NetPBS for hallmark sets

HBG

9/25/2019

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.

```
#set up required libraries
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)
```

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

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

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

```
list.file <- read.csv("data/hallmark.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"
```

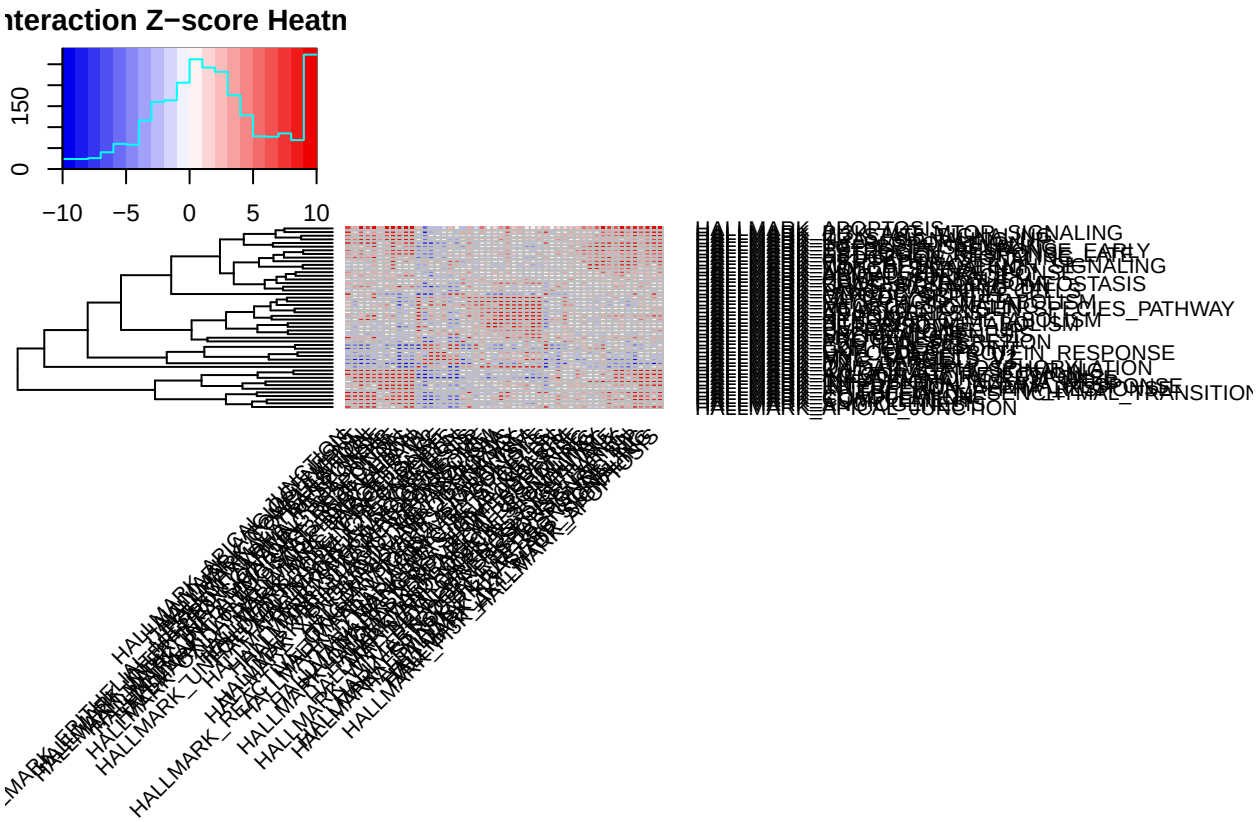
```
## [10] "HALLMARK_COAGULATION"
## [11] "HALLMARK_COMPLEMENT"
## [12] "HALLMARK_DNA_REPAIR"
## [13] "HALLMARK_E2F_TARGETS"
## [14] "HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION"
## [15] "HALLMARK_ESTROGEN_RESPONSE_EARLY"
## [16] "HALLMARK_ESTROGEN_RESPONSE_LATE"
## [17] "HALLMARK_FATTY_ACID_METABOLISM"
## [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("data/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

#calculate donw the interaction matrix

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("data/gene.lists/", hallmark.name[i], ".csv", sep="")
```

```
    fileA <- read.csv(hallmarkA, header=T, stringsAsFactors=F)
    hallmarkB <- paste("data/gene.lists/", 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)
```

