Visualize & cluster one-mode projection of a bi-partite graph

Genomic knowledge are often curated in the form of Genes. For example, a geneset of genes mapping to the chromosome locus chr8q24; a geneset of genes known to involve in DNA Repair Module, etc. Similarly, this data structure is also representative of patient-Genes data.

This can be thought of as a bipartite graph representing relation between individual Module to a gene. Often bioinformaticians are interested to visualize if there is any kind of relation between the Modules. This problem can be modeled as the conversion of two-mode network (bipartite graph) to one-mode network of Modules and visualize the resulting one-mode network. Here, I attempt to demonstrate how this can be achieved using R.

Load required libraries

```
library("stringr")
library("igraph")
library("RColorBrewer")
library("gplots")
library("cluster")
```

Load input data file

```
#file.dat <- url("https://dl.dropboxusercontent.com/u/30823824/dat_Genes.txt")
file.dat <- "data_table.tsv"
dat <- read.delim(file.dat, header=TRUE, stringsAsFactors=FALSE)
str(dat)
```

```
## 'data.frame':    37 obs. of  2 variables:
##  $ SampleID: chr  "T-01" "T-02" "T-03" "T-04" ...
##  $ Genes   : chr  "FOXA1:GATA3:XBP1:COX17:KATNAL1:SCO2:STAT6:TFF3:THY1" "AR:NCOA3:SPDEF:TFF1:XBP1" "
```

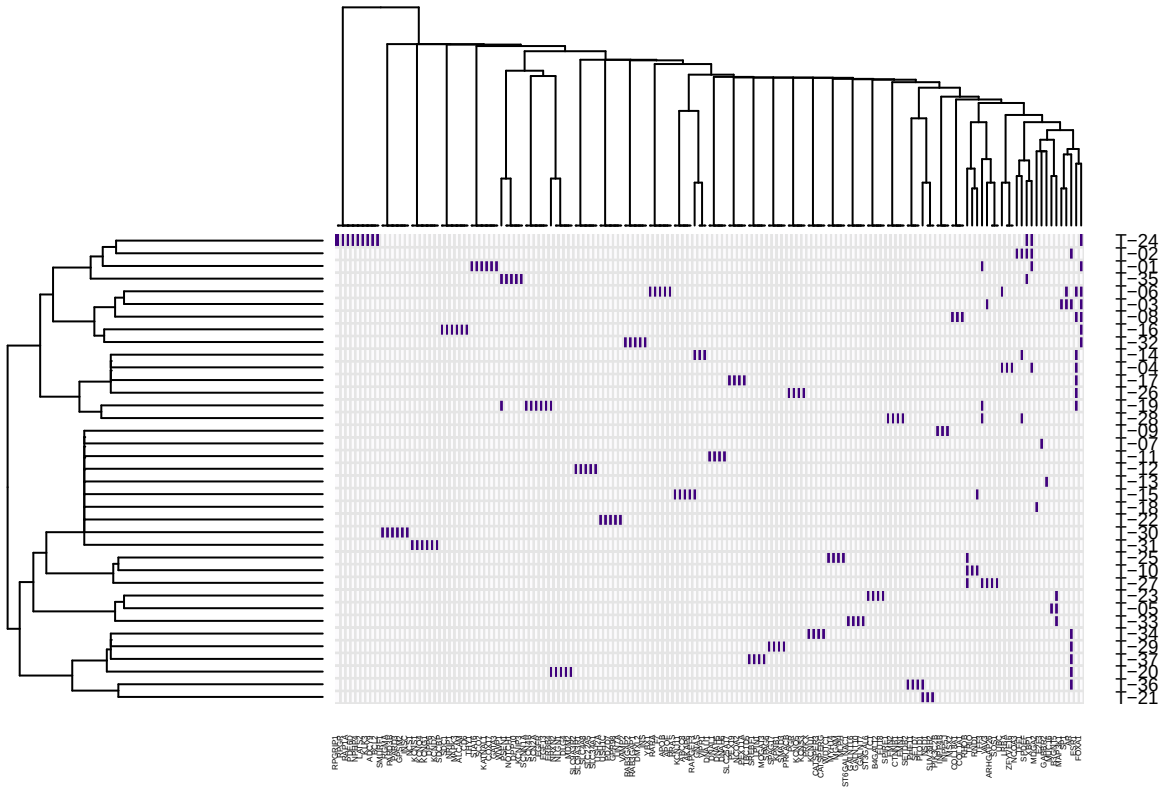
Compute bipartite graph

```
get.bip <- function(df){
  ids <- df$SampleID
  genes <- sort(unique(unlist(str_split(df$Genes, ":"))),decreasing=F)

  bip <- matrix(0, nrow=length(ids), ncol=length(genes), dimnames=list(ids, genes))

  list.genes <- str_split(df$Genes, ":")
  list.gene.index <- lapply(list.genes, function(x) which(colnames(bip) %in% x))

  for(i in 1:length(list.gene.index)){
    bip[i,list.gene.index[[i]]] <- 1
  }
  return(bip)
}

bip <- get.bip(dat)
```

```
bip[1:5,1:5]
```

```
##       ABCC8 ADCY9 AKAP6 ALCAM APOB
## T-01      0     0     0     0    0
## T-02      0     0     0     0    0
## T-03      0     0     0     0    0
## T-04      0     0     0     0    0
## T-05      0     0     0     0    0
```

```
#dat$TotalGenes <- unlist(lapply(list.gene.index, length))
#dat$Color <- rev(jColFun(nrow(dat.sub)))

jColFun <- colorRampPalette(brewer.pal(n = 9, "Purples"))

#Plot heatmap of the bi-partite graph
heatmap.2(bip, col = jColFun(256),
          Colv=TRUE, Rowv = TRUE,
          dendrogram ="both", trace="none",  key="FALSE",
          hclustfun = function(x) hclust(x, method = "ward.D2"),
          distfun = function(x) dist(x, method = "binary"),
          colsep=c(1:500), rowsep=c(1:500),
          sepcolor="grey90", sepwidth=c(0.05,0.05),
          cexRow=0.7, cexCol=0.3)
```



