HeatMap

Asad

2023-05-15

```
library(ComplexHeatmap)
```

```
## Loading required package: grid
```

```
## ========================================
## ComplexHeatmap version 2.10.0
## 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:
## 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))
## ========================================
```

```
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
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.4.2     v purrr   1.0.1
## v tibble  3.2.1     v stringr 1.5.0
## v tidyr   1.3.0     v forcats 1.0.0
## v readr   2.1.2

## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(AnnotationDbi)
```

```
## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:purrr':
##
##     reduce

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## The following object is masked from 'package:grDevices':
##
##     windows

##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(org.Hs.eg.db)
```

```
##
```

```
library(circlize)
```

```
## ========================================
## circlize version 0.4.15
## 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))
## ========================================
```

```
library(magick)
```

```
## Linking to ImageMagick 6.9.12.3
## Enabled features: cairo, freetype, fftw, ghostscript, heic, lcms, pango, raw, rsvg, webp
## Disabled features: fontconfig, x11
```

```
library(RColorBrewer)
setwd('E:/R-Programming-Practices/Data Visualization/Heatmap')
```

Heatmap from gene expression data (like output from DESeq2)

#Read count data

```
norm.counts<- read.csv('Normalized read counts.csv', row.names = 1)
z.data<- t(apply(norm.counts, 1, scale)) #Convert to z scores
colnames(z.data)<-rep(c('Healthy', 'COVID-19'), c(9,36)) #Change column names
z.data<- data.frame(z.data)
```

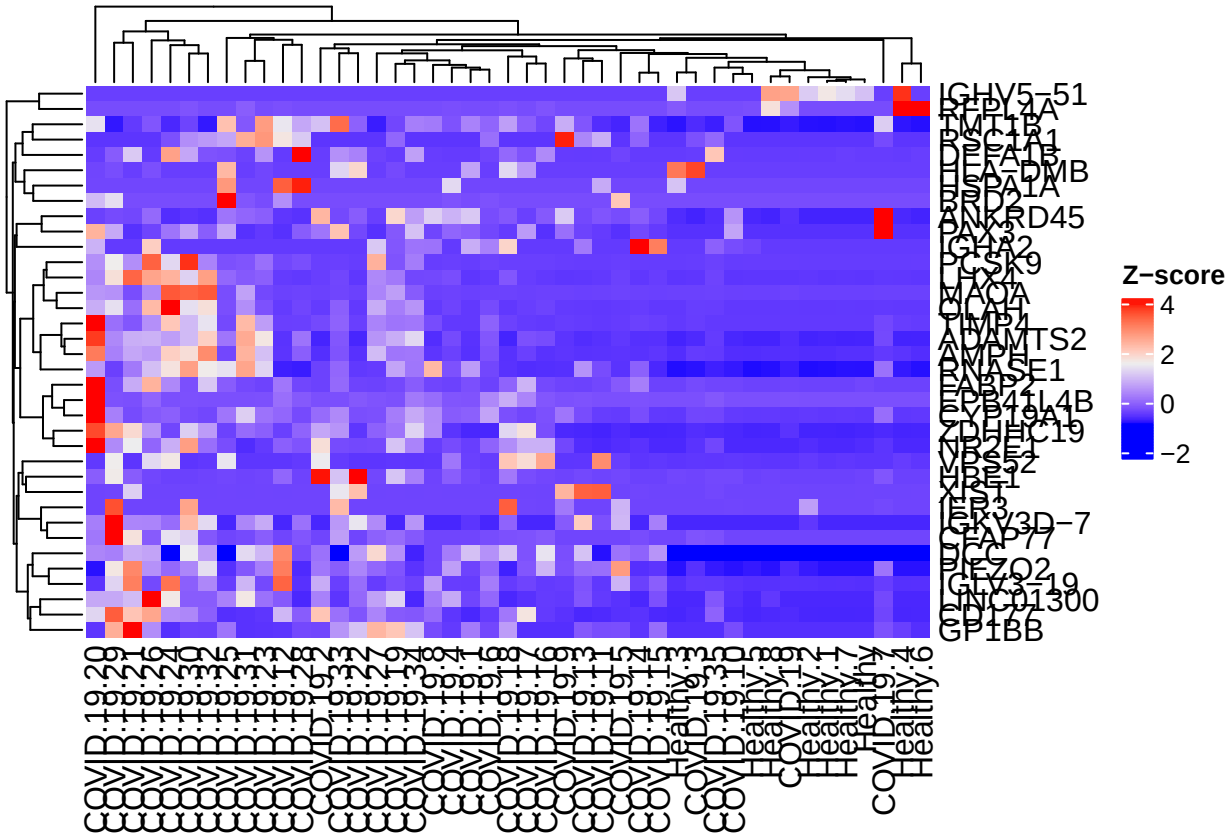
Read DEGs matrix and stratify

```
sig.genes<- read.csv('Significant_DEGs.csv', row.names = 1)
sig.genes<- sig.genes[order(sig.genes$padj),]
sig.genes_final<- filter(sig.genes, sig.genes$padj<0.001 & abs(sig.genes$log2FoldChange)>=5)
data.final<- z.data[row.names(sig.genes_final),]
data.final<- na.omit(data.final)
```

