PAPER TITLE TO BE DEFINED (in common.yaml)

0-Microarray data analysis

2021-11-26 13:38:54 +0100

Abstract

Lung interstitium macrophages (IMs) are non-alveolar resident tissue macrophages which contribute to the lung homeostasis. These cells were reported to be heterogeneous by our group and other teams, which contains two main distinct subpopulations: CD206+ IMs and CD206- IMs. However, the exact origin of IMs and the transcriptional programs that control IM differentiation remains unclear. In recent report, we analyzed the refilled IMs in the course of time after induced IM depletion with single-cell RNA sequencing (10X Genomics Chromium) and bulk RNA sequencing.

Contents

1 Description

2 Load packages and data

```
# packages
library(ComplexHeatmap)
library(RColorBrewer)
library(circlize)

# data
dir.files <- "./downloaded_data"
files.list <- list.files(dir.files, pattern = "Gene_Expression_Activity.
   csv", full.names = T)
names.list <- sub(basename(files.list), pattern = "_in_Gene_Expression_
   Activity.csv",
    replacement = "")
```

read csv files and bind tables to one:

```
expr.table <- data.frame(read.csv(files.list[1]), row.names = 1)
expr.table <- expr.table[, -2]
n.rep <- length(2:ncol(expr.table))

names.rep <- paste(rep(names.list[1]), 1:n.rep, sep = "_")
colnames(expr.table)[2:ncol(expr.table)] <- names.rep

for (i in 2:length(files.list)) {
    tb <- read.csv(files.list[i])
    tb <- tb[, 4:ncol(tb)]
    n <- ncol(tb)
    repname <- paste(rep(names.list[i]), 1:n, sep = "_")
    colnames(tb) <- repname

    expr.table <- cbind(expr.table, tb)
    n.rep <- append(n.rep, n)
    names.rep <- append(names.rep, repname)
}
meta.sample <- data.frame(sampleName = names.list, n.rep = n.rep)
head(expr.table)
```

```
## # A tibble: 6 x 70
##   Gene.Symbol GEXC_AMs_1 GEXC_AMs_2 GEXC_AMs_3 `GEXC_DC_Lu_CD1~ `
   GEXC_DC_Lu_CD1~
##   <chr>             <dbl>      <dbl>      <dbl>            <dbl>
             <dbl>
## 1 ---                12.2       25.8      -0.23             64.1
              75.2
## 2 ---                17.3       33.8     -15.0              40.0
              60.2
## 3 ---                -2.6       30.2      -9.45             45.3
              59.4
## 4 ---                23.3       44.9      16.6              63.8
              65.3
```

```
## 5 ---                16.4       61.3      -11.1              22.4
                43.5
## 6 ---                19.8       44.8       14.6              75.4
                82.6
## # ... with 64 more variables: GEXC_DC_Lu_CD103+_3 <dbl>,
## #   GEXC_DC_Lu_CD103+_4 <dbl>, GEXC_DC_Lu_CD103+_5 <dbl>,
## #   GEXC_DC_Lu_CD24+_1 <dbl>, GEXC_DC_Lu_CD24+_2 <dbl>,
## #   GEXC_DCLuLN_CD103+_1 <dbl>, GEXC_DCLuLN_CD103+_2 <dbl>,
## #   GEXC_DCLuLN_CD103+_3 <dbl>, GEXC_DCLuLN_CD11b+_1 <dbl>,
## #   GEXC_DCLuLN_CD11b+_2 <dbl>, GEXC_DCLuLN_CD11b+_3 <dbl>, GEXC_L+WT_1
    <dbl>,
## #   GEXC_L+WT_2 <dbl>, GEXC_L+WT_3 <dbl>, GEXC_LMIsW_1 <dbl>, ...
```

3 Data preparration

3.1 Make a list with genes to show in heatmap

The gene list IMvs(AM&DC).csv is calculated in ImmGen Datasets. We compared IM microarray data to both MA and DC and get the DE genes.

```
probset.DE <- read.csv("./IMvs(AM&DC).csv")
probset.toShow <- unique(as.character(probset.DE$ProbeSet_ID))

# the table is ordered by ratio, so take the top 100:
probset.toShow <- probset.toShow[1:100]
```

Then base on the intensity in IM, we choose the only top 50 probsets.

