Gene Set Interactions and Z-matrix

HBG

8/10/2019

R Markdown

Using 50 hallmark gene sets from MSigDB (molecular signature data base), and based on the protein-protein interaction network, the strength of interactions between each pair of gene sets is defined as the total number of interactions (edges) between the proteins (nodes) of both sets. Z-scores are evaluated via comparitons to null models of the original network; 10k null models have been used. A positive Z-score indicates an enriched interacton between both sets, whereas a negative Z-score indicates a suppressed interaction between them.

```
list.file <- read.csv("../../MSigDB.go.pathway/list", header=F, stringsAsFactors = F)
#A list of all 50 hallmark gene sets
hallmark.name <- list.file$V1
hallmark.dim <- length(hallmark.name) # 50 sets

hallmark.name # prints all hallmark set names here
```

```
##  [1] "HALLMARK_ADIPOGENESIS"
##  [2] "HALLMARK_ALLOGRAFT_REJECTION"
##  [3] "HALLMARK_ANDROGEN_RESPONSE"
##  [4] "HALLMARK_ANGIOGENESIS"
##  [5] "HALLMARK_APICAL_JUNCTION"
##  [6] "HALLMARK_APICAL_SURFACE"
##  [7] "HALLMARK_APOPTOSIS"
##  [8] "HALLMARK_BILE_ACID_METABOLISM"
##  [9] "HALLMARK_CHOLESTEROL_HOMEOSTASIS"
## [10] "HALLMARK_COAGULATION"
## [11] "HALLMARK_COMPLEMENT"
## [12] "HALLMARK_DNA_REPAIR"
## [13] "HALLMARK_E2F_TARGETS"
## [14] "HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION"
## [15] "HALLMARK_ESTROGEN_RESPONSE_EARLY"
## [16] "HALLMARK_ESTROGEN_RESPONSE_LATE"
## [17] "HALLMARK_FATTY_ACID_METABOLISM"
## [18] "HALLMARK_G2M_CHECKPOINT"
## [19] "HALLMARK_GLYCOLYSIS"
## [20] "HALLMARK_HEDGEHOG_SIGNALING"
## [21] "HALLMARK_HEME_METABOLISM"
## [22] "HALLMARK_HYPOXIA"
## [23] "HALLMARK_IL2_STAT5_SIGNALING"
## [24] "HALLMARK_IL6_JAK_STAT3_SIGNALING"
## [25] "HALLMARK_INFLAMMATORY_RESPONSE"
## [26] "HALLMARK_INTERFERON_ALPHA_RESPONSE"
## [27] "HALLMARK_INTERFERON_GAMMA_RESPONSE"
## [28] "HALLMARK_KRAS_SIGNALING_DN"
## [29] "HALLMARK_KRAS_SIGNALING_UP"
## [30] "HALLMARK_MITOTIC_SPINDLE"
## [31] "HALLMARK_MTORC1_SIGNALING"
## [32] "HALLMARK_MYC_TARGETS_V1"
```

```
## [33] "HALLMARK_MYC_TARGETS_V2"
## [34] "HALLMARK_MYOGENESIS"
## [35] "HALLMARK_NOTCH_SIGNALING"
## [36] "HALLMARK_OXIDATIVE_PHOSPHORYLATION"
## [37] "HALLMARK_P53_PATHWAY"
## [38] "HALLMARK_PANCREAS_BETA_CELLS"
## [39] "HALLMARK_PEROXISOME"
## [40] "HALLMARK_PI3K_AKT_MTOR_SIGNALING"
## [41] "HALLMARK_PROTEIN_SECRETION"
## [42] "HALLMARK_REACTIVE_OXIGEN_SPECIES_PATHWAY"
## [43] "HALLMARK_SPERMATOGENESIS"
## [44] "HALLMARK_TGF_BETA_SIGNALING"
## [45] "HALLMARK_TNFA_SIGNALING_VIA_NFKB"
## [46] "HALLMARK_UNFOLDED_PROTEIN_RESPONSE"
## [47] "HALLMARK_UV_RESPONSE_DN"
## [48] "HALLMARK_UV_RESPONSE_UP"
## [49] "HALLMARK_WNT_BETA_CATENIN_SIGNALING"
## [50] "HALLMARK_XENOBIOTIC_METABOLISM"
```

```
pin <- read.csv("../../human.pin.csv", header=T, stringsAsFactors = F)
# the PIN in pairwise format; other networks work in the same way
geneA <- pin$geneA
geneB <- pin$geneB

hallmark.matrix <- matrix(0, nrow=50, ncol=50)
# initiate an emply matrix, which will record interaction strengths among all 50 sets
for (i in 1:50) {
  for (j in 1:50) {
    hallmarkA <- paste("../../MSigDB.go.pathway/", hallmark.name[i], "/", hallmark.name[i], ".csv", sep=
    fileA <- read.csv(hallmarkA, header=T, stringsAsFactors=F)
    hallmarkB <- paste("../../MSigDB.go.pathway/", hallmark.name[j], "/", hallmark.name[j], ".csv", sep=
    fileB <- read.csv(hallmarkB, header=T, stringsAsFactors=F)
    A.genes <- fileA$gene
    B.genes <- fileB$gene
    #number of interactions between hallmark set A and set B
    AB.int <- length(which(((geneA %in% A.genes) & (geneB %in% B.genes)) |
                           (geneA %in% B.genes) & (geneB %in% A.genes)))
    hallmark.matrix[i, j] = hallmark.matrix[i, j] + AB.int
  }
}

write.table(hallmark.matrix, file="hhi.csv", sep=",", col.names=F,
            row.names=F, quote=F)
```