Plot heatmap

```
Heatmap(data.final, cluster_rows = T, cluster_columns = T,
                  column_labels = colnames(data.final), row_labels =
                  sig.genes[rownames(data.final),]$symbol, name = 'Z-score')
```

```
## Warning: The input is a data frame, convert it to a matrix.
```



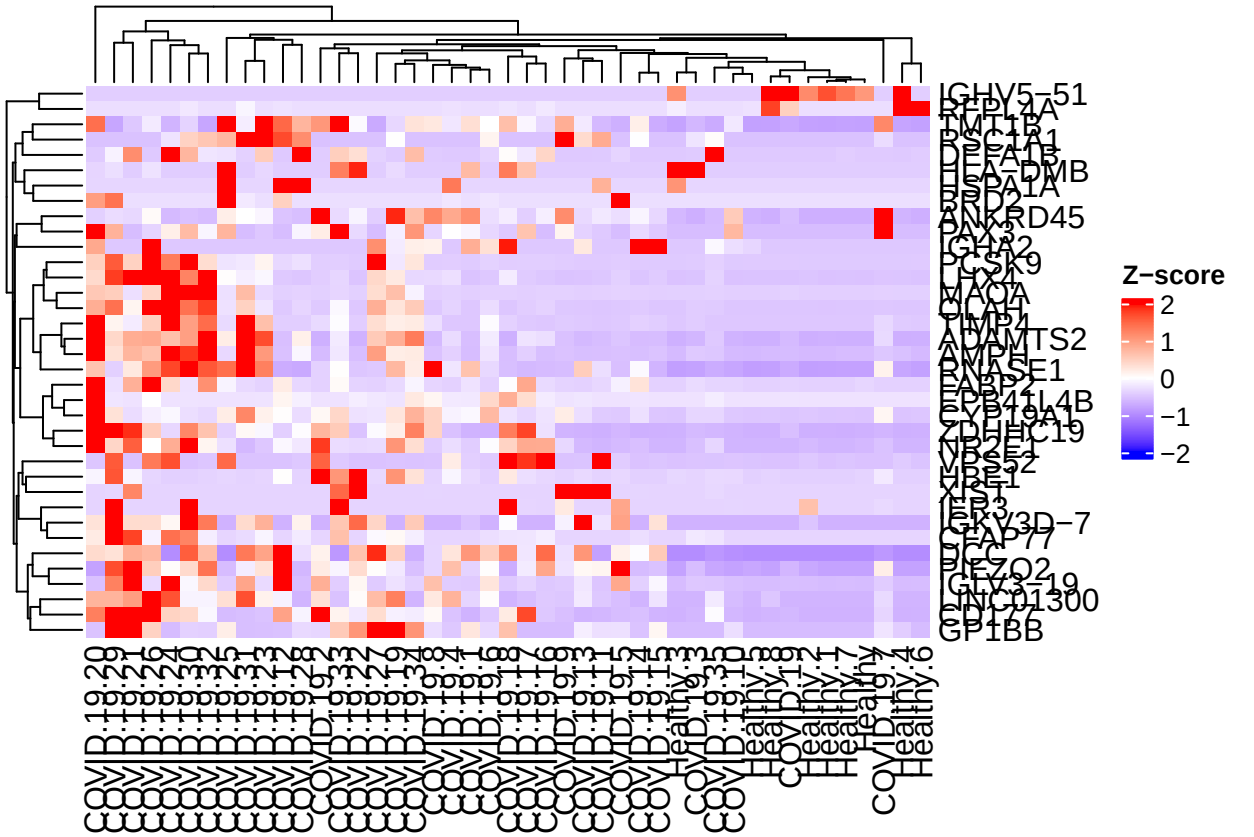
Change color

```
col_fun = colorRamp2(c(-2, 0, 2), c("blue", "white", "red"))
col_fun(seq(-2, 4))
```

```
## [1] "#0000FFFF" "#B38BFFFF" "#FFFFFFFF" "#FF9E81FF" "#FF0000FF" "#FF0000FF"
## [7] "#FF0000FF"
```

```
Heatmap(data.final, cluster_rows = T, cluster_columns = T,
        column_labels = colnames(data.final), row_labels =
          sig.genes[rownames(data.final),]$symbol, name = 'Z-score', col= col_fun)
```

```
## Warning: The input is a data frame, convert it to a matrix.
```