Take the to 50 probsets with highest intensity:

```
probset.DE <- probset.DE[order(probset.DE$Mean_A, decreasing = TRUE), ]
probset.top50 <- as.character(probset.DE[1:50, "ProbeSet_ID"])
```

```
probset.toShow <- intersect(probset.top50, probset.toShow)
```

subset expr.table

```
expr.table.toShow <- expr.table[probset.toShow, ]
genes.toShow <- unique(expr.table.toShow$Gene.Symbol)
length(genes.toShow)
```

```
## [1] 50
```

As genes are unique to each probset, we can use gene symbols as rownames.

```
rownames(expr.table.toShow) <- expr.table.toShow$Gene.Symbol
rownames(expr.table.toShow)
```

```
##  [1] "C1qa"               "Cxcl10"             "C1qb"
##  [4] "Mmp12 or Mmp1b"     "C1qc"               "Ptgs2"
##  [7] "C3ar1"              "Ccl4"               "Itgam"
## [10] "Cx3cr1"             "Cd72 or Tesk1"      "Col14a1"
## [13] "Mmp13"              "Pla2g7"             "Mafb"
## [16] "Rasgrp1"            "Abca9"              "Ms4a4a"
## [19] "Ephx1"              "Hpgd"               "Ecm1"
## [22] "Cxcl13"             "Lifr"               "Ccl2"
## [25] "Ms4a6b"             "Hpgds"              "Plk2"
## [28] "---"                "Stab1"              "Itga9"
## [31] "Ifnb1"              "Cmklr1"             "H2-M2"
## [34] "Blnk"               "Emr4"               "Dnajb4"
## [37] "Ccl7"               "Ms4a7"              "Tmem119"
## [40] "Clec10a"            "Retnla"             "Tm4sf19"
## [43] "Olfr111"            "Abcc3"              "Ifit1"
## [46] "Gbp3"               "St3gal6 or Dcbld2" "Rtp4"
## [49] "Maf"                "St8sia6"
```

Remove the row without annotation:

```
expr.table.toShow <- expr.table.toShow[expr.table.toShow$Gene.Symbol != "
   ---", ]
```

3.2 Make metadata table

```
data.frame(meta.sample$sampleName, order = 1:nrow(meta.sample))
```

```
## # A tibble: 22 x 2
##    meta.sample.sampleName  order
##    <chr>                   <int>
##  1 GEXC_AMs                    1
##  2 GEXC_DC_Lu_CD103+           2
##  3 GEXC_DC_Lu_CD24+            3
##  4 GEXC_DCLuLN_CD103+          4
##  5 GEXC_DCLuLN_CD11b+          5
##  6 GEXC_L+WT                   6
##  7 GEXC_LMIsW                  7
##  8 GEXC_MF_BM                  8
##  9 GEXC_MF_CNS                 9
## 10 GEXC_MF_Lu_CD11b+_CD24-    10
## # ... with 12 more rows
```

```
meta.sample$cellType3 <- c("Mac␣Alv␣Lu", #1
                           "DC␣CD103+␣Lu", #2
                           "DC␣CD24+␣Lu", #3
                           "DC␣CD103+␣LuLN", #4
                           "DC␣CD11b+␣LuLN", #5
                           "Mo␣Ly6C+␣Lu", #6
                           "Mac␣Int␣Lu", #7
                           "Mac␣BM", #8
                           "Mac␣CNS", #9
                           "Mac␣Int␣Lu", #10
                           "Mac␣F4/80hi␣PC", #11
                           "Mac␣F4/80lo␣PC", #12
                           "Mac␣SI", #13
                           "Mac␣SLN", #14
                           "Mac␣SP", #15
                           "Mo␣Ly6C-␣MHCII-␣BL", #16
                           "Mo␣Ly6C-␣MHCII+␣BL", #17
                           "Mo␣Ly6C-␣MHCIIint␣BL", #18
                           "Mo␣Ly6C+␣MHCII-␣BL", #19
                           "Mo␣Ly6C+␣MHCII+␣BL", #20
                           "Mo␣Ly6C-␣MHCII-␣BM", #21
                           "Mo␣Ly6C+␣MHCII-␣BM" #22
              )

meta.sample$cellType <- c("aMac", #1
              rep("DC", 4),  # 2-5
              "Mo", # 6
              "iMac", #7
              rep("Mac", 2), # 8-9
              "iMac", #10
              rep("Mac", 5), # 11-15
              rep("Mo", 7) # 16-22
              )
meta.sample$organ <- c(rep("Lu", 3), #1-3
           rep("LuLN", 2), #4-5
           rep("Lu", 2), #6-7
           "BM", #8
           "CNS", #9
           "Lu", #10
           rep("PC",2), #11-12
           "SI", #13
           "SLN", #14
           "SP", #15
           rep("BL",5 ),
           rep("BM", 2)

           )
meta.sample$cellType2 <- c(
              "Mac", #1
              rep("DC", 4),  # 2-5
              "Mo", # 6
              "Mac", #7
              rep("Mac", 2), # 8-9
              "Mac", #10
              rep("Mac", 5), # 11-15
              rep("Mo", 7) # 16-22
              )
meta.sample$organ2 <- c(
            "Lu-Alv", #1
            rep("Lu", 2), #2-3
           rep("LuLN", 2), #4-5
           "Lu", #6
           "Lu-Int", #7
           "BM", #8
           "CNS", #9
           "Lu-Int", #10
           rep("PC",2), #11-12
           "SI", #13
           "SLN", #14
           "SP", #15
           rep("BL",5 ),
           rep("BM", 2)

           )
```

```
meta.table <- data.frame(CellType = rep(meta.sample$cellType, meta.sample$
   n.rep),
    OrganType = rep(meta.sample$organ, meta.sample$n.rep), CellType2 = rep
       (meta.sample$cellType2,
        meta.sample$n.rep), OrganType2 = rep(meta.sample$organ2, meta.
           sample$n.rep),
    cellType3 = rep(meta.sample$cellType3, meta.sample$n.rep), row.names =
        names.rep)
```