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 3.5.2
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
```

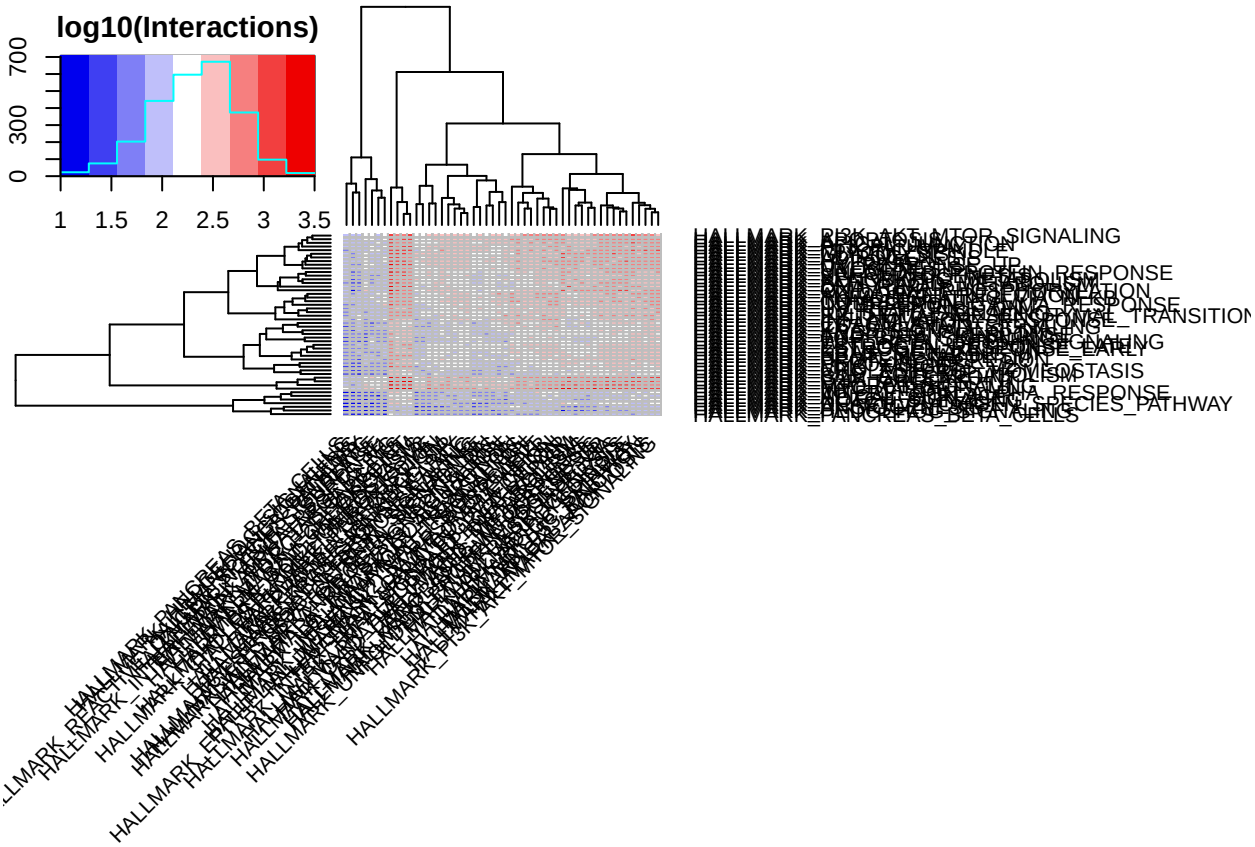
```
##
##     union
```

