Heatmap e comparações de TCR e BCR em ACC

Jean Resende

Contexto

Heatmaps

Abundância

```r
# -- pacotes necessarios
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(tidyr)
library(tibble)
library(circlize)
```

```
========================================
circlize version 0.4.16
CRAN page: https://cran.r-project.org/package=circlize
Github page: https://github.com/jokergoo/circlize
Documentation: https://jokergoo.github.io/circlize_book/book/

If you use it in published research, please cite:
Gu, Z. circlize implements and enhances circular visualization
  in R. Bioinformatics 2014.

This message can be suppressed by:
  suppressPackageStartupMessages(library(circlize))
========================================
```

```r
library(RColorBrewer)
library(ComplexHeatmap)
```

```
Loading required package: grid
```

```
========================================
ComplexHeatmap version 2.18.0
Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
Github page: https://github.com/jokergoo/ComplexHeatmap
Documentation: http://jokergoo.github.io/ComplexHeatmap-reference

If you use it in published research, please cite either one:
- Gu, Z. Complex Heatmap Visualization. iMeta 2022.
- Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
    genomic data. Bioinformatics 2016.


The new InteractiveComplexHeatmap package can directly export static
complex heatmaps into an interactive Shiny app with zero effort. Have a try!

This message can be suppressed by:
  suppressPackageStartupMessages(library(ComplexHeatmap))
========================================
```

```r
library(TCGAbiolinks)
```

```r
load("../../03_exploratory_analysis/tcrbcr_clones.RData")
load("../../../../Projetos/Bigdata/BigData/BigData/repertorio_tcrbcr_acc/data/coldataACC.R
```

```r
data <- tcrbcr_clones

data$chain <- ifelse(
  substr(data$V, 1,3) == "IGH", "IGH",
  ifelse(substr(data$V, 1,3) == "IGK", "IGK",
         ifelse(substr(data$V, 1,3) == "IGL", "IGL",
                ifelse(substr(data$V,1,3) == "TRA", "TRA",
                       ifelse(substr(data$V,1,3) == "TRB", "TRB",
                              ifelse(substr(data$V, 1,3) == "TRG", "TRG",
                                     ifelse(substr(data$V, 1,3) == "TRD", "TRD",
                                            NA)))))))

#data$TCR <- sum(data$count[data$chain %in% c("TRA","TRB","TRG","TRD")])

mat_data <- data %>%
  group_by(sample_id, chain) %>%
  summarize(count = sum(count)) %>%
  pivot_wider(names_from = chain,
              values_from = count,
              values_fill = NA)
```

```
`summarise()` has grouped output by 'sample_id'. You can override using the
`.groups` argument.
```

```r
mat_data$IGK[is.na(mat_data$IGK)] <-  0
mat_data$IGL[is.na(mat_data$IGL)] <-  0
mat_data$TRB[is.na(mat_data$TRB)] <-  0
mat_data$TRG[is.na(mat_data$TRG)] <-  0
mat_data$IGH[is.na(mat_data$IGH)] <-  0
mat_data$TRA[is.na(mat_data$TRA)] <-  0
mat_data$TRD[is.na(mat_data$TRD)] <-  0

mat_data$TCR <- rowSums(mat_data[,c(4,5,7,8)])
mat_data$BCR <- rowSums(mat_data[,c(2,3,6)])
mat_data$mat_sum_tcrbcr <- mat_data$TCR + mat_data$BCR

mat_data <- column_to_rownames(mat_data, "sample_id")
```

```
#mat_data <- as.matrix(t(mat_data))
```

```
nrow(mat_data[mat_data$TCR == 0,])
```

```
[1] 11
```

```
nrow(mat_data[mat_data$BCR == 0,])
```

```
[1] 9
```

```
nrow(mat_data[mat_data$BCR != 0 & mat_data$TCR !=0,])
```

```
[1] 48
```

```
sum(mat_data$mat_sum_tcrbcr)
```

```
[1] 161148
```

```
sum(mat_data$TCR)
```

```
[1] 1485
```

```
sum(mat_data$BCR)
```

```
[1] 159663
```

```
coldataACC_clones <- data.frame(
  sample_id = coldataACC$sample_id,
  barcode = coldataACC$barcode,
  gender = coldataACC$gender,
    steroid = coldataACC$steroid,
    tumor_stage = coldataACC$tumor_stage,