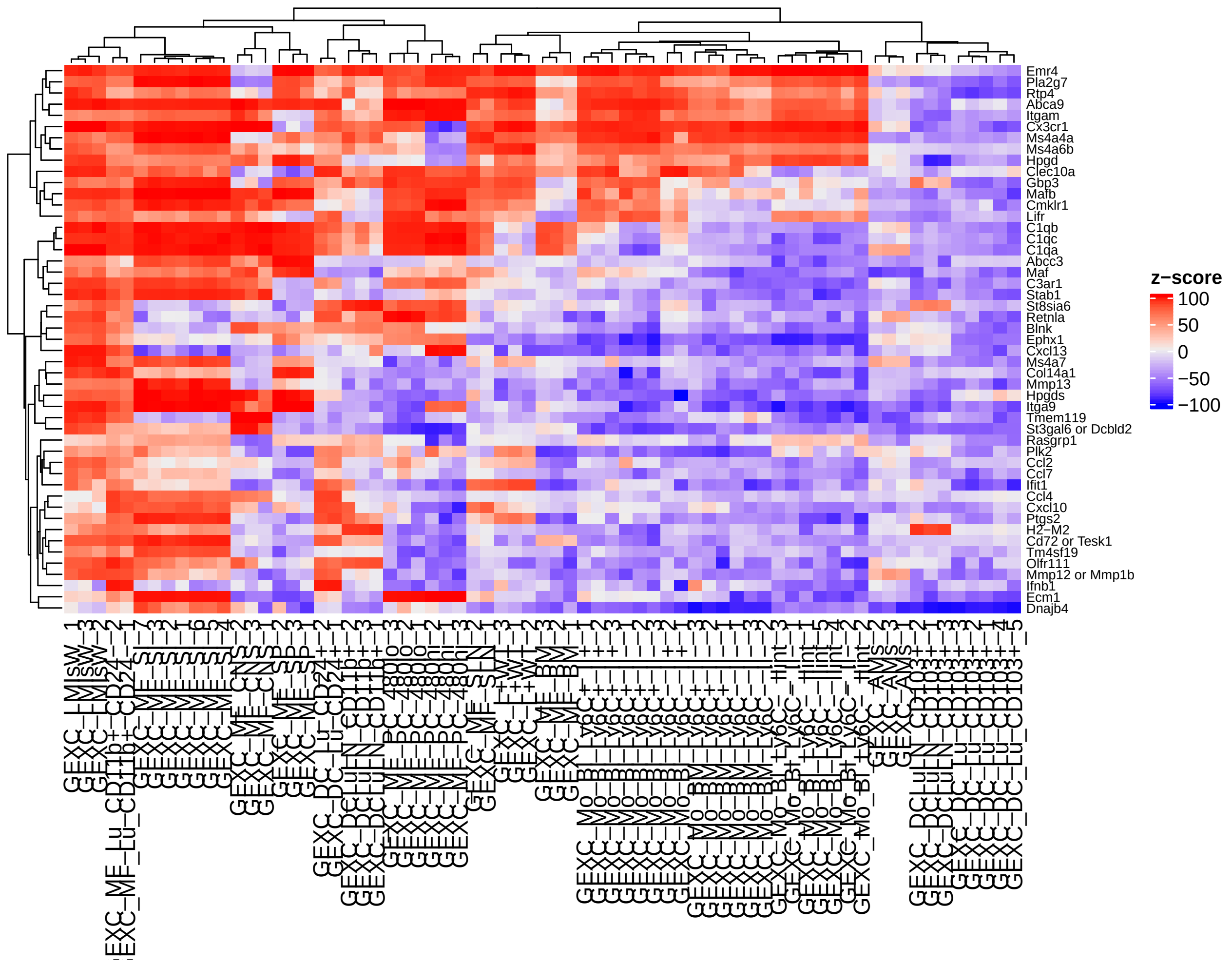
```
HeatmapAnnotation(Cell_type = meta.table$CellType, Organ_type = meta.table
   $OrganType)
```

```
## A HeatmapAnnotation object with 2 annotations
##   name: heatmap_annotation_0
##   position: column
##   items: 69
##   width: 1npc
##   height: 10.3514598035146mm
##   this object is subsetable
##   23.1191666666667mm extension on the right
##
##        name annotation_type color_mapping height
##   Cell_type discrete vector        random    5mm
##  Organ_type discrete vector        random    5mm
```

