Protocol for Biological Networks creation

Anaïs Baudot
November 2014
CNRS I2M Marseille

Protocol to generate the 7 Biological Networks containing interactions between human proteins from different interaction sources. The networks used in the article "Identifying Communities from Multiplex Biological Networks" were created as of December, 2014. The pdf file was created as of April, 2015. The network sizes hence may differ slightly.

Libraries

```
library("reshape")
library("biomaRt")
library("igraph")
library("PSICQUIC")
library("AnnotationDbi")
library("graph")
library("graphite")
```

Set working directory

Set your working directory

Functions

1) Function to check and clean networks
   For each raw network created, the function will remove auto-interactions and duplicates, and extract the largest connected component. Based on the igraph package.
   It takes as an input a matrix with 2 columns, format "ncol" (one line per interaction, the 2 interactors are separated by a tab), and returns a network as an igraph object.

```
# parameter = RawNetwork as matrix

CheckNetwork <- function (RawNetwork) {

  Net<-graph.edgelist(RawNetwork, directed=FALSE)

  Net2 <- igraph::simplify(Net) # simplify

  # Select largest connected component (the first one)
  select_nodes <- which(clusters(Net2)$membership == 1)

  # Extract network corresponding to the largest connected component
  Net_CC <- induced.subgraph(Net2, select_nodes)

  return(Net_CC)
}
```

2) Function to fetch pathway data from pathway databases
   Uses the graphite package (Sales G, Calura E and Romualdi C (2014). graphite: GRAPH Interaction from pathway Topological Environment. R package version 1.12.0.). This function was initially developped by Marie-Claire Kasai, Master Student in Bioinformatics, 2014.

```r
# parameter = name of the pathway database, such as "kegg" or "reactome"
GetPathwayData <- function (db.name){
  db <- get(db.name)
  OneDb.HGNC <- data.frame()

  for (i in 1:length(db)) {
      current.pathway <- (convertIdentifiers(db[[i]],"symbol"))
      current.pathway.info <- try(data.frame(src = current.pathway@edges$src,
                                             dest = current.pathway@edges$dest,
                                             direction = current.pathway@edges$direction,
                                             type = current.pathway@edges$type,
                                             name = current.pathway@title),silent=TRUE)
      OneDb.HGNC <- rbind(OneDb.HGNC,current.pathway.info)
    }

  OneDb.HGNC<-OneDb.HGNC[complete.cases(OneDb.HGNC),] #remove NA rows

  #transform the directed edge into undirected edge
  OneDb.HGNC[OneDb.HGNC == "directed"] <- "undirected"

  #create a sorted labelled per edge
  OneDb.HGNC$name1 <- pmin(as.vector(OneDb.HGNC$src), as.vector(OneDb.HGNC$dest))
  OneDb.HGNC$name2 <- pmax(as.vector(OneDb.HGNC$src), as.vector(OneDb.HGNC$dest))
  OneDb.HGNC$inter.label <- paste(OneDb.HGNC$name1,OneDb.HGNC$name2, sep="_")

  #select the unique inter.label = unique edge
  OneDb.HGNC <- subset(OneDb.HGNC,!(duplicated(OneDb.HGNC$inter.label)))

  # select the 2 columns of interest
  NetworkRaw<-as.matrix(OneDb.HGNC[,c(1,2)])
  colnames(NetworkRaw) <- c("SymbolA", "SymbolB")

  return(NetworkRaw)

}
```

Co-Expression Network from RNA-seq expression data

Load data file

- Download and unzip expression data file from Human Protein Atlas in the working directory
  http://www.proteinatlas.org/about/download
  RNA data (rna.csv.zip)

- These data correspond to RNA-seq expression levels in 44 cell lines and 32 tissues. It contains Ensembl genes and corresponding FPKM values.

```r
data<-read.csv("rna.csv", sep=",", header=TRUE)
head(data, n=3L)
```

```
##              Gene Sample Value Unit Abundance
## 1 ENSG00000000003  A-431  21.3 FPKM    Medium
## 2 ENSG00000000003   A549  32.5 FPKM    Medium
## 3 ENSG00000000003 AN3-CA  38.2 FPKM    Medium
```

Select column of interest (Gene, Sample, Value), transform to matrix and transform gene column in row names

```r
data2<-data[,c(1,2,3)]

matrix<-cast(data2, Gene ~ Sample) # matrix

rownames(matrix)<-matrix[,1]
matrix[,1]<-NULL
matrix<-as.matrix(matrix)
```

Compute correlations between gene expression

The spearman correlations between all pairs of genes are computed according to their expression in all the samples.
Correlations need to be computed between Genes = rows, hence we need to transpose the matrix first.

```r
CR<-cor(t(matrix), method="spearman")

#Round to second decimal
CR2<-round(CR,2)
```