    vital_status = coldataACC$vital_status,
    cortisol.excess = coldataACC$cortisol.excess,
    immune.subtype = coldataACC$immune.subtype,
    other.hormones = coldataACC$other.hormones,
    reads = coldataACC$reads)
```

```
# --  preprocessamento da matriz

# amostras que tiveram tcr e bcr extraidas por trust4
samples.match <- coldataACC_clones$sample_id[
  coldataACC_clones$sample_id %in% rownames(mat_data)]

# amostras que o trust4 nao extraiu tcr e bcr
idx <- match(coldataACC_clones$sample_id, rownames(mat_data))
table(is.na(idx))
```

```
FALSE  TRUE
   68    11
```

```
samples.no_match <- coldataACC_clones$sample_id[is.na(idx)] # 3 NAs pq nao foi baixado ess

# matriz contendo as amostras que nao tiveram tcr e bcr
names(mat_data)
```

```
 [1] "IGK"             "IGL"             "TRB"             "TRG"
 [5] "IGH"             "TRA"             "TRD"             "TCR"
 [9] "BCR"             "mat_sum_tcrbcr"
```

```
mat_data_add <- data.frame(IGK = rep(0,8),
                           IGL = rep(0,8),
                           TRB = rep(0,8),
                           TRG = rep(0,8),
                           IGH = rep(0,8),
                           TRA = rep(0,8),
                           TRD = rep(0,8),
                             TCR = rep(0,8),
                             BCR = rep(0,8),
                             mat_sum_tcrbcr = rep(0,8))

rownames(mat_data_add) <- na.omit(samples.no_match)

# completando a matriz geral
mat_data_all <- rbind(mat_data, mat_data_add)

# adicionando a coluna referente aos esteroides
idx <- match(rownames(mat_data_all), coldataACC_clones$sample_id)
mat_data_all$steroid <- coldataACC_clones$steroid[idx]

# ordenando a matriz
mat_data_all <- mat_data_all %>%
  arrange(desc(steroid), desc(mat_sum_tcrbcr))
```

```
rm("coldataACC", "data", "mat_data", "mat_data_add", "tcrbcr_clones", "idx", "samples.matc
```

```
#data <- t(mat_data_all[, 1:7])
data <- mat_data_all[, 1:7]
data <- log2(data + 1)
data <- scale(data)

#data <- t(data)

#data[data > 2] <- 2
#data[data < (-2)] <- (-2)
```

```
coldata <- coldataACC_clones[!is.na(coldataACC_clones$sample_id),]
rownames(coldata) <- coldata$sample_id

idx <- match(rownames(data), rownames(coldata))

coldata <- coldata[idx,]
rownames(data) == coldata$sample_id
```

```
 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[31] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
[46] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[61] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[76] TRUE
```

```
coldata$TCR <- mat_data_all$TCR
coldata$BCR <- mat_data_all$BCR
coldata$sum_TCR_BCR <- mat_data_all$mat_sum_tcrbcr
```

```
ACC_clinical <- GDCquery_clinic(project = "TCGA-ACC")

ACC_clinical$submitter_id
```

```
 [1] "TCGA-OR-A5JA" "TCGA-OR-A5JO" "TCGA-OR-A5LN" "TCGA-OR-A5L8" "TCGA-PK-A5HA"
 [6] "TCGA-OR-A5J4" "TCGA-OR-A5KY" "TCGA-OR-A5JW" "TCGA-OR-A5KW" "TCGA-OR-A5K2"
[11] "TCGA-OR-A5K4" "TCGA-OR-A5JG" "TCGA-OR-A5LR" "TCGA-OR-A5J5" "TCGA-OR-A5JS"
[16] "TCGA-PK-A5H9" "TCGA-P6-A5OF" "TCGA-PA-A5YG" "TCGA-OR-A5J2" "TCGA-OR-A5L4"
[21] "TCGA-OR-A5JZ" "TCGA-OR-A5KZ" "TCGA-PK-A5HB" "TCGA-OR-A5LK" "TCGA-OR-A5JJ"
[26] "TCGA-OR-A5JR" "TCGA-OR-A5JK" "TCGA-OR-A5J9" "TCGA-OU-A5PI" "TCGA-OR-A5L3"
[31] "TCGA-OR-A5J1" "TCGA-OR-A5KO" "TCGA-OR-A5JI" "TCGA-OR-A5JE" "TCGA-OR-A5JC"
[36] "TCGA-OR-A5JT" "TCGA-OR-A5LE" "TCGA-OR-A5L9" "TCGA-OR-A5K3" "TCGA-OR-A5J6"
[41] "TCGA-OR-A5LH" "TCGA-OR-A5K5" "TCGA-OR-A5JB" "TCGA-OR-A5LD" "TCGA-OR-A5K9"
[46] "TCGA-OR-A5JF" "TCGA-OR-A5KT" "TCGA-OR-A5J3" "TCGA-OR-A5LP" "TCGA-OR-A5LM"
[51] "TCGA-OR-A5LA" "TCGA-OR-A5K0" "TCGA-OR-A5L6" "TCGA-OR-A5LS" "TCGA-OR-A5LC"
[56] "TCGA-OR-A5K1" "TCGA-OR-A5LG" "TCGA-OR-A5LB" "TCGA-OR-A5JM" "TCGA-OR-A5LJ"
[61] "TCGA-OR-A5LF" "TCGA-OR-A5KB" "TCGA-OR-A5JH" "TCGA-PK-A5HC" "TCGA-OR-A5L2"