4 Make heatmaps

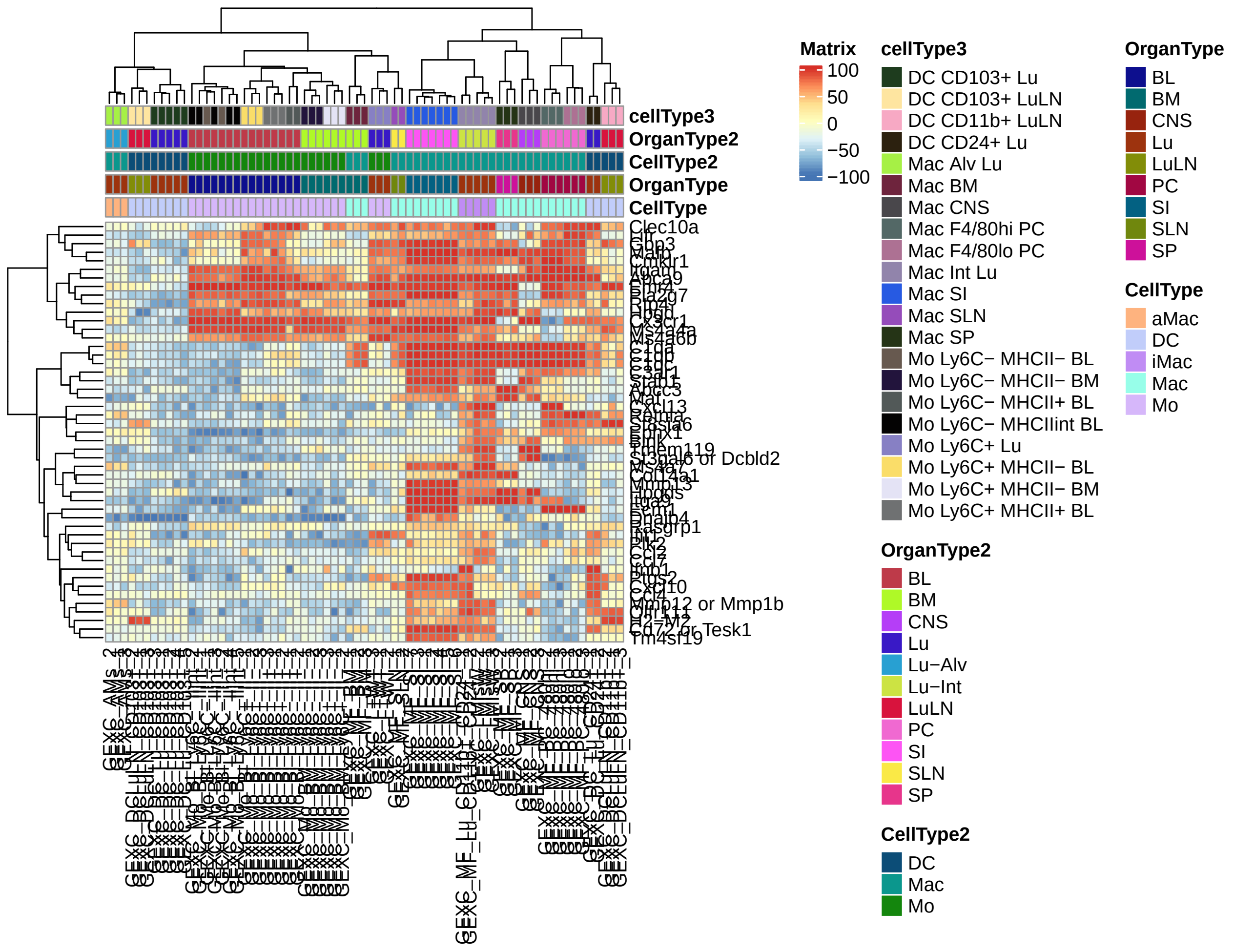
Use Heatmap:

```
Heatmap(
  as.matrix(expr.table.toShow[2:ncol(expr.table.toShow)]),
  # use_raster = FALSE, # use FALSE to export to vector image.
  name                         = "z-score",
  # col                          = colorRamp2(seq(from=-2,to=2,length=11),
     rev(brewer.pal(11, "Spectral"))),
  # show_row_names               = TRUE,
  # show_column_names            = FALSE,
  row_names_gp                 = gpar(fontsize = 7),

  # row_title_rot                = 0,
  # cluster_rows                 = TRUE,
  # cluster_row_slices           = FALSE,
  #cluster_columns             = FALSE
)
```



