiple diseases
Malaria
Alzheimer disease
Renal cell carcinoma
Lipid and atherosclerosis
GnRH signaling pathway
IL-17 signaling pathway
0
10
20
Count
p.adjust
0.0025
0.0050
0.0075
0.0100
0.0125


Diabetic cardiomyopathy
Thermogenesis
Coronavirus disease - COVID-19
Pathways of neurodegeneration - multiple diseases
Prion disease
Amyotrophic lateral sclerosis
Chemical carcinogenesis - reactive oxygen species
Parkinson disease
Ribosome
Alzheimer disease
Oxidative phosphorylation
Huntington disease
Phagosome
Leukocyte transendothelial migration
Lipid and atherosclerosis
Leishmaniasis
GnRH signaling pathway
IL-17 signaling pathway
Malaria
Renal cell carcinoma
0.06
0.09
0.12
GeneRatio
Count
10
15
20
25
p.adjust
0.0125
0.0100
0.0075
0.0050
0.0025

```
#GSVA

## counts 数据
pbmc_final <- NormalizeData(pbmc_final) %>% FindVariableFeatures() %>%
ScaleData()

## Centering and scaling data matrix

expr=as.matrix(pbmc_final@assays$RNA@counts)

## 准备基因集(C7)
genesets = msigdbr(species = "Homo sapiens", category = "C2",subcategor
y = "KEGG")
write.csv(genesets,file = "E:/MangeXU/PBMC_Presentation/Enrich/KEGG_Gen
eSets.CSV")
keggSet = genesets%>% split(x = .$gene_symbol, f = .$gs_description)

##运行 gsva
keggEs <- gsva(expr, gset.idx.list = keggSet, kcdf="Gaussian", paralle
l.sz=1)

dim(keggEs)

## [1]  186 3808

grouP <- pbmc_final$orig.ident%>% as.factor()
keggEs<-keggEs[,order(grouP)]
grouP<-grouP[order(grouP)]
desigN <- model.matrix(~ grouP + 0)

comparE <- makeContrasts(grouPpbmc1k - grouPpbmc3k, levels=desigN)
fiT1 <- lmFit(keggEs, desigN)
fiT2 <- contrasts.fit(fiT1, comparE)
fiT3 <- eBayes(fiT2)
keggDiff <- topTable(fiT3, coef=1, number=200)


df <- data.frame(ID = rownames(keggDiff), score = keggDiff$t )
df$group =sapply(1:nrow(keggDiff),function(x){if(keggDiff[x,"logFC"]>0
& keggDiff[x,"adj.P.Val"]<0.001 & keggDiff[x,"t"]> 10){return("up")}els
e if(keggDiff[x,"logFC"]<0 & keggDiff[x,"adj.P.Val"]<0.001 & keggDiff
[x,"t"]< -10) {return("down")} else{return("noSig")} })
df1<- df[which(df$group != "noSig"),]

df1$hjust = ifelse(df1$score>0,1,0)
df1$nudge_y = ifelse(df1$score>0,-0.1,0.1)
sortdf1 <- df1[order(df1$score),]
sortdf1$ID <- factor(sortdf1$ID, levels = sortdf1$ID)
limt = max(abs(df1$score))

ggplot(sortdf1, aes(ID, score,fill=group))+geom_bar(stat = 'identity',a
lpha = 0.7)+
  scale_fill_manual(breaks=c("down","up"),values = c("#008020","#08519C
"))+
  geom_text(data = df1, aes(label = df1$ID, y = df1$nudge_y),nudge_x =
0,nudge_y =0,hjust =df1$hjust,size = 2)+
  scale_y_continuous(limits=c(-limt,limt),breaks = c(-100,-75,-50,-25,
0,25,50,75,100))+
  coord_flip()+
  theme_bw()+
  theme(panel.grid =element_blank())+
  theme(panel.border = element_rect(size = 1.0))+  theme(plot.title = e