Compute CSI: Connection Specific Index

```
# First compute pearson correlation
dat.pcc <- cor(t(bip), method="pearson")
dat.pcc[1:5,1:5]
```

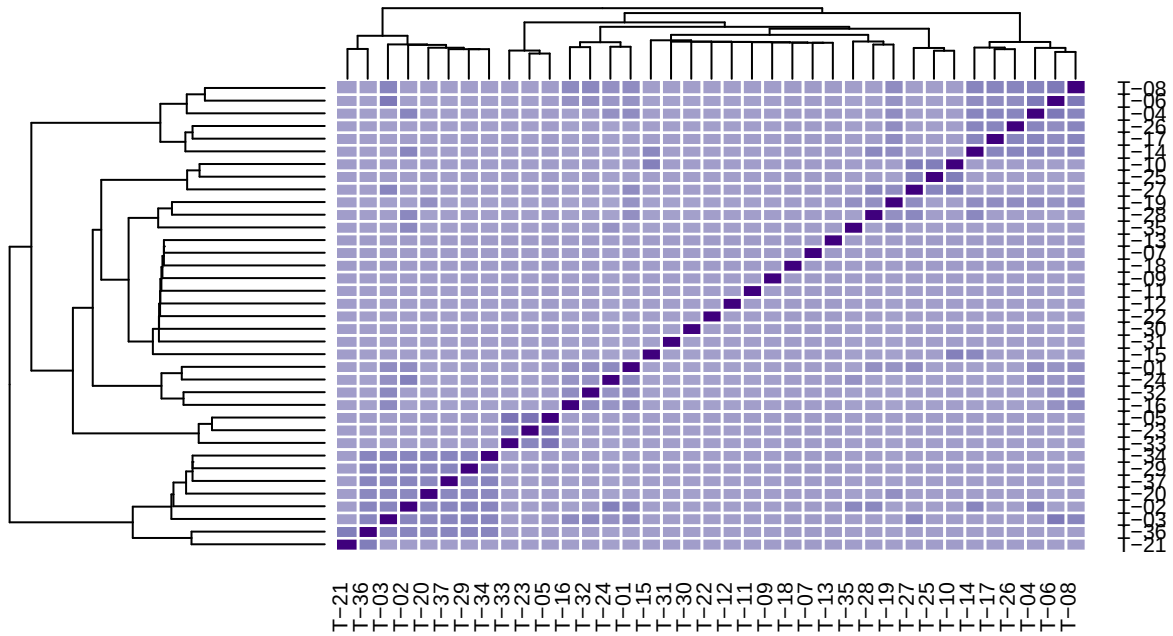
```
##             T-01        T-02        T-03        T-04        T-05
## T-01  1.00000000  0.10974355  0.10974355  0.10974355 -0.02916748
## T-02  0.10974355  1.00000000  0.17260274  0.17260274 -0.02144028
## T-03  0.10974355  0.17260274  1.00000000 -0.03424658 -0.02144028
## T-04  0.10974355  0.17260274 -0.03424658  1.00000000 -0.02144028
## T-05 -0.02916748 -0.02144028 -0.02144028 -0.02144028  1.00000000
```

```
# Function to compute CSI: Connection Specific Index ###
get.csi <- function(dat){
  mat <- matrix(0, nrow=nrow(dat), ncol=ncol(dat), dimnames=list(rownames(dat),colnames(dat)))

    for(i in 1:nrow(dat)){
        a <- rownames(dat)[i]

        for(j in 1:ncol(dat)){
            b <- colnames(dat)[j]
            pcc.ab <- dat[a,b] - 0.05

            conn.pairs.a <- colnames(dat)[which(dat[a,] >= pcc.ab)]
            conn.pairs.b <- rownames(dat)[which(dat[,b] >= pcc.ab)]

            conn.pairs.ab <- length(union(conn.pairs.a,conn.pairs.b))
            n <- nrow(dat)

            csi <- 1 - (conn.pairs.ab/n)
            mat[i,j] <- csi
        }
    }
    return(mat)
}

dat.csi <- get.csi(dat.pcc)
dat.csi[1:5,1:5]
```