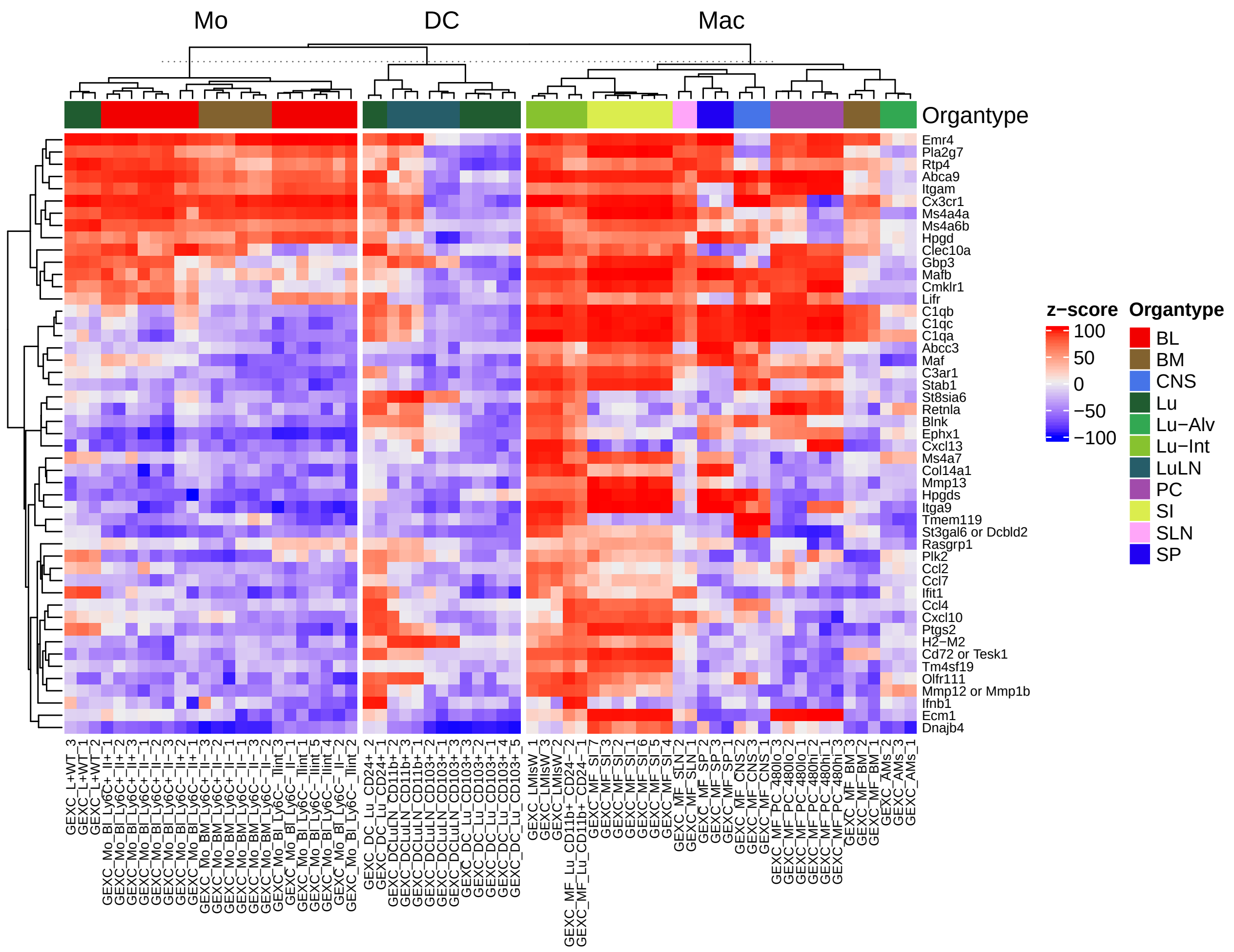
Using pheatmap.

```
pheatmap(
  as.matrix(expr.table.toShow[2:ncol(expr.table.toShow)]), annotation_col = meta.table
  # use_raster = FALSE, # use FALSE to export to vector image.
  #name                         = "z-score",
  # col                          = colorRamp2(seq(from=-2,to=2,length=11),
     rev(brewer.pal(11, "Spectral"))),
  # show_row_names               = TRUE,
  # show_column_names            = FALSE,
  #row_names_gp                 = gpar(fontsize = 7),

  # row_title_rot                = 0,
  # cluster_rows                 = TRUE,
  # cluster_row_slices           = FALSE,
  #cluster_columns              = FALSE
)
```