Similar script will be applied to MS02star null models generated using "01.ms02star.R" Then the Z-score matrics can be calculated. Some eample output from the null models are saved in output/ms02.hhi/

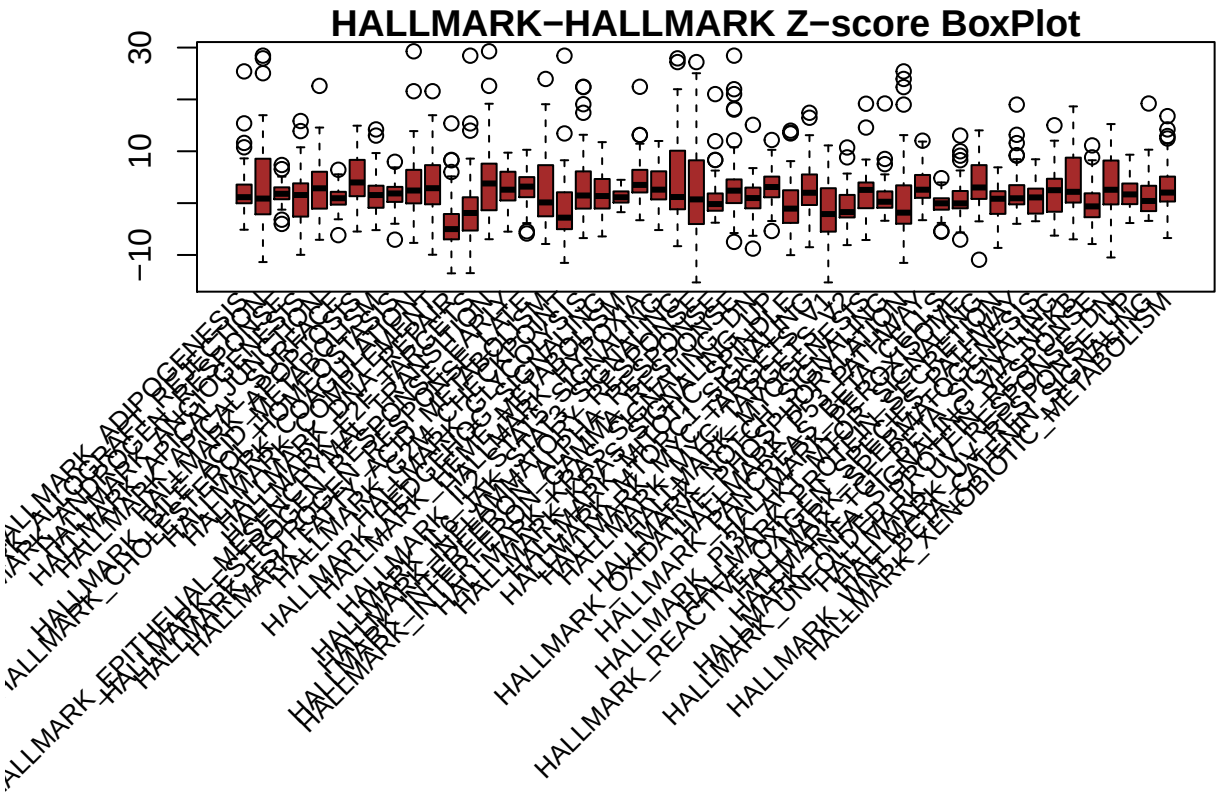
```
#Z-scores calculated from 10k null models is saved in "hhi.z.csv""

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(-10,10,length=21))

hallmark.matrix <- hhi.z
#We can set the nams of gene set as alphabetic serial numbers (from 1:50)
#colnames(hallmark.matrix) <- c(1:50)
#row.names(hallmark.matrix) <- c(1:50)
#png("hhi.z.heatmap.new2.png", width=12, height=4, res=600, units="in")
heatmap.2(hallmark.matrix, col=my_palette, trace='none', breaks=colors,
          key.xlab=NA, key.title="Interaction Z-score Heatmap",
          key.ylab=NA, key.xtickfun = NULL, key.ytickfun = NULL,
          srtCol=45, adjCol=c(1,0),
          dendrogram = "row",
          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()
```