[66] "TCGA-P6-A5OG" "TCGA-P6-A5OH" "TCGA-OR-A5KQ" "TCGA-OR-A5LI" "TCGA-OR-A5KS"
[71] "TCGA-OR-A5L1" "TCGA-OR-A5KP" "TCGA-OR-A5JU" "TCGA-OR-A5JL" "TCGA-OR-A5LL"
[76] "TCGA-OR-A5J7" "TCGA-OR-A5J8" "TCGA-OR-A5JP" "TCGA-OR-A5LT" "TCGA-OR-A5JQ"
[81] "TCGA-OR-A5K8" "TCGA-OR-A5LO" "TCGA-OR-A5KV" "TCGA-OR-A5JD" "TCGA-OR-A5K6"
[86] "TCGA-OR-A5L5" "TCGA-OR-A5KX" "TCGA-OR-A5JY" "TCGA-OR-A5JV" "TCGA-OR-A5KU"
[91] "TCGA-PK-A5H8" "TCGA-OR-A5JX"
```

```
idx <- match(substr(coldata$barcode[76], 1,12),
             substr(coldataACC_clones$barcode, 1,12))

coldata$gender[76] <- ACC_clinical$gender[idx]
coldata$tumor_stage[76] <- "stage ii"
coldata$vital_status[76] <- ACC_clinical$vital_status[idx]
```

```
# -- cores dos metadados
col.bin.steroid <- c("Steroid_High"="black", "Steroid_Low"="gray")
col.bin.gender <- c("female"="grey", "male"="black")
col.bin.cortisol <- c("Cortisol"="black","No"="grey")
col.bin.vital_status <- c("Alive"="grey", "Dead"="black")

col.stage <- c("stage i"= "#bdbdbd", "stage ii"="#969696",
               "stage iii"="#525252","stage iv"="#000000",
               "not reported"="#ffffff")
col.imm_sub <- c("C1"="#ffffff", "C2"="#d9d9d9", "C3"="#969696",
                 "C4"="#525252", "C5"="#252525", "C6"="#000000")
col.oth_horm <- c("Mineralcorticoids"="#000000", "Sexual"="#525252", "No"="#bdbdbd")

col.seq_greys.read <- colorRamp2(breaks = seq(21000000, 83000000, length.out=9),
                                 colors = brewer.pal(9,"Greys"))
col.seq_greys <- colorRamp2(breaks = seq(0, 4, length.out=9),
                            colors = brewer.pal(9,"Greys"))

col.ha <- HeatmapAnnotation(steroid = coldata$steroid,
                            immune_subtype = coldata$immune.subtype,
                            other.hormones = coldata$other.hormones,
                            cortisol.excess = coldata$cortisol.excess,
                            tumor_stage = coldata$tumor_stage,
                            gender = coldata$gender,
                            vital_status = coldata$vital_status,
                            count_TCR_log10 = log10(coldata$TCR + 1),
                            count_BCR_log10 = log10(coldata$BCR + 1),
                            count_reads = coldata$reads,
                            col = list(steroid=col.bin.steroid,
                                       immune_subtype = col.imm_sub,
                                       other.hormones = col.oth_horm,
                                       cortisol.excess = col.bin.cortisol,
                                       tumor_stage = col.stage,
                                       gender = col.bin.gender,
                                       vital_status = col.bin.vital_status,
                                       count_TCR_log10 = col.seq_greys,
                                       count_BCR_log10 = col.seq_greys,
                                       count_reads = col.seq_greys.read))
```

```
data <- t(data)
data <- data[c(5,1,2,6,3,4,7),]
```

```r
row.ha <- rowAnnotation(type=c(rep("B",3), rep("T",4)))
df <- as.data.frame(data)
df$cell_type <- c(rep("B",3), rep("T",4))
```

```r
g_counts <- Heatmap(data,
                    split = df$cell_type,
                    top_annotation = col.ha,
                    show_column_names = FALSE,
                    cluster_columns = FALSE,
                    cluster_rows = F,
                    col=colorRamp2(breaks = seq(-2,2, length.out=9),
                                   colors = rev(brewer.pal(9,"BrBG"))),
                    row_title_gp = gpar(fontsize=10),
                    row_title_side = "left",
                    row_names_side = "right",
                    row_names_gp = gpar(fontsize=10))
```