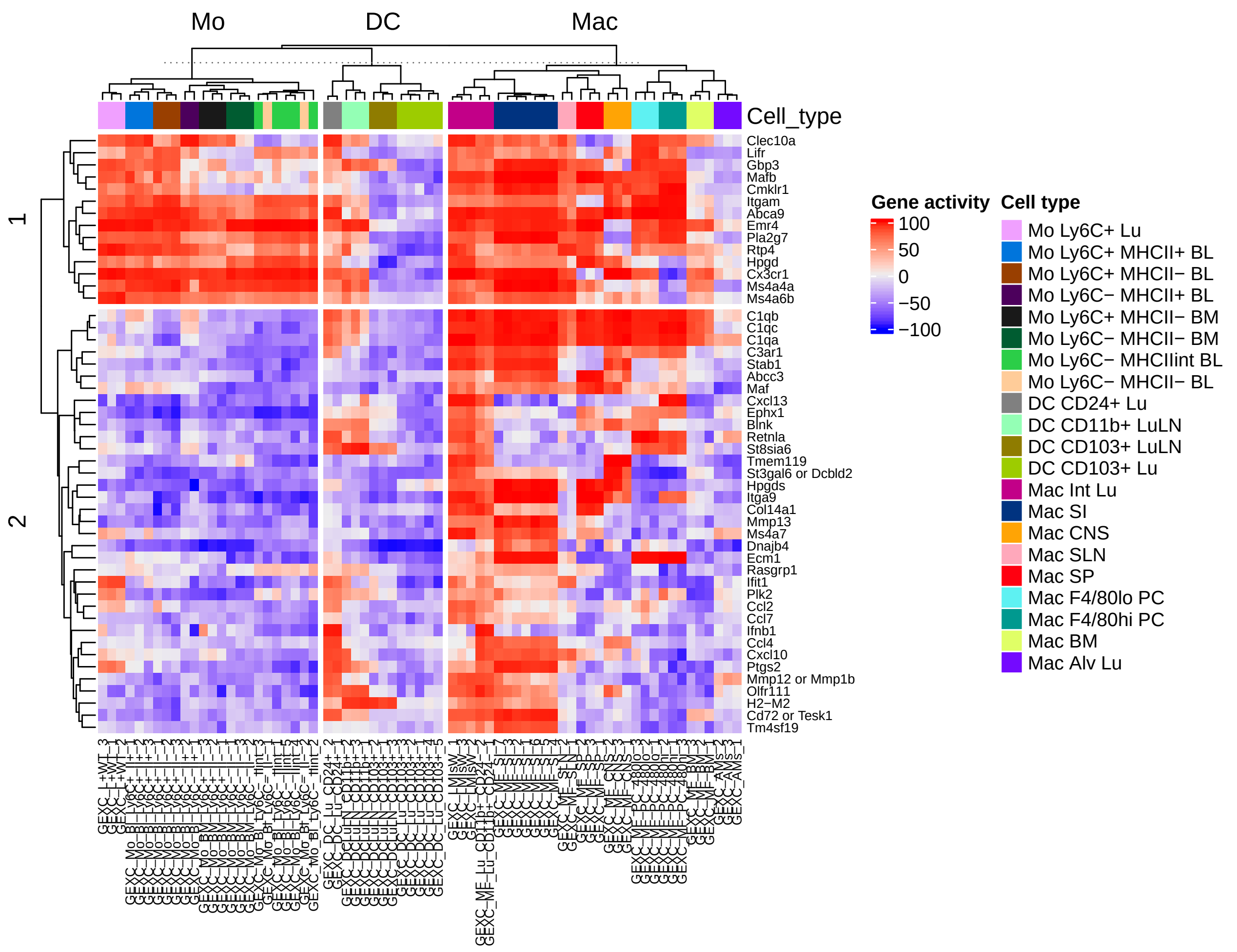
Using Heatmap with annotations.

```
hp <- Heatmap(
  as.matrix(expr.table.toShow[2:ncol(expr.table.toShow)]),
  # use_raster = FALSE, # use FALSE to export to vector image.
  name                            = "z-score",
  # col                             = colorRamp2(seq(from=-2,to=2,length=11), rev(brewer.pal(11, "Spectral"))),
  # show_row_names                  = TRUE,
  # show_column_names               = FALSE,
  row_names_gp                    = gpar(fontsize = 7),
  column_names_gp                 = gpar(fontsize = 7),

  #column_split = meta.table$CellType2,
  column_split = factor(meta.table$CellType2, levels = c("Mac", "Mo", "DC")),
  top_annotation = HeatmapAnnotation(Organtype=meta.table$OrganType2,
                                     col = list(Organtype = c(`Lu-Alv`="#32a852",
                                                          `Lu-Int`="#87c22f",
                                                          Lu="#205c30",
                                                          LuLN="#265d69",
                                                          BM="#82622f",
                                                          CNS="#4674e8",
                                                 PC="#a14bab",
                                                 SI="#dbed4e",
                                                 SLN="#ffa6f9",
                                                 SP="#2000f2",
                                                 BL="#f20000")) )

)

p <- draw(hp)
```