```
library(gplots)
```

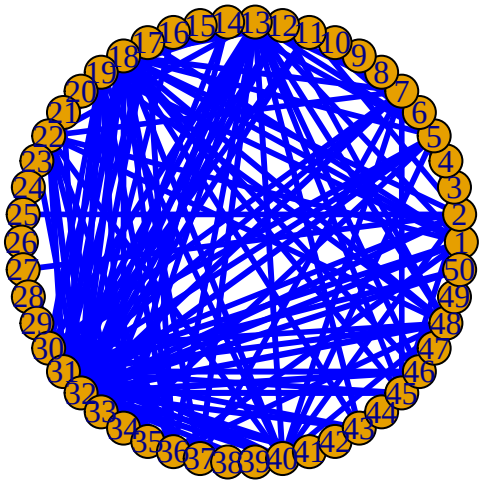
```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

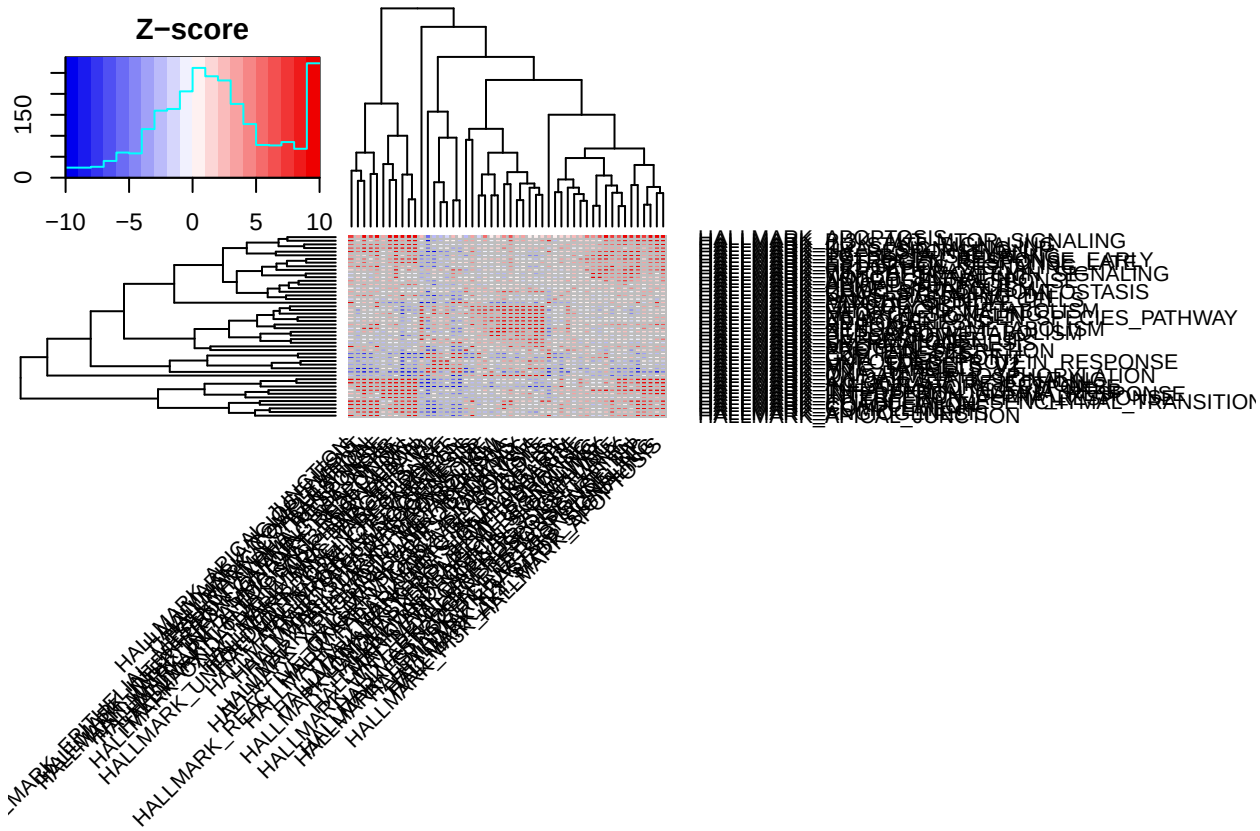
```
#the above librarys are used for network and heatmap, respectively
# A gene set network based on interaction strengths will be generated
hallmark.int <- log10(hallmark.matrix)
# a log10 scale used
colnames(hallmark.int) <- hallmark.name
rownames(hallmark.int) <- hallmark.name
# the intraction numbers range from 0 to 3103, in log10 scale it from 1 to 3.491782
my_palette <- colorRampPalette(c("blue2", "white", "red2"))(n = 9)
colors = c(seq(1,3.5,length=10))
#png("hhi.int.heatmap.png", width=12, height=11, res=600, units="in")
heatmap.2(hallmark.int, col=my_palette, trace='none', breaks=colors,
          key.xlab=NA, key.title="log10(Interactions)",
          key.ylab=NA, key.xtickfun = NULL, key.ytickfun = NULL,
          srtCol=45, adjCol=c(1,0), dendrogram = "both",
          margins=c(14,18.5), sepwidth=c(0.01,0.01), #symbreaks = TRUE,
          sepcolor="grey", colsep=1:hallmark.dim, rowsep=1:hallmark.dim)
```



```
#dev.off()

colnames(hallmark.matrix) <- c(1:50)
rownames(hallmark.matrix) <- c(1:50)
hallmark.net <- graph.adjacency(hallmark.matrix, mode="undirected", weighted = T, diag = F)
#apparently every two sets are connected, we only plot those with more than 600 connections
E(hallmark.net)$weight <- ifelse(abs(E(hallmark.net)$weight) > 600, abs(E(hallmark.net)$weight), 0)
plot.igraph(hallmark.net, vertex.label=V(hallmark.net)$name, layout=layout_in_circle,
            edge.color="blue", edge.width=log10(E(hallmark.net)$weight))
```