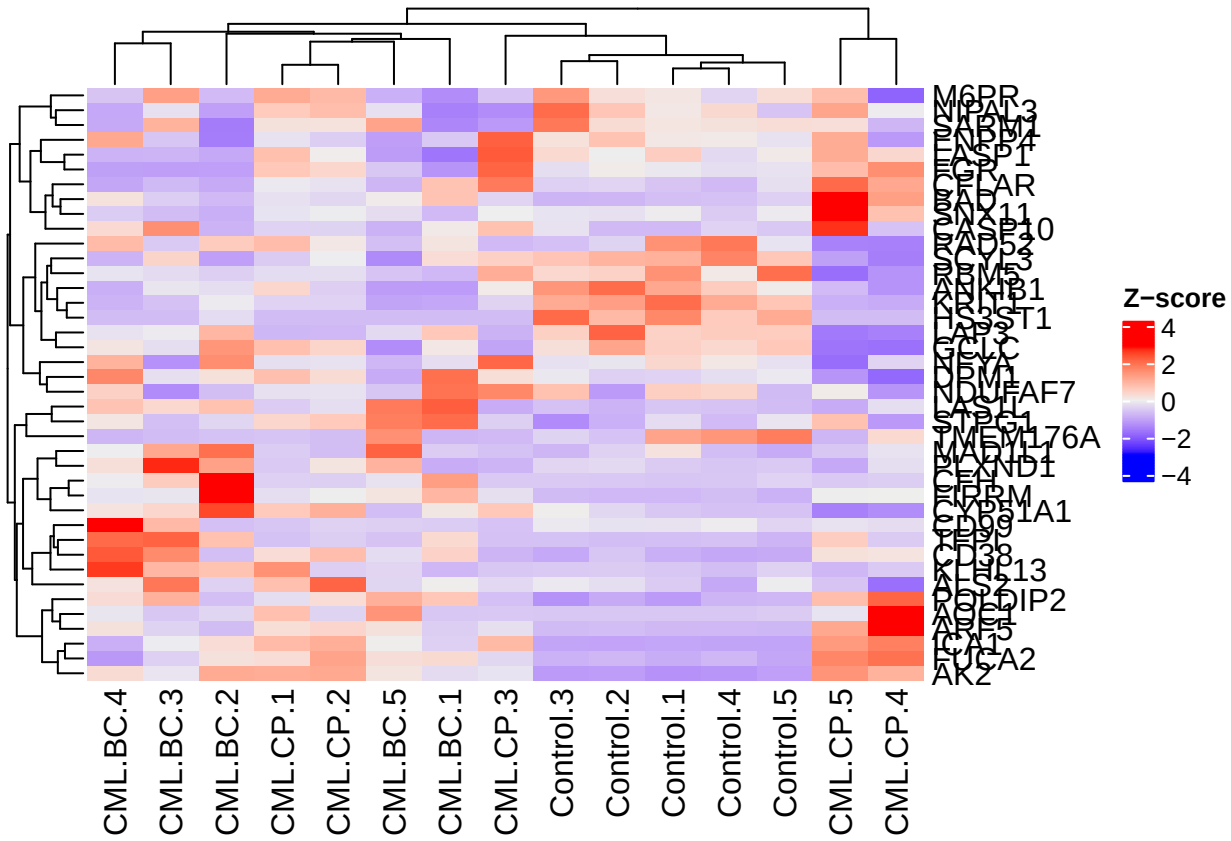


Heatmap from random data

```
data2<- read.csv('Voom_CPM.csv', row.names = 1)
symbol<- mapIds(org.Hs.eg.db, keys = rownames(data2),
                keytype = 'ENSEMBL', column = 'SYMBOL')
```

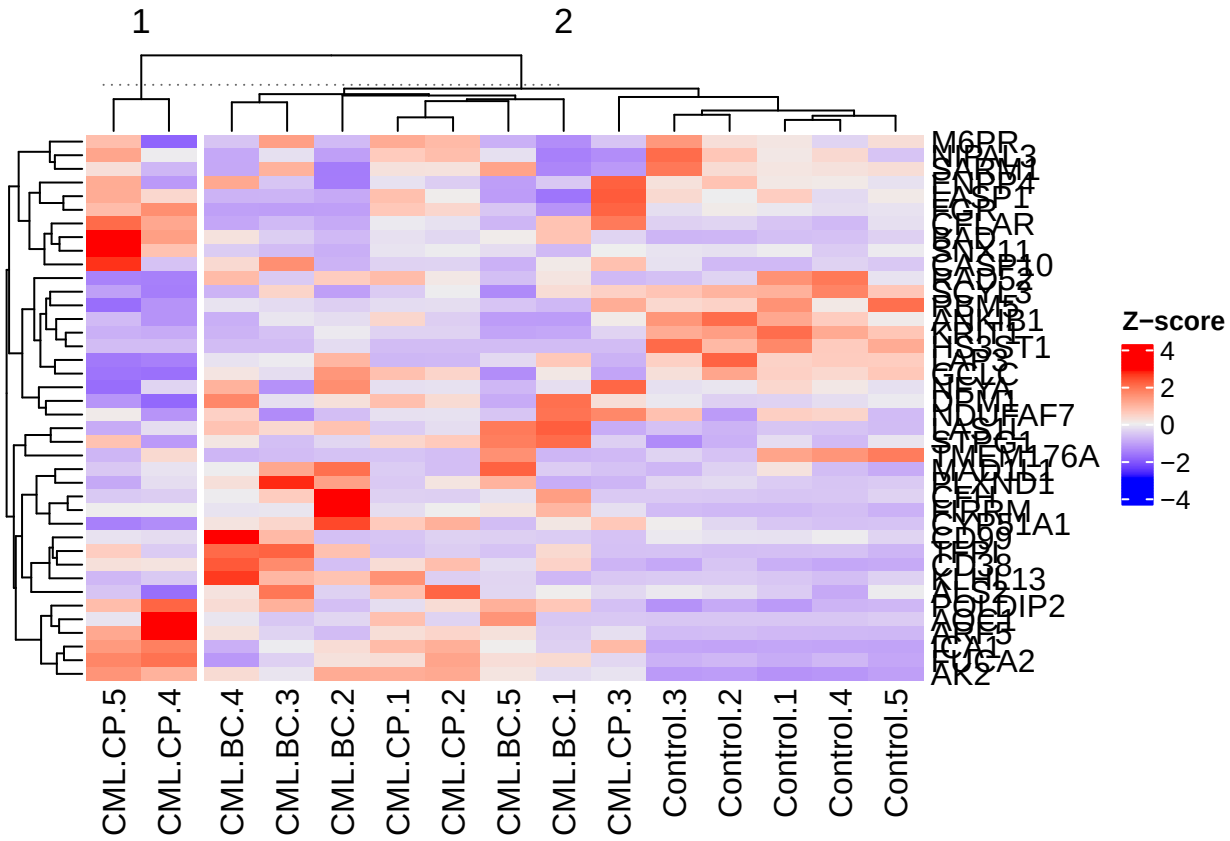
```
## 'select()' returned 1:many mapping between keys and columns
```

```
symbol<- as.data.frame(symbol)

data2.z<- t(apply(data2,1,scale))
colnames(data2.z)<- colnames(data2)
data2.z<- data2.z[1:40,]

Heatmap(data2.z, row_labels = symbol[rownames(data2.z),], name = 'Z-score')
```