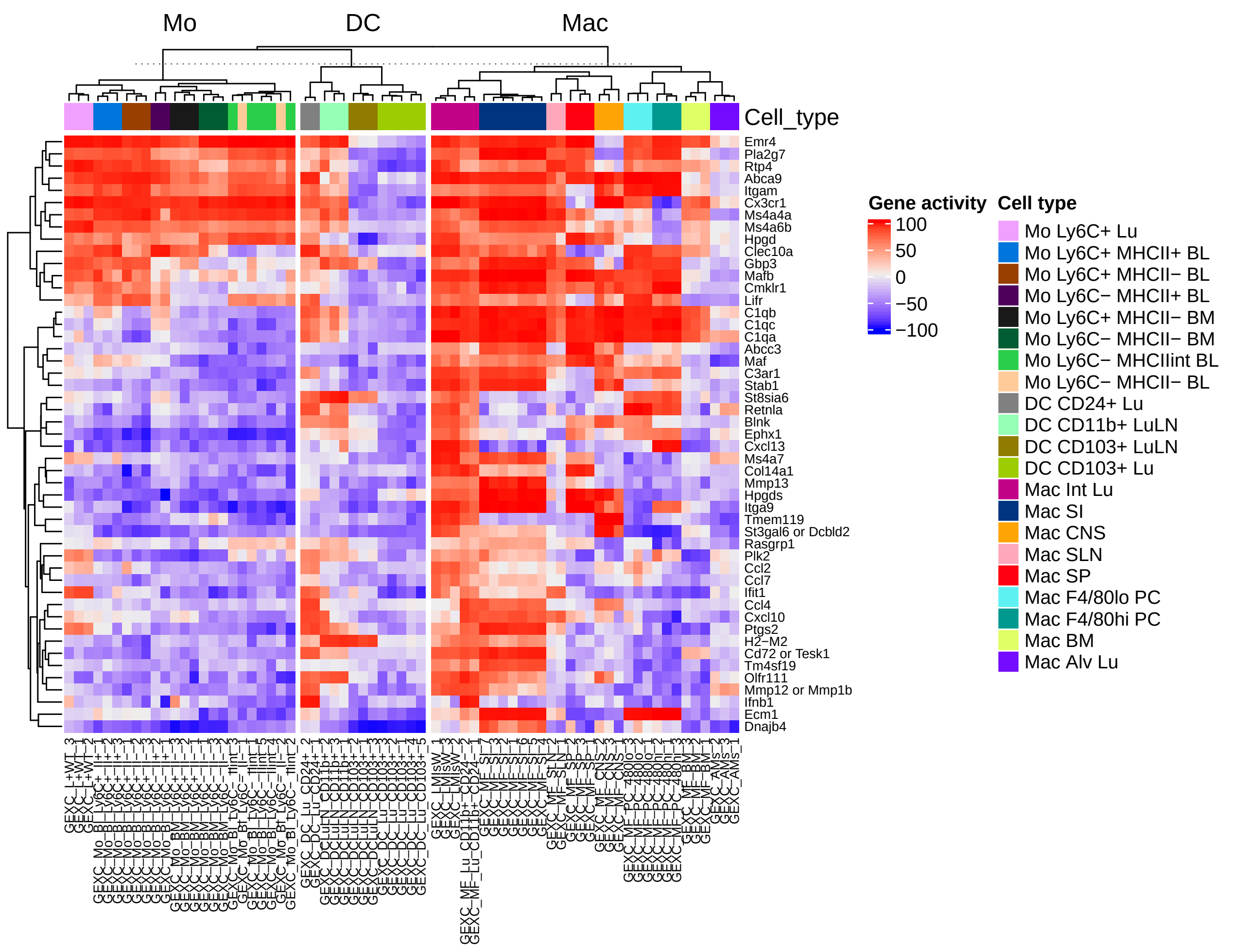


Change colors and annotations

```
col.cellType3 <- read.csv("../0-Microarrays/colors_celltype3.csv", sep = "
   \t", header = FALSE, row.names = 1)
colors.cellType3 <- as.character(col.cellType3$V2)
names(colors.cellType3) <- rownames(col.cellType3)
genes.toSplit <- rownames(expr.table.toShow)
genes.toSplit <- genes.toSplit %in% c("Tmem119", "Cx3cr1")

# the one with row split:
hp2 <- Heatmap(
  as.matrix(expr.table.toShow[2:ncol(expr.table.toShow)]),
  # use_raster = FALSE, # use FALSE to export to vector image.
  name                            = "Gene␣activity",
  row_names_gp                    = gpar(fontsize = 7),
  column_names_gp                 = gpar(fontsize = 7),
  column_split = factor(meta.table$CellType2, levels = c("Mac", "Mo", "DC")),
  row_split = 2,
  top_annotation = HeatmapAnnotation(Cell_type=meta.table$cellType3,
                       col = list(
                             Cell_type = colors.cellType3 ),
                       annotation_legend_param = list(
                             Cell_type = list(title = "Cell␣type",
                                     at = names(colors.cellType3),
                                     labels = names(colors.cellType3))) )

)

p2 <- draw(hp2)
```



```
# the one WITHOUT row split:
hp3 <- Heatmap(
  as.matrix(expr.table.toShow[2:ncol(expr.table.toShow)]),
  # use_raster = FALSE, # use FALSE to export to vector image.
  name                          = "Gene␣activity",
  row_names_gp                  = gpar(fontsize = 7),
  column_names_gp               = gpar(fontsize = 7),
  column_split = factor(meta.table$CellType2, levels = c("Mac", "Mo", "DC"
     )),
  #row_split = 2,
  top_annotation = HeatmapAnnotation(Cell_type=meta.table$cellType3,