Expressão

```r
data <- mat_data_all[, 1:7]

for (i in rownames(data)) {
  data[i,] <- data[i,] / coldata$reads[coldata$sample_id == i]
}

data <- log2(data + 1)
data <- scale(data)

data <- t(data)
data <- data[c(5,1,2,6,3,4,7),]
```

```r
g_expression <- Heatmap(data,
                        split = df$cell_type,
                        top_annotation = col.ha,
                        show_column_names = FALSE,
                        cluster_columns = FALSE,
                        cluster_rows = F,
                        col=colorRamp2(breaks = seq(-2,2, length.out=9),
                                       colors = rev(brewer.pal(9,"BrBG"))),
                        row_title_gp = gpar(fontsize=10),
                        row_title_side = "left",
                        row_names_side = "right",
                        row_names_gp = gpar(fontsize=10))
```

Richness

```
dir_metrics <- "../01_stats_TRUST4/results_pipeline_clones"

arquivos <- list.files(dir_metrics, pattern = "\\.tsv$", full.names = TRUE)

data_metrics <- data.frame()

for (arquivo in arquivos) {
  dados <- read.table(arquivo, header = TRUE, sep = "\t", stringsAsFactors = FALSE)
  dados_richnes <- data.frame(t(dados$Richness))
  rownames(dados_richnes) <- gsub("_report.tsv", "", basename(arquivo))

  data_metrics <- rbind(data_metrics, dados_richnes)
}

colnames(data_metrics) <- dados$chain

mat_add <- data.frame(IGH = rep(0,8),
                      IGK = rep(0,8),
                      IGL = rep(0,8),
                      TRA = rep(0,8),
                      TRB = rep(0,8),
                      TRG = rep(0,8),
                      TRD = rep(0,8))

rownames(mat_add) <- colnames(data)[colnames(data) %in% rownames(data_metrics) == FALSE]

data_metrics <- rbind(data_metrics, mat_add)

data_metrics <- as.matrix(data_metrics)

data_metrics <- log2(data_metrics + 1)
data_metrics <- scale(data_metrics)
```

```r
data_metrics <- t(data_metrics)

df <- as.data.frame(data_metrics)
df$cell_type <- c(rep("B",3), rep("T",4))

idx <- match(colnames(data), colnames(data_metrics))
data_metrics <- data_metrics[,idx]
```

```r
g_richness <- Heatmap(data_metrics,
                      split = df$cell_type,
                      top_annotation = col.ha,
                      show_column_names = FALSE,
                      cluster_columns = FALSE,
                      cluster_rows = F,
                      col=colorRamp2(breaks = seq(-2,2, length.out=9),
                                     colors = rev(brewer.pal(9,"BrBG"))),
                      row_title_gp = gpar(fontsize=10),
                      row_title_side = "left",
                      row_names_side = "right",
                      row_names_gp = gpar(fontsize=10))
```