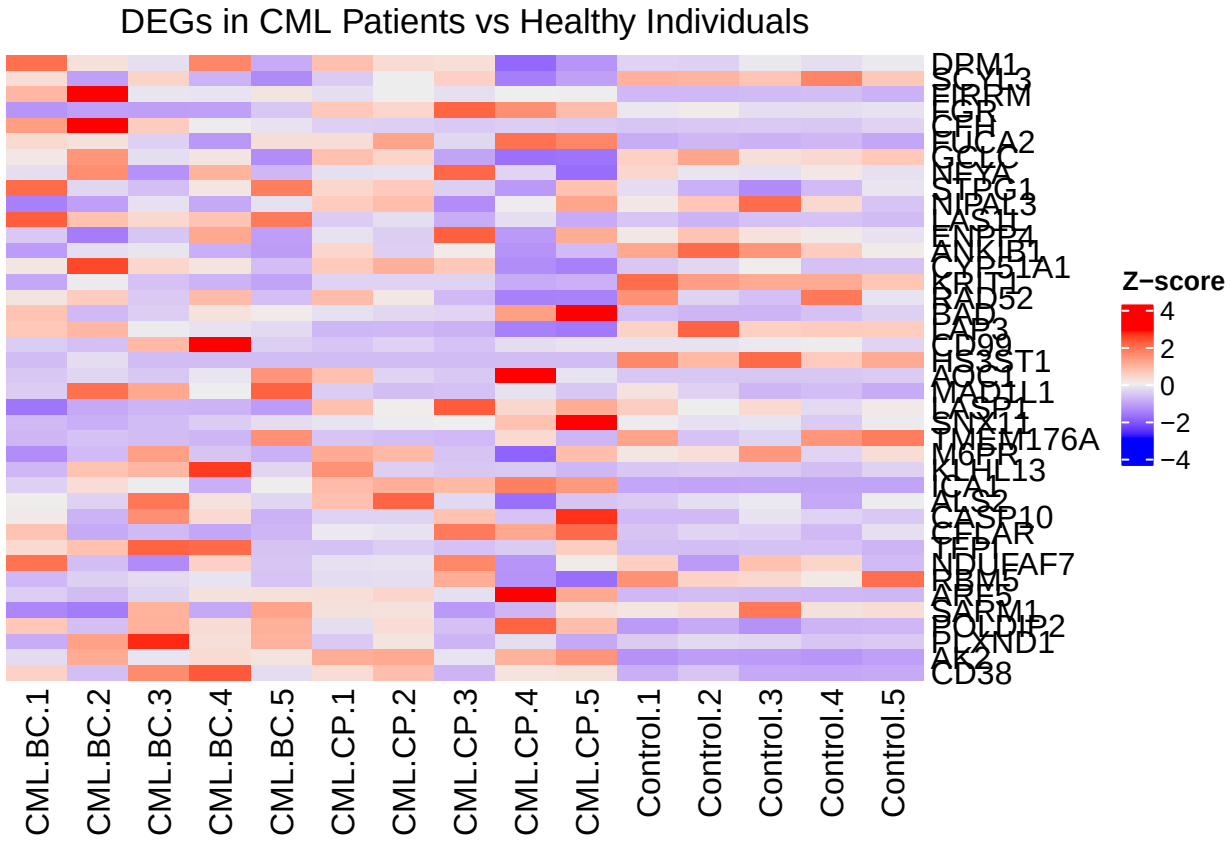
Split heatmap based on clustering

```
Heatmap(data2.z, row_labels = symbol[rownames(data2.z),], name = 'Z-score',
        column_km = 2)
```