                             col = list(
                                         Cell_type = colors.cellType3 ),
                             annotation_legend_param = list(
                                         Cell_type = list(title = "Cell␣type",
                                                          at = names(colors.
                                                             cellType3),
                                                          labels = names(colors.
                                                             cellType3))) )

)

p3 <- draw(hp3)
```



5 Session information

R sesssion:

```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.3 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] circlize_0.4.13      RColorBrewer_1.1-2   ComplexHeatmap_2.6.2
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7          highr_0.9           pillar_1.6.2
##  [4] compiler_4.0.3      formatR_1.11        tools_4.0.3
##  [7] digest_0.6.27       evaluate_0.14       lifecycle_1.0.0
## [10] tibble_3.1.3        clue_0.3-59         pkgconfig_2.0.3
## [13] png_0.1-7           rlang_0.4.11        rstudioapi_0.13
## [16] cli_3.0.1           magick_2.7.2        yaml_2.2.1
## [19] parallel_4.0.3      xfun_0.24           stringr_1.4.0
## [22] knitr_1.33          cluster_2.1.0       GlobalOptions_0.1.2
## [25] vctrs_0.3.8         S4Vectors_0.28.1    IRanges_2.24.1
## [28] stats4_4.0.3        GetoptLong_1.0.5    fansi_0.5.0
## [31] rmarkdown_2.9       magrittr_2.0.1      matrixStats_0.60.0
## [34] ellipsis_0.3.2      htmltools_0.5.1.1   BiocGenerics_0.36.1
## [37] shape_1.4.6         colorspace_2.0-2    utf8_1.2.2
## [40] stringi_1.7.3       crayon_1.4.1        rjson_0.2.20
## [43] Cairo_1.5-12.2
```

6 References