

Hallmark Gene Sets Summary

HBG

12/7/2018

This script summarize the sub-network and enrichment results of 50 hallmark sets from MSigDB

```
rm(list=ls())
library(igraph)

##
## 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

library(pracma)

#list of 50 hallmark sets
list.file <- read.csv("list", header=F, stringsAsFactors=F)
hmk.set.list <- list.file$V1

#human PIN
human.pin <- read.csv("../human.pin.csv", header=T, stringsAsFactors = F)
geneA <- human.pin$geneA
geneB <- human.pin$geneB

#All genes appear in PIN
all.list <- unique(c(geneA, geneB))

hmk.summary <- c("Gene#", "Vertice#", "Edge#",
                "MaxD", "MaxD.gene", "MaxB", "MaxB.gene", "MaxC", "MaxC.gene",
                "PCC.DB", "PCC.DC", "PCC.BC",
                "Max.Cliq#", "Max.Cliq.D",
                "KEGG.match#", "KEGG.z.max", "KEGG.z.mean", "KEGG.z.sd",
                "BP.match#", "BP.z.max", "BP.z.mean", "BP.z.sd",
                "CC.match#", "CC.z.max", "CC.z.mean", "CC.z.sd",
                "MF.match#", "MF.z.max", "MF.z.mean", "MF.z.sd")
```

```

for (i in 1:length(hmk.set.list)) {
  ##the gene list and sub-network
  file.name <- paste(hmk.set.list[i], "/", hmk.set.list[i], ".csv", sep="")
  geneset <- read.csv(file.name, header=T, stringsAsFactors=F)
  gene.list <- geneset$gene
  gene.number <- length(gene.list) # this is the original number of genes the set
  #Constructing the sub network fromn the gene set
  subA <- geneA[which((geneA %in% gene.list) & (geneB %in% gene.list))]
  subB <- geneB[which((geneA %in% gene.list) & (geneB %in% gene.list))]
  genes.pin <- unique(c(subA, subB))
  gene.number.pin <- length(unique(genes.pin)) # this is the number of vertices of the sub-network
  sub.web <- data.frame(cbind(subA, subB))
  sub.edge.number <- length(sub.web[,1]) #this is the total edges of the sub-network

  # using igraph to build a graph of the sub-network
  sub.graph <- graph.data.frame(sub.web, directed = F)

  #normalized values for betweenness, degree of vertices
  sub.betweenness <- betweenness(sub.graph, v=V(sub.graph), directed=F, weights=NULL, normalized=T)
  sub.degree <- degree(sub.graph, v=V(sub.graph), normalized = T)
  sub.closeness <- closeness(sub.graph, v=V(sub.graph), weights=NULL, normalized=T)
  max.degree <- max(sub.degree)
  max.degree.gene <- genes.pin[which(sub.degree == max.degree)]
  mdg <- c()
  for (j in 1:length(max.degree.gene)) {
    mdg <- paste(mdg, max.degree.gene[j], sep="|")
  }
  mdg <- as.character(mdg)
  max.betweenness <- max(sub.betweenness)
  max.betweenness.gene <- genes.pin[which(sub.betweenness == max.betweenness)]
  mbg <- c()
  for (k in 1:length(max.betweenness.gene)) {
    mbg <- paste(mbg, max.betweenness.gene[k], sep="|")
  }
  max.closeness <- max(sub.closeness)
  max.closeness.gene <- genes.pin[which(sub.closeness == max.closeness)]
  mcg <- c()
  for (l in 1:length(max.closeness.gene)) {
    mcg <- paste(mcg, max.closeness.gene[l], sep="|")
  }
  #number.of.clique <- clique.number(sub.graph) #total number of cliques
  sub.largest.clique <- largest_cliques(sub.graph)
  number.of.largest.clique <- length(sub.largest.clique)
  degree.of.largest.clique <- length(sub.largest.clique[[1]])
  cor.deg.bet <- cor(sub.degree, sub.betweenness) #correlation of degree/betweenness
  cor.deg.clo <- cor(sub.degree, sub.closeness) #correlation of degree/closeness
  cor.bet.clo <- cor(sub.betweenness, sub.closeness) #correlation of betweenness/closeness

  ##the enrichments result from NetBAS and DAVID (top 10 from NetBAS)
  kegg.file <- paste(hmk.set.list[i], "/", "HMK.kegg.top10.txt", sep="")
  bp.file <- paste(hmk.set.list[i], "/", "HMK.bp.top10.txt", sep="")
  cc.file <- paste(hmk.set.list[i], "/", "HMK.cc.top10.txt", sep="")
  mf.file <- paste(hmk.set.list[i], "/", "HMK.mf.top10.txt", sep="")