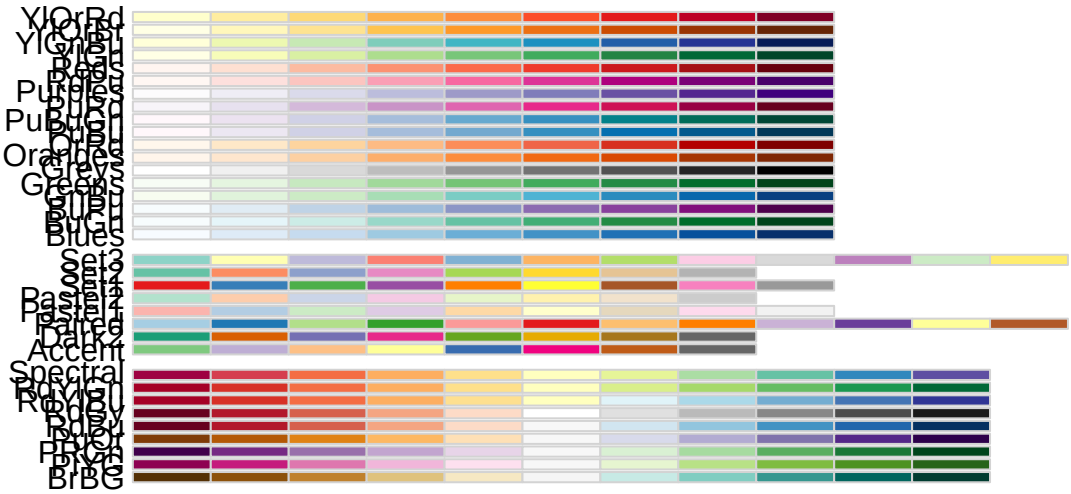
Reorder row and column names (Dendogram will not work)

```
Heatmap(data2.z, row_labels = symbol[rownames(data2.z),], name = 'Z-score',
   row_order = sort(rownames(data2.z)), column_order = sort(colnames(data2.z)),
              column_title = "DEGs in CML Patients vs Healthy Individuals")
```