lement_text(hjust = 0.5,size = 30),
          axis.text.y = element_blank(),
          axis.title = element_text(hjust = 0.5,size = 18),
          axis.line = element_blank(),
          axis.ticks.y = element_blank(),
          legend.position = limt)
```

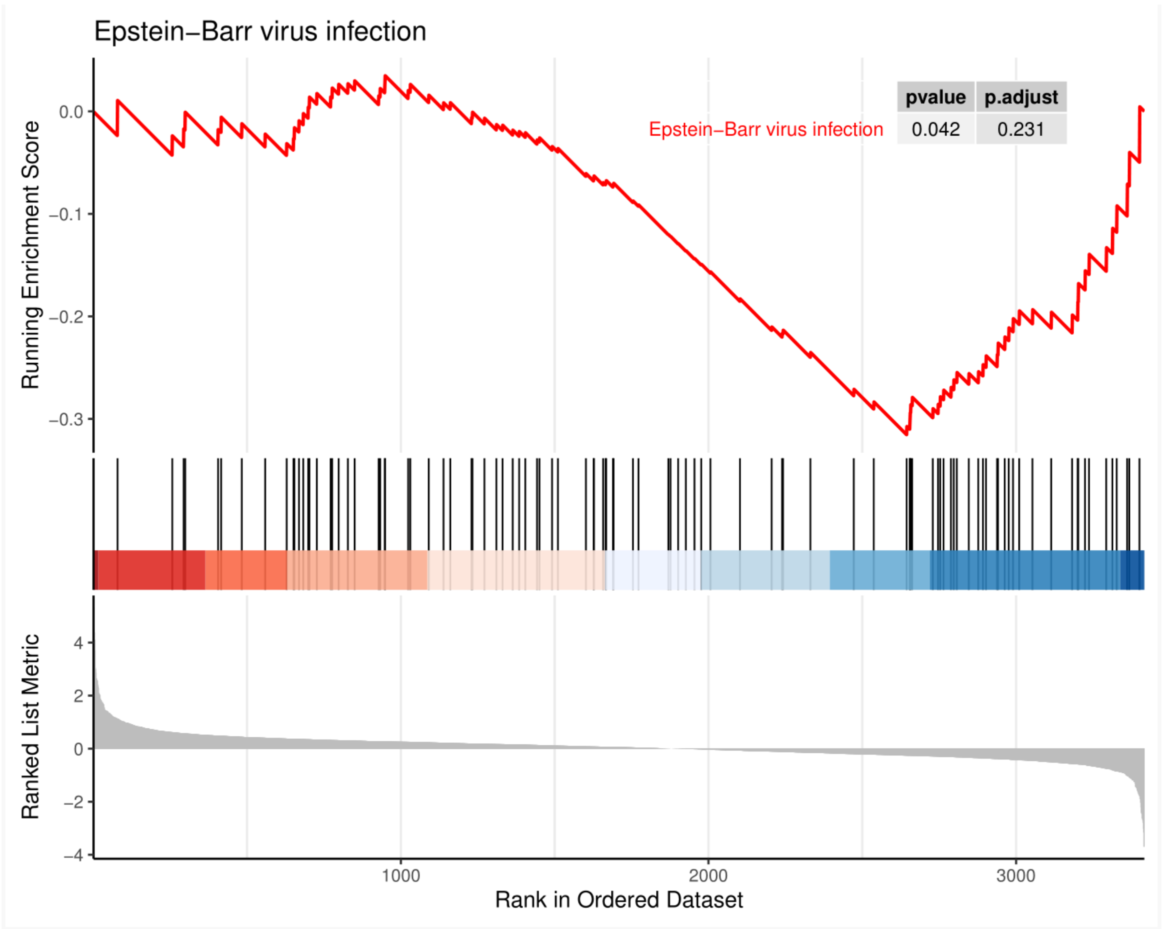


```
ggsave2("PBMC_GSVA.png", path = "E:/MangeXU/PBMC_Presentation/Enrich",
width = 20, height = 35 , units = "cm")

pheatmap(keggEs, show_colnames=F, cluster_cols = F, width = 45,height =
 50)
```



```
ggsave2(keggEs, filename = 'E:/MangeXU/PBMC_Presentation/Enrich/keggEs.
pdf', width =30, height =45)

#GSEA

Idents(pbmc_final)="orig.ident"
markers<-FindMarkers(pbmc_final,group.by="orig.ident", ident.1 = "pbmc1
k", ident.2 = "pbmc3k", min.pct = 0.1, logfc.threshold = 0)
need_DEG <- markers[,c(2,5)]
colnames(need_DEG) <- c('log2FoldChange','pvalue')
need_DEG$SYMBOL <- rownames(need_DEG)
df <- bitr(rownames(need_DEG),
           fromType = "SYMBOL",
           toType =  "ENTREZID",
           OrgDb = "org.Hs.eg.db")

need_DEG <- merge(need_DEG, df, by='SYMBOL')
geneList <- need_DEG$log2FoldChange
names(geneList) <- need_DEG$ENTREZID
geneList <- sort(geneList, decreasing = T)


#gsea 富集
KEGG_kk_entrez <- gseKEGG(geneList= geneList,
                   organism="hsa",
                   pvalueCutoff = 1)

KEGG_kk <- DOSE::setReadable(KEGG_kk_entrez,
                             OrgDb="org.Hs.eg.db",
                             keyType='ENTREZID')
#选取富集结果
kk_gse <- KEGG_kk
kk_gse_entrez <- KEGG_kk_entrez
```

```
#条件筛选
#一般认为|NES|>1，NOM pvalue<0.05，FDR（padj）<0.25 的通路是显著富集的
kk_gse_cut <- kk_gse[kk_gse$pvalue<0.05 & kk_gse$p.adjust<0.251 & abs(k
k_gse$NES)>1]
kk_gse_cut_down <- kk_gse_cut[kk_gse_cut$NES < 0,]
kk_gse_cut_up <- kk_gse_cut[kk_gse_cut$NES > 0,]

#选择展现 NES 前几个通路
down_gsea <- kk_gse_cut_down[tail(order(kk_gse_cut_down$NES,decreasing
= T),40),]
up_gsea <- kk_gse_cut_up[head(order(kk_gse_cut_up$NES,decreasing = T),4
0),]
diff_gsea <- kk_gse_cut[head(order(abs(kk_gse_cut$NES),decreasing = T),
40),]

#经典的 GSEA 图
down_gsea$Description

i=1
gseap1 <- gseaplot2(kk_gse,
                    down_gsea$ID[i],#富集的 ID 编号
                    title = down_gsea$Description[i],#标题
                    color = "red", #GSEA 线条颜色
                    base_size = 14,#基础字体大小
                    rel_heights = c(1.5, 0.5, 1),#副图的相对高度
                    subplots = 1:3,   #要显示哪些副图 如 subplots=c(1,3)
#只要第一和第三个图
                    ES_geom = "line", #enrichment score 用线还是用点"dot"
                    pvalue_table = T) #显示 pvalue 等信息
```



```
ggsave(gseap1, filename = 'E:/MangeXU/PBMC_Presentation/Enrich/GSEA.pdf
', width =10, height =8)

save.image(file = "PBMC_enrich.RData")
```