We will creat a Z-score network for the hallmark sets here

```
#remove self-interactions
hhi.z.nodiag <- hhi.z
diag(hhi.z.nodiag) = NA

#generate a boxplot here
#pdf("hhi.z.boxplot.nodiag.pdf", height=6, width=15, paper='special')
par(mar=c(15,5,1,1))
boxplot(hhi.z.nodiag, main="HALLMARK-HALLMARK Z-score BoxPlot", xaxt="n", col='brown')
text(seq(1,50), par("usr")[3]-1, srt=45, adj=1, xpd=T,
     labels=paste(hallmark.name), cex=0.8)
```

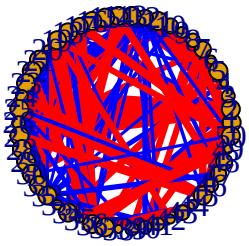



```
#dev.off()

#construct a network from the Z-score matrix (self-interactoins removed)
hhi.z.mat <- as.matrix(hhi.z.nodiag)
hhi.z.net <- graph.adjacency(hhi.z.mat, mode="undirected", weighted=T, diag=F)
summary(hhi.z.net)
```

```
## IGRAPH f62c79a UNW- 50 1225 --
## + attr: name (v/c), weight (e/n)
```

```
#we have counted the gene numbers of all hallmark sets and saved in "gene.count.csv"
#(it can be done in R, too)
#we will use number of genes to weight the size of the nodes
gene.count <- read.csv("gene.count.csv", header=T, stringsAsFactors = F)
v.size <- gene.count$gene.number

#set positive interactions in red and negative interactions in blue
E(hhi.z.net)$color <- ifelse(E(hhi.z.net)$weight > 0, "red", "blue")
coloring <- E(hhi.z.net)$color

#the top 5% enriched (hhi.top5) and suppressed (hhi.bot5) interactions
hhi.top5 <- quantile(hhi.z.mat[!is.na(hhi.z.mat)], probs=seq(0,1,1/20))[20]
hhi.bot5 <- quantile(hhi.z.mat[!is.na(hhi.z.mat)], probs=seq(0,1,1/20))[2]

#we only plot the top5 enriched/suppressed interactions
hhi.weight <- ifelse((E(hhi.z.net)$weight > hhi.top5 | E(hhi.z.net)$weight < hhi.bot5) , abs(E(hhi.z.ne

#pdf("hhi.z.network.top5.pdf", height=8, width=8, paper='special')
#note that the node sizes are weighted by the square root of the gene numbers
```