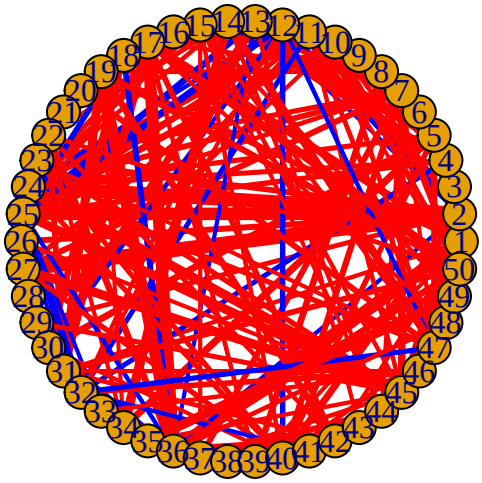

```
#the z-scores are calculated via comparisions with 10k null models

hhi.dat <- read.csv("hhi.z.csv", header=F, stringsAsFactors = F)
hhi.z <- matrix(unlist(hhi.dat), nrow=hallmark.dim, ncol=hallmark.dim)
colnames(hhi.z) <- hallmark.name
row.names(hhi.z) <- hallmark.name

my_palette <- colorRampPalette(c("blue2", "white", "red2"))(n = 20)
#colors= c(seq(-5,-1.5, length=10), seq(-1.49, 1.49, length=10), seq(1.5,5, length=10))
colors = c(seq(-10,10,length=21))

#png("hhi.z.heatmap.png", width=12, height=11, res=600, units="in")
heatmap.2(hhi.z, col=my_palette, trace='none', breaks=colors,
          key.xlab=NA, key.title="Z-score", key.ylab=NA, key.xtickfun = NULL, key.ytickfun = NULL,
          srtCol=45, adjCol=c(1,0), dendrogram = "both",
          margins=c(14,18.5), sepwidth=c(0.01,0.01), symbreaks = TRUE,
          sepcolor="grey", colsep=1:hallmark.dim, rowsep=1:hallmark.dim)
```




```
#dev.off()

hhi.z.mat <- as.matrix(hhi.z)
hhi.z.net <- graph.adjacency(hhi.z.mat, mode="undirected", weighted=T, diag=F)
summary(hhi.z.net)
```

```
## IGRAPH 63be128 UNW- 50 1225 --
## + attr: name (v/c), weight (e/n)
```