Adding annotation bar

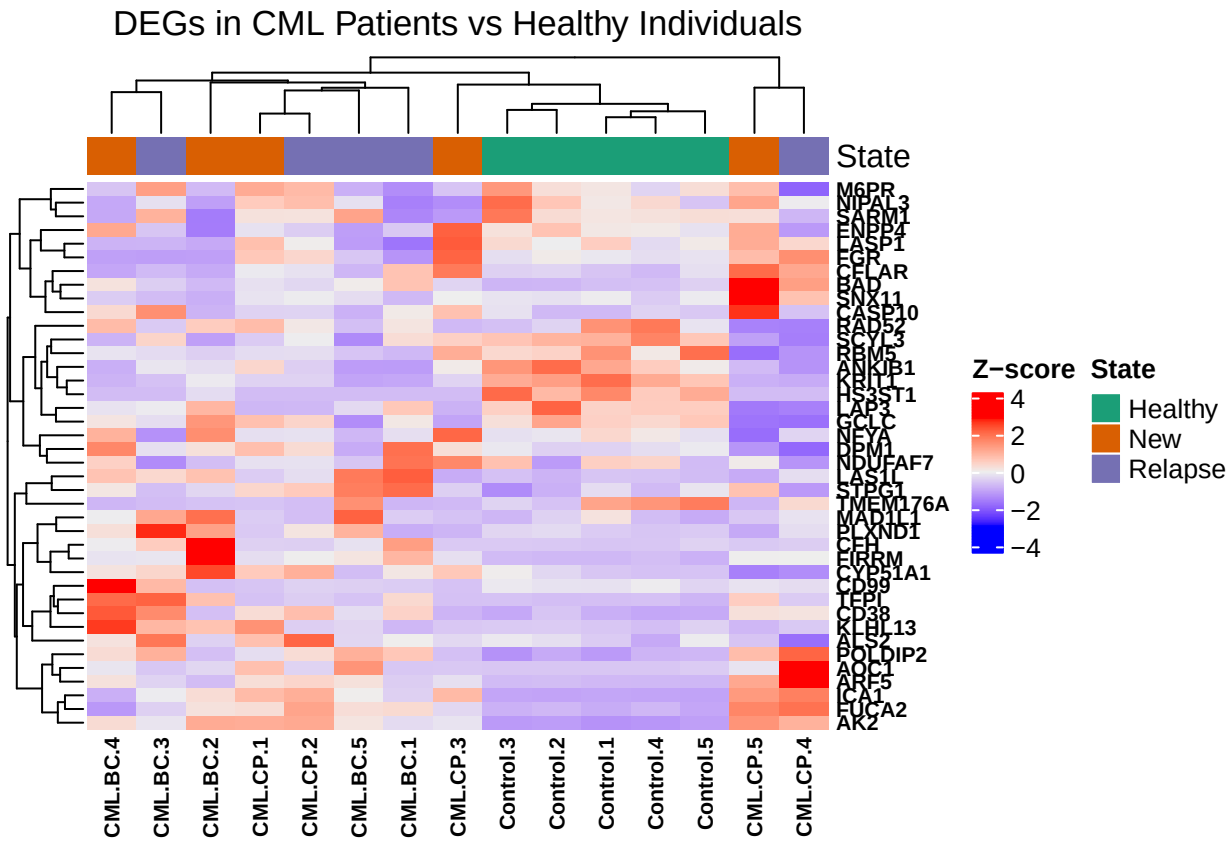
```
#Prepare data
Char1<- c(rep('Healthy', 5), 'New', 'Relapse', 'New', 'Relapse','New', 'Relapse',
          'New', 'Relapse','New', 'Relapse')
Char2<- colnames(data2)

#Adjust color
display.brewer.all()
```




```
brewer.pal(n=7, "Dark2")
```