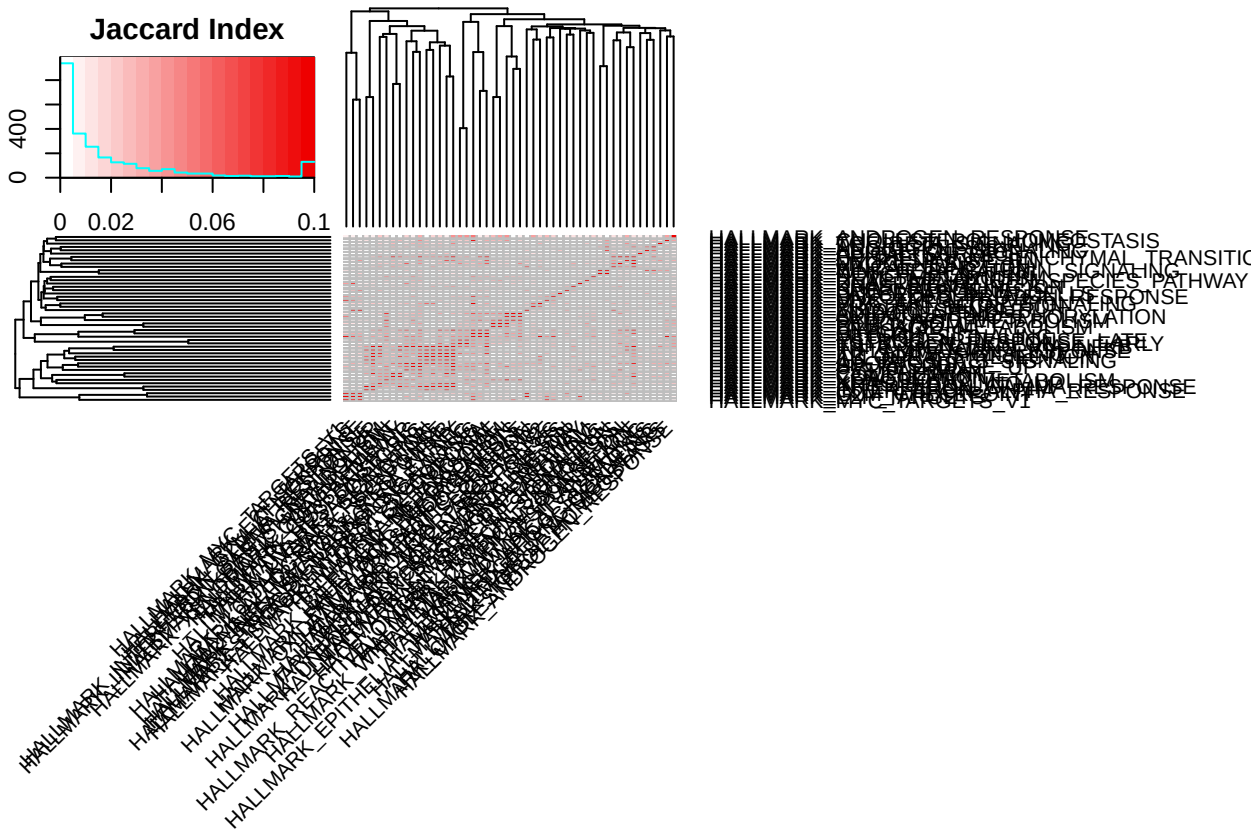
```
#such that the areas of the nodes reflect the size of the gene sets
#the edges are weighted by the Z-scores (divided by 4 for better visualizations)
plot.igraph(hhi.z.net, vertex.label=c(1:50), layout=layout_in_circle,
            edge.color = coloring, edge.width=hhi.weight/4, vertex.size=sqrt(v.size)*1.2)
```



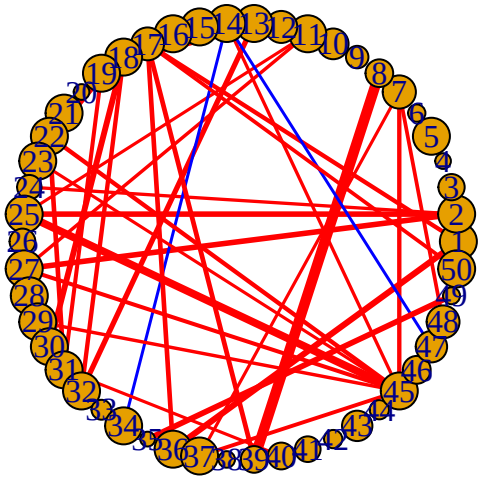
```
#dev.off()
```

Overlap matrix based on Jaccard-indices

```
#calculating the Jaccard matrix
overlap.matrix <- matrix(0, ncol=50, nrow=50)
for (i in 1:50) {
file.name <- paste("data/gene.lists/", hallmark.name[i], ".csv", sep="")
  file.a <- read.csv(file.name, header=T)
   list.a <- file.a$gene
   for (j in 1:50) {
     file.nameb <- paste("data/gene.lists/", hallmark.name[j], ".csv", sep="")
     file.b <- read.csv(file.nameb, header=T)
     list.b <- file.b$gene
     jac <- length(which(list.a %in% list.b))/length(unique(c(list.a, list.b)))
     overlap.matrix[i,j] = overlap.matrix[i,j] + jac
   }
}

#Then plot the Jaccard heatmap;
#note that there are no negative J values
colnames(overlap.matrix) <- hallmark.name
row.names(overlap.matrix) <- hallmark.name
my_palette <- colorRampPalette(c("white", "red2"))(n = 20)
colors = c(seq(0,0.1,length=21))
#png("hallmark.jaccard.png", width=12, height=11, res=600, units="in")
heatmap.2(overlap.matrix, col=my_palette, trace='none', breaks=colors,
          key.xlab=NA, key.title="Jaccard Index", 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=1:50, rowsep=1:50)
```




```
#dev.off()

#remove self-Jaccard indices (they will be 1)
#and construct a Jaccard-network of the gene sets using the top 5% J-indices

overlap.nodiag <- overlap.matrix
diag(overlap.nodiag) = NA
colnames(overlap.nodiag) <- c(1:50)
row.names(overlap.nodiag) <- c(1:50)
ol.mat <- as.matrix(overlap.nodiag)
ol.net <- graph.adjacency(ol.mat, mode="undirected", weighted=T, diag=F)
E(ol.net)$color <- ifelse(E(ol.net)$weight > 0.01, "red", "blue")
coloring <- E(ol.net)$color

#top 5% Jaccard value
jaccard.top5 <- quantile(overlap.matrix, probs=seq(0,1,1/20))[20]

E(ol.net)$color <- ifelse(E(ol.net)$weight > jaccard.top5, "red", "blue")
coloring <- E(ol.net)$color
ol.weight <- ifelse(E(ol.net)$weight < jaccard.top5, -0.05, E(ol.net)$weight)

coords <- layout_in_circle(ol.net)
#we need to enlarge the Jaccard indices for the edge width for a better visualization
#pdf("hallmark.jaccard.top5.pdf", height=8, width=8,paper='special')
plot.igraph(ol.net, cex=0.6, layout = coords, #vertex.label=c(1:50),
            edge.color = coloring, edge.width=ol.weight*15, vertex.size=sqrt(v.size)*1.2)
```

