0. Installing, loading library and inputs

Before installing MEGENA package, it requires the following dependencies:

Imports: Rcpp (>= 0.11.1)
Depends: R (>= 3.0.1),igraph (>= 0.7.0),Matrix (>= 1.1-3),cluster (>= 1.15.3),foreach (>= 1.4.1),doSNOW (>= 1.0.12),ggplot2 (>= 0.9.3.1),snow (>= 0.3-13),survival (>= 2.37.7),reshape (>= 0.8.4),fpc (>= 2.1-7)
LinkingTo: Rcpp,BH

Once these are properly installed, then install MEGENA.
MEGENA only requires gene expression matrix as an input data. The following code chuck loads MEGENA package, set input gene expression data matrix "datExpr":

```
#####################################################################################################################
rm(list = ls())

library(MEGENA)
wkdir <- NULL  # This needs to be set up to desired folder. All outputs will be written in "wkdir" in the following code chunks.

# input expression data: load datExpr into R working space.
dat.file <- NULL # This can be a text file containing gene expression matrix. If !is.null(datExpr), this will not be used to load data.
data(Sample_Expression) # load toy example data "datExpr", that comes with the package.

# annotation attributes: If genes need to be converted different ids, annot.file is a table in text file (.txt) providing necessary annotation
annot.file <- NULL # if NULL, ids won't be convereted
symbol.col <- 3 # If annot.file provided, symbol.col = column index for gene symbols
id.col <- 2 # if annot.file provided, id.col = column index for gene ids to be converted.
```

1. Set up computation parameters:

The following code chuck specifies necessary parameters required for MEGENA,

```
################ computation parameters
n.cores <- 8; # number of cores to be used for parallelization.
doPar <- TRUE; # TRUE = perform parallel PFN -> MCA. FALSE = no parallelization
method = "pearson" # Currently "pearson" (Pearson's correlation) and "spearman" (Spearman's correlation available) for correlation.
FDR <- 0.05; # FDR threshold to identify significant interactions.
n.cor.perm = 100; # Number of permutations in correlation screening.
n.hub.perm = 100; # Number of permutations to calculate hub significance in MHA

mod.pval = 0.05; # module p-value threshold in MCA
hub.pval = 0.05; # hub p-value threshold in MHA

min.size = 10 # minimum module size
max.size = NULL # maximum module size

##############################################################################################################
```

2. PFN computation

The following code chuck sets working environment by specifying working directory, loading gene expression data, and computes PFN with correlation screening with provided FDR threshold by "FDR". The final network is written into file "MEGENA_Network.txt" into a 3-column edgelist where the third column specifies edge weights:

```
##################################################################################################################
# set working directory
if (is.null(wkdir))
{
 wkdir <- "MEGENA_toy_example"
```

```
 dir.create(wkdir)
}

setwd(wkdir)

################
# load data
if (!any(ls() == "datExpr"))
{
 if (!is.null(dat.file)) datExpr <- load.data(dat.file,gsub.from = NULL)
}

# calculate correlation
ijw <- calculate.correlation(datExpr,
doPerm = n.cor.perm,doPar = FALSE,num.cores = n.cores,method = method,
FDR.cutoff = FDR,n.increment = 100,is.signed = FALSE,
output.permFDR = TRUE,output.corTable = TRUE,saveto = NULL)

# register multiple cores for parallelization
if (doPar) {
	set.parallel.backend(n.cores)
}

rm(datExpr)

# compute PFN
el <- calculate.PFN(ijw[,1:3],doPar = doPar,num.cores = n.cores)
write.table(el,file = "MEGENA_Network.txt",sep = "\t",row.names = F,col.names = T,quote = F)
rm(ijw)
```

3. *Perform Multiscale Clustering Analysis (MCA) -> Multiscale Hub Analysis (MHA)*

- Summary files

The following code chuck performs MCA and MHA, where these outputs are stored in a single R object called “MEGENA.output”. The full information of “MEGENA.output” is summarized in module level by function “MEGENA.ModuleSummary()”, in hub level by “get.hub.summary()”. The respective outputs are written into files: “module_summary.txt”,” hub_summary.txt”.

- Module file

The final set of significant modules are written in “multiscale_significant.modules.txt”. If annotation is provided, the gene symbol mapped file is “multiscale_significant.modules.mapped.txt”. The modules in these files are written in .gmt format, similar to MSigDB signatures, and can be read into R space by function “read.geneSet(filename)”.

- Modulewise subnetwork visualizations

These modules and hubs are also used to identify respective subnetworks in PFN, and plotted under the subfolder “subnetworks”.

```
##################################################################################################################
################ do clustering
# make igraph objects
g <- graph.data.frame(el,directed = F)

# check minimum and maximum module sizes
if (is.null(max.size)) max.size = vcount(g)/2
if (is.null(min.size)) min.size = 10

# perform MCA -> MHA
MEGENA.output <- do.MEGENA(g,
mod.pval = mod.pval,hub.pval = hub.pval,remove.unsig = TRUE,
```

```
min.size = min.size,max.size = max.size,
doPar = TRUE,num.cores = n.cores,n.perm = n.hub.perm,save.output = TRUE)

save(MEGENA.output,file = "MEGENA.output.RData")

############### summarize module level results and get significant output
# load annotation for probes
annot.table <- NULL;
if (!is.null(annot.file)) annot.table <- read.delim(file = annot.file,sep = "\t",header = T)

cat("- Summarize modules...\n")
output <- MEGENA.ModuleSummary(MEGENA.output,
mod.pvalue = mod.pval,hub.pvalue = hub.pval,
min.size = min.size,max.size = max.size,
annot.table = annot.table,id.col = id.col,symbol.col = symbol.col,
output.sig = TRUE)

summary.table <- output$module.table

write.table(summary.table,file = "module_summary.txt",sep = "\t",row.names = F,col.names = T,quote = F)

################ output final modules
output.geneSet.file(output$modules,"multiscale_significant.modules.txt")
if (!is.null(annot.table)) output.geneSet.file(output$mapped.modules,"multiscale_significant.modules.mapped.txt")

################ make modulewise subnetwork plots
cat("- Plotting subnetworks...\n")
hubs <- lapply(MEGENA.output$hub.output$module.degreeStat[names(output$modules)],function(x) as.character(x[[1]][which(x[[3]] < hub.pval)]))

label.genes <- NULL
if (!is.null(annot.table))
{
 label.genes <- as.character(annot.table[[symbol.col]]);
 names(label.genes) <- as.character(annot.table[[id.col]])
}

dir.create("subnetworks");
for (i in 1:length(output$modules))
{
 outputfname <- paste("subnetworks/",names(output$modules)[i],".png",sep = "")
 output.net <- plot.subnetwork(genes = output$modules[[i]],network = g,keydrivers = hubs[[i]],
 label.genes = label.genes,use.label = !is.null(label.genes),outputfname = outputfname,
		vertex.label.color = "red",vertex.frame.color = "black",vertex.color = "black",vertex.label.dist=0.25,vertex.label.degree = -pi/2,edge.color = "grey")
}


############### summarize hubs
cat("- Summarizing hubs...\n")
hub.Df <- get.hub.summary(MEGENA.output)
write.table(hub.Df,file = "hub_summary.txt",sep = "\t",row.names = F,col.names = T,quote = F)

quit(save = "no",status = 0)
```