Select correlation threshold and build a network

The correlation threshold is set to 0.7. All pairs of genes that have a correlation above this threshold (positive or negative) are considered as linked in the Co-Expression network.

```r
# select indexes over the threshold
inds=which(abs(CR2[1:nrow(CR2),]) >= 0.7, arr.ind=TRUE)

#retrieve names matching the index
rnames=rownames(CR2)[inds[,1]]
cnames=colnames(CR2)[inds[,2]]

#construct a network from indexes
net<-apply(cbind(rnames, cnames),1, function(x) unname(unlist(x)))
result<-t(net)

# transform matrix into data.frame
Network_EnsemblID<-as.data.frame(result)

# change column names
colnames(Network_EnsemblID) <- c("ensg1", "ensg2")
```

Convert Ensembl Ids to HGNC gene names

The network constructed in the previous step contains genes names in Ensembl Id format. For homogeneity with the other networks, these names are here converted to HGNC gene symbols, thanks to the biomaRt package.

```
# BiomaRt
mart <- useMart(biomart="ensembl", dataset="hsapiens_gene_ensembl")

# find synonyms
syn<-getBM(attributes=c("ensembl_gene_id", "hgnc_symbol"), filters="ensembl_gene_id", values=result[,1]

#merging tables
#first column
m<- merge(Network_EnsemblID, syn, by.x="ensg1", by.y="ensembl_gene_id")
#second column
m2<-merge(m, syn, by.x="ensg2", by.y="ensembl_gene_id")

# select the column of interest with the new names
m3<-m2[,c(3,4)]

#remove the lines that have "NA" or blank
m3[m3==""]<-NA
m4<-na.omit(m3)

#transform into matrix
ExprNet<-as.matrix(m4)
```

Check network

Apply the function CheckNetwork.

```
Expr <- CheckNetwork(ExprNet)
```

Print out the network

The expression Network contains:

```
ecount(Expr) # interactions
```

```
## [1] 1108728
```

```
vcount(Expr) # nodes
```

```
## [1] 9212
```

```
write.graph(Expr, "Expressionnet_CC_HGNC.gr", format="ncol", weights=NULL)
```

Protein-protein interaction Network from PSICQUIC + CCSB Interactome Database

The goal is to construct a protein-protein interaction network containing binary direct physical protein interactions. Two sources are used :
The PSICQUIC portal, which allow to retrieve interaction data from different databases participating to the IMEX consortium. These databases describe the interactions in the PSI-MI format, allowing to select direct interactions between human proteins.
Reference: del-Toro, N., Dumousseau, M., Orchard, S., Jimenez, R. C., Galeota, E., Launay, G., … Hermjakob, H. (2013). A new reference implementation of the PSICQUIC web service. Nucleic Acids Research, 41(Web Server issue), W601–6.
The Center for Cancer Systems Biology (CCSB) generates human interactome data. They have released in particular the dataset HI-II-2014 containing validated yeast 2-hybrid interactions between human proteins.
Reference: Rolland, T., Taşan, M., Charloteaux, B., Pevzner, S. J., Zhong, Q., Sahni, N., … Vidal, M. (2014). A Proteome-Scale Map of the Human Interactome Network. Cell, 159(5), 1212–1226. doi: 10.1016/j.cell.2014.10.050
The code below creates networks from these 2 sources and merge them together.

Human PPI from PSICQUIC portal

Download data from PSICQUIC

```
psicquic <- PSICQUIC()

# Select databases
DB<-c("BioGrid","BIND","DIP","InnateDB-IMEx","IntAct","MatrixDB","MBInfo",
      "MINT","Reactome","Reactome-FIs","UniProt")

# interactions human, from list of DB, type direct
tbl.big <- PSICQUIC::interactions(psicquic, species="9606",provider=DB, type="direct interaction")
```

Mapping gene names to HGNC
Different gene name formats are used in the different databases. Everything is converted to HGNC gene symbol.

```
idMapper <- IDMapper("9606")

tbl.big.2 <- addGeneInfo(idMapper,tbl.big) # mapping
```

Extract columns of interest (protein A / protein B) and construct binary network

```
NetHGNC<-tbl.big.2[, c("A.name", "B.name")]

# replace empty cells by NA
NetHGNC[NetHGNC == ""] <- NA
NetHGNC[NetHGNC == "-"] <- NA

#remove incomplete lines
NetHGNC2<-NetHGNC[complete.cases(NetHGNC),]
colnames(NetHGNC2) <- c("SymbolA", "SymbolB")
```