```

```

kegg.data <- read.csv(kegg.file, header=T, sep="\t", stringsAsFactors=F)
bp.data <- read.csv(bp.file, header=T, sep="\t", stringsAsFactors=F)
cc.data <- read.csv(cc.file, header=T, sep="\t", stringsAsFactors=F)
mf.data <- read.csv(mf.file, header=T, sep="\t", stringsAsFactors=F)

#kegg
#kegg.pathway <- kegg.data$Pathway
kegg.netbas <- kegg.data$NetBAS
kegg.david <- kegg.data$DAVID
kegg.match <- length(which(!is.na(kegg.david))) #number of matched pathways in DAVID
kegg.z.max <- kegg.netbas[1]
kegg.z.mean <- mean(kegg.netbas)
kegg.z.sd <- sd(kegg.netbas)

#bp
bp.netbas <- bp.data$NetBAS
bp.david <- bp.data$DAVID
bp.match <- length(which(!is.na(bp.david)))
bp.z.max <- bp.netbas[1]
bp.z.mean <- mean(bp.netbas)
bp.z.sd <- sd(bp.netbas)

#cc
cc.netbas <- cc.data$NetBAS
cc.david <- cc.data$DAVID
cc.match <- length(which(!is.na(cc.david)))
cc.z.max <- cc.netbas[1]
cc.z.mean <- mean(cc.netbas)
cc.z.sd <- sd(cc.netbas)

#mf
mf.netbas <- mf.data$NetBAS
mf.david <- mf.data$DAVID
mf.match <- length(which(!is.na(mf.david)))
mf.z.max <- mf.netbas[1]
mf.z.mean <- mean(mf.netbas)
mf.z.sd <- sd(mf.netbas)

hmk.summary <- rbind(hmk.summary, c(gene.number, gene.number.pin, sub.edge.number,
max.degree, mdg,
max.betweenness, mbg,
max.closeness, mcg,
cor.deg.bet, cor.deg.clo, cor.bet.clo,
number.of.largest.clique, degree.of.largest.clique,
kegg.match, kegg.z.max, kegg.z.mean, kegg.z.sd,
bp.match, bp.z.max, bp.z.mean, bp.z.sd,
cc.match, cc.z.max, cc.z.mean, cc.z.sd,
mf.match, mf.z.max, mf.z.mean, mf.z.sd))
}

## Warning in closeness(sub.graph, v = V(sub.graph), weights = NULL,
## normalized = T): At centrality.c:2784 :closeness centrality is not well-
## defined for disconnected graphs

```


[illegible]

```
## Warning in closeness(sub.graph, v = V(sub.graph), weights = NULL,  
## normalized = T): At centrality.c:2784 :closeness centrality is not well-  
## defined for disconnected graphs
```

```
## Warning in closeness(sub.graph, v = V(sub.graph), weights = NULL,  
## normalized = T): At centrality.c:2784 :closeness centrality is not well-  
## defined for disconnected graphs
```

```
## Warning in closeness(sub.graph, v = V(sub.graph), weights = NULL,  
## normalized = T): At centrality.c:2784 :closeness centrality is not well-  
## defined for disconnected graphs
```

```
## Warning in closeness(sub.graph, v = V(sub.graph), weights = NULL,  
## normalized = T): At centrality.c:2784 :closeness centrality is not well-  
## defined for disconnected graphs
```

```
## Warning in closeness(sub.graph, v = V(sub.graph), weights = NULL,  
## normalized = T): At centrality.c:2784 :closeness centrality is not well-  
## defined for disconnected graphs
```

```
## Warning in closeness(sub.graph, v = V(sub.graph), weights = NULL,  
## normalized = T): At centrality.c:2784 :closeness centrality is not well-  
## defined for disconnected graphs
```

```
## Warning in closeness(sub.graph, v = V(sub.graph), weights = NULL,  
## normalized = T): At centrality.c:2784 :closeness centrality is not well-  
## defined for disconnected graphs
```

```
write.table(hmk.summary, file="Summary.Hallmark.Sets.csv", row.names = F, col.names = F, sep="," , quote
```