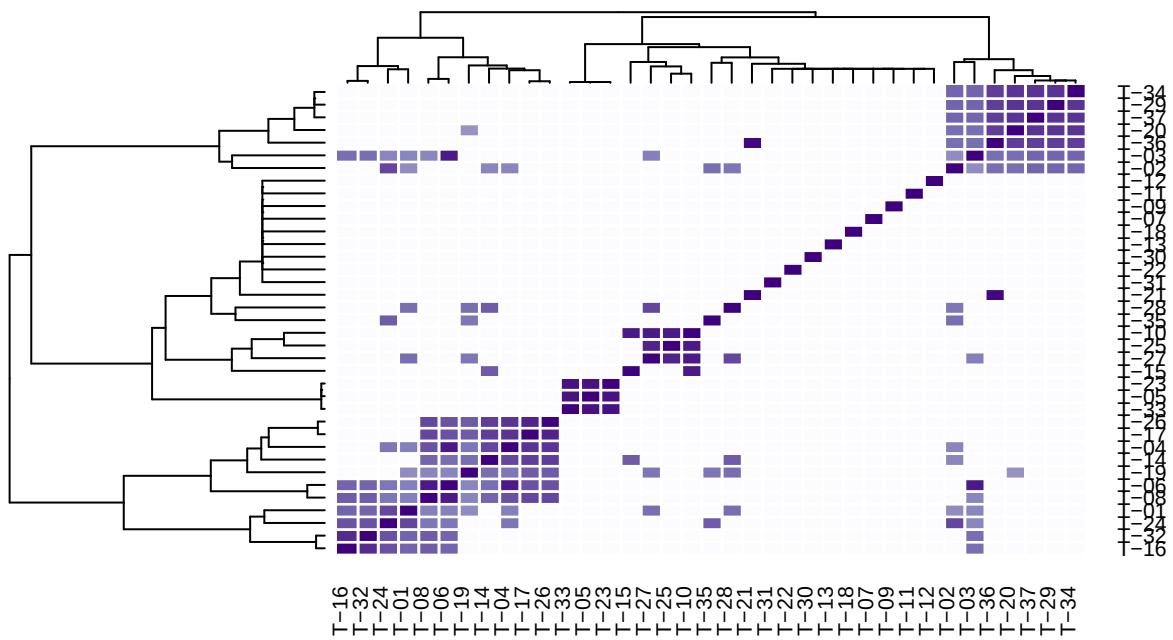
```
##           T-01      T-02      T-03      T-04     T-05
## T-01 0.9729730 0.5405405 0.5675676 0.5945946 0.000000
## T-02 0.5405405 0.9729730 0.5405405 0.5675676 0.000000
## T-03 0.5675676 0.5405405 0.9729730 0.0000000 0.000000
## T-04 0.5945946 0.5675676 0.0000000 0.9729730 0.000000
## T-05 0.0000000 0.0000000 0.0000000 0.0000000 0.972973
```

```
heatmap.2(as.matrix(dat.pcc), col = jColFun(256),
          Colv=TRUE, Rowv = TRUE,
          dendrogram ="both", trace="none",
          hclustfun = function(x) hclust(x, method = "ward.D2"),
          distfun = function(x) dist(x, method = "minkowski", p=2),
          colsep=c(1:500), rowsep=c(1:500),
          sepcolor="white", sepwidth=c(0.05,0.05),
          key="FALSE",cexRow=0.8, cexCol=0.8, main="PCC")
```

PCC




```
heatmap.2(as.matrix(dat.csi), col = jColFun(256),
          Colv=TRUE, Rowv = TRUE,
          dendrogram ="both", trace="none",
          hclustfun = function(x) hclust(x, method = "ward.D2"),
          distfun = function(x) dist(x, method = "minkowski", p=2),
          colsep=c(1:500), rowsep=c(1:500),
          sepcolor="white", sepwidth=c(0.05,0.05),
          key="FALSE",cexRow=0.8, cexCol=0.8, main="CSI")
```



Visualizing the network

```
#Convert the adjacency (corellation) matrix to an igraph object
g <- graph.adjacency(dat.csi, mode = "undirected", weighted=TRUE, diag=FALSE)
g <- simplify(g, edge.attr.comb=list(weight="sum"))
```

```
#get the node size ratio
#for(ctr in 1:length(V(g)$name)){
#  index <- which(dat$Module == V(g)$name[ctr])
#  V(g)$size[ctr] <- dat$TotalGenes[index]
#   V(g)$color[ctr] <- dat$Color[index]
#}
V(g)$size <- 5

# Set vertex attributes
V(g)$label <- V(g)$name
V(g)$label.color <- "black"
V(g)$label.cex <- .8
V(g)$label.dist <- 0.3
V(g)$label.family <- "Helvetica"
V(g)$frame.color <- "grey"

E(g)$color <- rgb(.6,.6,0,E(g)$weight)
E(g)$width <- E(g)$weight * 5
```

```
### Generate Graph Output File
file.graph.output <- "graph_projection.gml"
write_graph(graph=g, file=file.graph.output, format="gml")
```

```
plot(g, layout=layout.kamada.kawai)
```