Human PPI from CCSB interactome database

Download the data file onto your working directory from
http://interactome.dfci.harvard.edu/H_sapiens/index.php
Caution : column headers contain many spaces that need to be removed manually.

```
# Read file
file_CCSB<-read.table("HI-II-14.tsv", header=TRUE)

#select column of interest
CCSB2<-file_CCSB[, c("SymbolA", "SymbolB")]

# check NA or empty cells
CCSB2[CCSB2 == ""] <- NA
CCSB2[CCSB2 == "-"] <- NA
CCSB3<-CCSB2[complete.cases(CCSB2),]
colnames(CCSB3) <- c("SymbolA", "SymbolB")
```

Merge the 2 PPI Networks

```
TotalPPI <-rbind (CCSB3, NetHGNC2)
TotalPPI<-as.matrix(TotalPPI)
```

Check network

Goal is to select the largest connected component, to remove auto-interactions and duplicates

```
PPI <- CheckNetwork(TotalPPI)
```

Print out network

The PPI Network contains:

```
ecount(PPI) # interactions
```

```
## [1] 60920
```

```
vcount(PPI) # nodes
```

```
## [1] 12060
```

```
write.graph(PPI, "PPI_CC_HGNC.gr", format="ncol", weights=NULL)
```

Interaction Networks from Signaling Pathway databases

The fetching of pathways, and their conversions to binary interaction maps are done thanks to the R package graphite. http://www.bioconductor.org/packages/release/bioc/html/graphite.html Please check the vignette for definitions and protocols. http://www.bioconductor.org/packages/release/bioc/vignettes/graphite/inst/doc/graphite.pdf Sales G, Calura E and Romualdi C (2014). graphite: GRAPH Interaction from pathway Topological Environment. R package version 1.12.0.

Five pathway databases are used to create 5 different networks.These databases are Kegg, Spike, Biocarta, PID and Reactome.

KEGG

Create the network using the function GetPathwayData defined at the beginning of this document.

```
KeggRaw <- GetPathwayData ("kegg")
```

Check network
Goal is to select the largest connected component, to remove auto-interactions and duplicates

```
Kegg <- CheckNetwork(KeggRaw)
```

Print out network
The Kegg Network contains:

```
ecount(Kegg) # interactions
```

```
## [1] 50066
```

```
vcount(Kegg) # nodes
```

```
## [1] 4582
```

```
write.graph(Kegg, "Kegg_CC_HGNC.gr", format="ncol", weights=NULL)
```

Biocarta

Create the network using the function GetPathwayData defined at the beginning of this document.

```
BiocartaRaw <- GetPathwayData ("biocarta")
```

Check network
Goal is to select the largest connected component, to remove auto-interactions and duplicates

```
Biocarta <- CheckNetwork(BiocartaRaw)
```

Print out network
The Biocarta Network contains:

```r
ecount(Biocarta) # interactions
```

```
## [1] 5281
```

```r
vcount(Biocarta) # nodes
```

```
## [1] 934
```

```r
write.graph(Biocarta, "Biocarta_CC_HGNC.gr", format="ncol", weights=NULL)
```

Spike

Create the network using the function GetPathwayData defined at the beginning of this document.

```r
spikeRaw <- GetPathwayData ("spike")
```

Check network
Goal is to select the largest connected component, to remove auto-interactions and duplicates

```r
Spike <- CheckNetwork(spikeRaw)
```

Print out network
The Spike Network contains:

```r
ecount(Spike) # interactions
```

```
## [1] 1708
```

```r
vcount(Spike) # nodes
```

```
## [1] 829
```

```r
write.graph(Spike, "Spike_CC_HGNC.gr", format="ncol", weights=NULL)
```

PID Pathway Interaction Database

Create the network using the function GetPathwayData defined at the beginning of this document.

```r
nciRaw <- GetPathwayData ("nci")
```

Check network
Goal is to select the largest connected component, to remove auto-interactions and duplicates

```r
nci <- CheckNetwork(nciRaw)
```

Print out network
The nci Network contains:

```r
ecount(nci) # interactions
```

```
## [1] 14253
```

```r
vcount(nci) # nodes
```

```
## [1] 2387
```

```r
write.graph(nci, "PID_CC_HGNC.gr", format="ncol", weights=NULL)
```

Reactome

Create the network using the function GetPathwayData defined at the beginning of this document.

```r
reactomeRaw <- GetPathwayData ("reactome")
```

Check network
Goal is to select the largest connected component, to remove auto-interactions and duplicates

```r
reactome <- CheckNetwork(reactomeRaw)
```

Print out network
The reactome Network contains:

```r
ecount(reactome) # interactions
```

```
## [1] 112049
```

```r
vcount(reactome) # nodes
```

```
## [1] 6744
```

```r
write.graph(reactome, "reactome_CC_HGNC.gr", format="ncol", weights=NULL)
```