CPK

```r
data_metrics <- data.frame()

for (arquivo in arquivos) {
  dados <- read.table(arquivo, header = TRUE, sep = "\t", stringsAsFactors = FALSE)
  dados_cpk <- data.frame(t(dados$CPK))
  rownames(dados_cpk) <- gsub("_report.tsv", "", basename(arquivo))

  data_metrics <- rbind(data_metrics, dados_cpk)
}

colnames(data_metrics) <- dados$chain

data_metrics[is.na(data_metrics)] <- 0

mat_add <- data.frame(IGH = rep(0,8),
                      IGK = rep(0,8),
                         IGL = rep(0,8),
                         TRA = rep(0,8),
                         TRB = rep(0,8),
                         TRG = rep(0,8),
                         TRD = rep(0,8))

rownames(mat_add) <- colnames(data)[colnames(data) %in% rownames(data_metrics) == FALSE]

data_metrics <- rbind(data_metrics, mat_add)

data_metrics <- as.matrix(data_metrics)

data_metrics <- log2(data_metrics + 1)
data_metrics <- scale(data_metrics)

data_metrics <- t(data_metrics)

idx <- match(colnames(data), colnames(data_metrics))
data_metrics <- data_metrics[,idx]
```

```
g_cpk <- Heatmap(data_metrics,
                 split = df$cell_type,
                 top_annotation = col.ha,
                 show_column_names = FALSE,
                 cluster_columns = FALSE,
                 cluster_rows = F,
                 col=colorRamp2(breaks = seq(-2,2, length.out=9),
                                colors = rev(brewer.pal(9,"BrBG"))),
                 row_title_gp = gpar(fontsize=10),
                 row_title_side = "left",
                 row_names_side = "right",
                 row_names_gp = gpar(fontsize=10))
```

Entropia

```
data_metrics <- data.frame()

for (arquivo in arquivos) {
  dados <- read.table(arquivo, header = TRUE, sep = "\t", stringsAsFactors = FALSE)
```

```r
  dados_entropy <- data.frame(t(dados$Entropy))
  rownames(dados_entropy) <- gsub("_report.tsv", "", basename(arquivo))

  data_metrics <- rbind(data_metrics, dados_entropy)
}

colnames(data_metrics) <- dados$chain

mat_add <- data.frame(IGH = rep(0,8),
                      IGK = rep(0,8),
                      IGL = rep(0,8),
                      TRA = rep(0,8),
                      TRB = rep(0,8),
                      TRG = rep(0,8),
                      TRD = rep(0,8))

rownames(mat_add) <- colnames(data)[colnames(data) %in% rownames(data_metrics) == FALSE]

data_metrics <- rbind(data_metrics, mat_add)

data_metrics <- as.matrix(data_metrics)

data_metrics <- log2(data_metrics + 1)
data_metrics <- scale(data_metrics)

data_metrics <- t(data_metrics)

idx <- match(colnames(data), colnames(data_metrics))
data_metrics <- data_metrics[,idx]
```

```r
g_entropy <- Heatmap(data_metrics,
                     split = df$cell_type,
                     top_annotation = col.ha,
                     show_column_names = FALSE,
                     cluster_columns = FALSE,
                     cluster_rows = F,
                     col=colorRamp2(breaks = seq(-2,2, length.out=9),
                                    colors = rev(brewer.pal(9,"BrBG"))),
                     row_title_gp = gpar(fontsize=10),
                     row_title_side = "left",
                     row_names_side = "right",
                          row_names_gp = gpar(fontsize=10))
```

Clonality

```
data_metrics <- data.frame()

for (arquivo in arquivos) {
  dados <- read.table(arquivo, header = TRUE, sep = "\t", stringsAsFactors = FALSE)
  dados_clonality <- data.frame(t(dados$Clonality))
  rownames(dados_clonality) <- gsub("_report.tsv", "", basename(arquivo))

  data_metrics <- rbind(data_metrics, dados_clonality)
}

colnames(data_metrics) <- dados$chain

data_metrics[is.na(data_metrics)] <- 0

mat_add <- data.frame(IGH = rep(0,8),
                      IGK = rep(0,8),
                      IGL = rep(0,8),
                      TRA = rep(0,8),
                      TRB = rep(0,8),
                      TRG = rep(0,8),
                      TRD = rep(0,8))

rownames(mat_add) <- colnames(data)[colnames(data) %in% rownames(data_metrics) == FALSE]

data_metrics <- rbind(data_metrics, mat_add)

data_metrics <- as.matrix(data_metrics)

data_metrics <- log2(data_metrics + 1)
data_metrics <- scale(data_metrics)

data_metrics <- t(data_metrics)

idx <- match(colnames(data), colnames(data_metrics))
data_metrics <- data_metrics[,idx]
```

```r
g_clonality <- Heatmap(data_metrics,
                    split = df$cell_type,
                    top_annotation = col.ha,
                    show_column_names = FALSE,
                    cluster_columns = FALSE,
                    cluster_rows = F,
                    col=colorRamp2(breaks = seq(-2,2, length.out=9),
                                   colors = rev(brewer.pal(9,"BrBG"))),
                    row_title_gp = gpar(fontsize=10),
                    row_title_side = "left",
                    row_names_side = "right",
                    row_names_gp = gpar(fontsize=10))
```