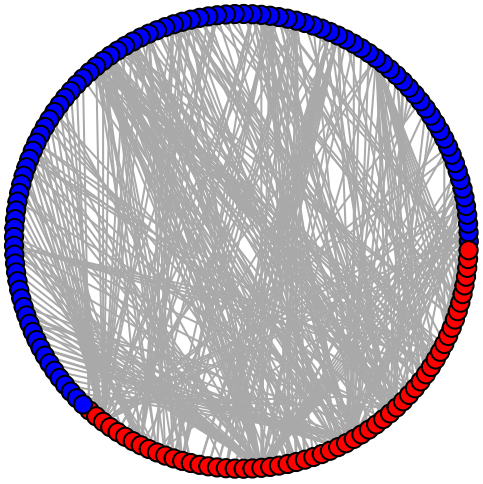


```
#dev.off()
```

Plot a subnetwork between set 5 and set 24

```
file.5 <- paste("data/gene.lists/", hallmark.name[5], ".csv", sep="")
file.24 <- paste("data/gene.lists/", hallmark.name[24], ".csv", sep="")
read.5 <- read.csv(file.5, header=T, stringsAsFactors = F)
read.24 <- read.csv(file.24, header=T, stringsAsFactors = F)
gene.5 <- read.5$gene
gene.24 <- read.24$gene

five24.int <- which(((geneA %in% gene.5) & (geneB %in% gene.24)) |
                      ((geneA %in% gene.24) & (geneB %in% gene.5)))
five24.A <- geneA[five24.int]
five24.B <- geneB[five24.int]

five24.subnet <- data.frame(cbind(five24.A, five24.B))
five24.sub.graph <- graph.data.frame(five24.subnet, directed=F)


'%ni%' <- Negate('%in%')

blue.id <- which((as_ids(V(five24.sub.graph)) %in% gene.5) &
                   (as_ids(V(five24.sub.graph)) %ni% gene.24))
red.id <-  which((as_ids(V(five24.sub.graph)) %in% gene.24) &
                   (as_ids(V(five24.sub.graph)) %ni% gene.5))
yellow.id <- which((as_ids(V(five24.sub.graph)) %in% gene.5) &
                     (as_ids(V(five24.sub.graph)) %in% gene.24))
five24.sub.order <- V(five24.sub.graph)[c(blue.id, yellow.id, red.id)]
five24.coords <- layout_in_circle(five24.sub.graph, order = five24.sub.order)

color <- rep("NA", times=length(V(five24.sub.graph)))
color[red.id] <- rep("red", times=length(red.id))
color[blue.id] <- rep("blue", times=length(blue.id))
color[yellow.id] <- rep("yellow", times=length(yellow.id))
V(five24.sub.graph)$color <- color

#pdf("five24.pdf")
plot.igraph(five24.sub.graph, vertex.color=V(five24.sub.graph)$color,
```

```
            vertex.size=8, edge.width=1,vertex.label=NA,
            order=five24.sub.order,layout=five24.coords)
```



```
#dev.off()
length(E(five24.sub.graph))  #=344
```

```
## [1] 344
```