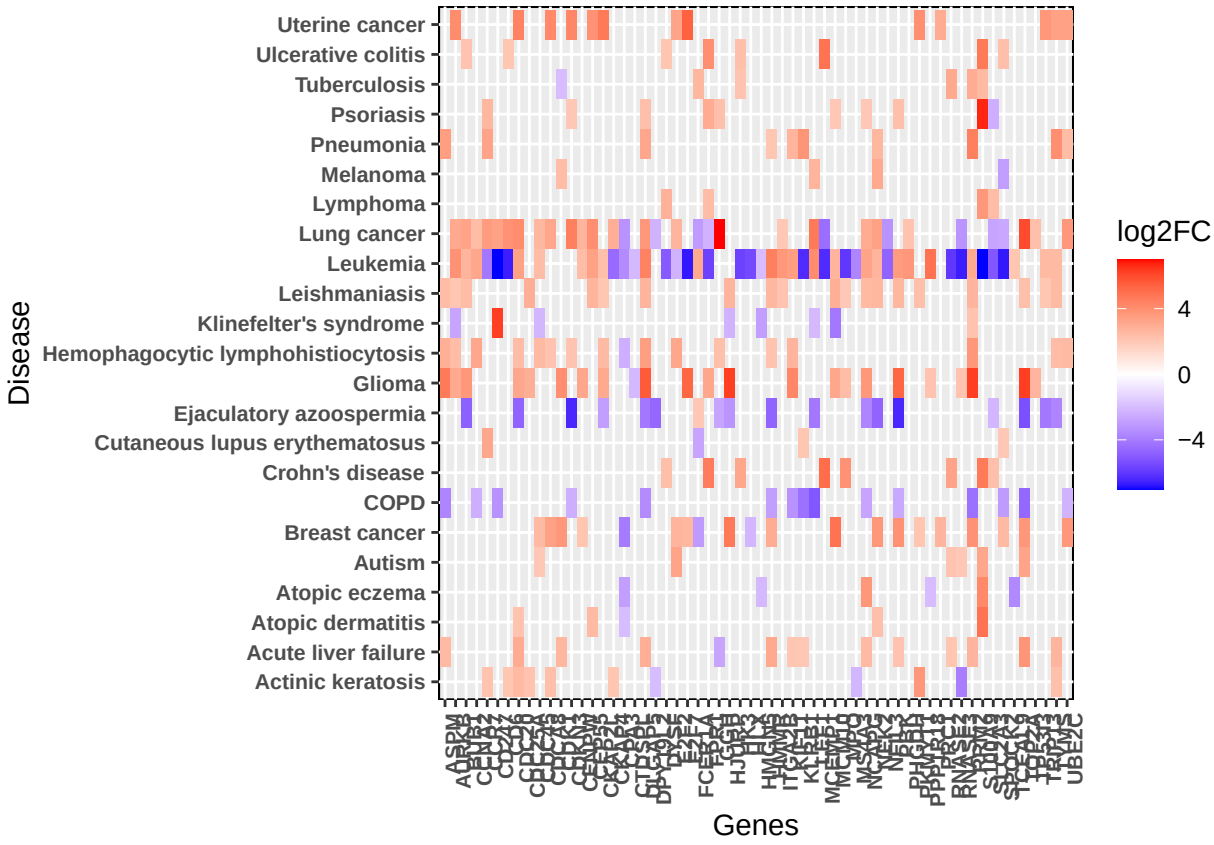
```
## [1] "#1B9E77" "#D95F02" "#7570B3" "#E7298A" "#66A61E" "#E6AB02" "#A6761D"
```

```
ann<- data.frame(Char2, Char1)
colnames(ann)<- c('Study', 'State')
rownames(ann)<- ann[,1]
detach(package:org.Hs.eg.db, unload = TRUE)
detach(package:AnnotationDbi, unload = TRUE)
ann2<- ann %>% select('State')
col<- list('State'=c('Healthy'='#1B9E77', 'New'='#D95F02', 'Relapse'='#7570B3'))

#Adjust column annotation
colAnn <- HeatmapAnnotation(df = ann2,name = 'Disease State',
                            which = 'col',
                            col = col,
                            annotation_width = unit(c(1, 4), 'cm'),
                            gap = unit(1, 'mm'))
#Heatmap
Heatmap(data2.z, top_annotation = colAnn, name='Z-score', row_labels =
          symbol[rownames(data2.z),],
        column_names_gp = grid::gpar(fontsize = 8, fontface='bold'),
        row_names_gp = grid::gpar(fontsize = 8, fontface='bold'),
        column_title = 'DEGs in CML Patients vs Healthy Individuals')
```



Heatmap from one/two column data

```
data3<- read.csv('Expression Atlas.csv')

plot<- ggplot(data3, aes(x=Genes, y= Disease, fill=log2FC))+geom_tile()+
        scale_fill_gradient2(high = 'red', mid='white', low = 'blue') +
        theme(axis.text.x = element_text(angle = 90, size = 8, face =
        'bold', hjust = 1), axis.text.y = element_text(size = 8, face =
        'bold'), panel.background = element_rect(colour = 'black'),
         panel.grid.major = element_line(color = 'white'))
plot
```


Uterine cancer
Ulcerative colitis
Tuberculosis
Psoriasis
Pneumonia
Melanoma
Lymphoma
Lung cancer
Leukemia
Leishmaniasis
Klinefelter's syndrome
Hemophagocytic lymphohistiocytosis
Glioma
Ejaculatory azoospermia
Cutaneous lupus erythematosus
Crohn's disease
COPD
Breast cancer
Autism
Atopic eczema
Atopic dermatitis
Acute liver failure
Actinic keratosis
Disease
Genes
log2FC
4
0
−4