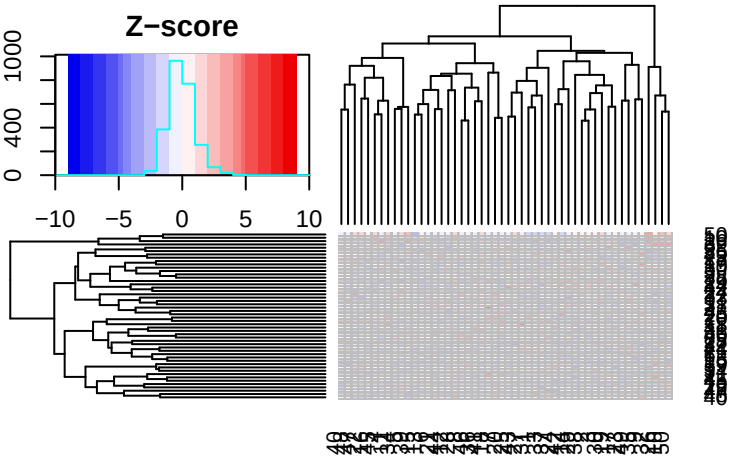
```
E(hhi.z.net)$color <- ifelse(E(hhi.z.net)$weight > 0, "red", "blue")
coloring <- E(hhi.z.net)$color
hhi.weight <- ifelse(abs(E(hhi.z.net)$weight) > 8, abs(E(hhi.z.net)$weight), -0.5)

#pdf("hhi.z.network.8plus.pdf", height=8, width=8, paper='special')
plot.igraph(hhi.z.net, vertex.label=c(1:50), layout=layout_in_circle,
            edge.color = coloring, edge.width=hhi.weight/4)
```



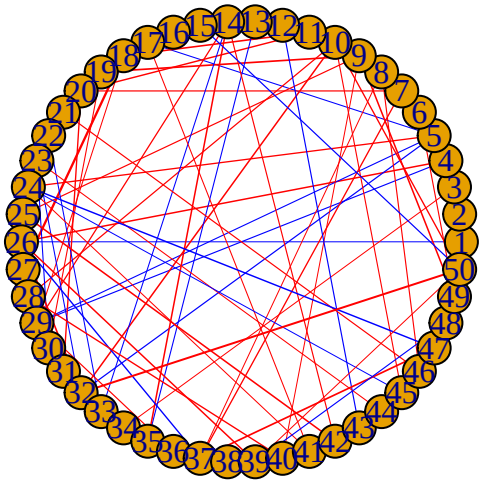
```
#dev.off()
```

```
# comparison to 50 random sets; z-scores are calculated using the same protocol
rdm.list <- c(1:50)
random.dim <- 50

rri.dat <- read.csv("../../random/rri.z.csv", header=F, stringsAsFactors = F)
rri.z <- matrix(unlist(rri.dat), nrow=random.dim, ncol=random.dim)

colnames(rri.z) <- rdm.list
rownames(rri.z) <- rdm.list

#my_palette <- colorRampPalette(c("blue2", "white", "red2"))(n = 20)
#colors = c(seq(-10,10,length=21))

heatmap.2(rri.z, col=my_palette, trace='none', breaks=colors,
          key.xlab=NA, key.title="Z-score", key.ylab=NA, key.xtickfun = NULL, key.ytickfun = NULL,
          #srtCol=45, adjCol=c(1,0), dendrogram = "both",
          margins=c(14.5,18), sepwidth=c(0.01,0.01), symbreaks = TRUE,
          sepcolor="grey", colsep=1:random.dim, rowsep=1:random.dim)
```

```
## Warning in image.default(z = matrix(z, ncol = 1), col = col, breaks =
## tmpbreaks, : unsorted 'breaks' will be sorted before use
```



```
rri.z.mat <- as.matrix(rri.z)
rri.z.net <- graph.adjacency(rri.z.mat, mode="undirected", weighted=T, diag=F)
#summary(rr.z.net)

E(rri.z.net)$color <- ifelse(E(rri.z.net)$weight > 0, "red", "blue")
coloring <- E(rri.z.net)$color
rri.weight <- ifelse(abs(E(rri.z.net)$weight) > 2, abs(E(rri.z.net)$weight), -0.5)

#pdf("rri.z.network.2plus.pdf", height=8, width=8, paper='special')
plot.igraph(rri.z.net, vertex.label=c(1:50), layout=layout_in_circle,
            edge.color = coloring, edge.width=rri.weight/4)
```




```
#dev.off()
```