ProNet Tutorial

Xiang-Yun Wu and Xia-Yu Xia
xiaxiayu.thu@hotmail.com

October 21, 2015

Abstract

Increasing quantity and quality of omics data creates the strong demand on integrative approaches to analyze large datasets, and network-based representations has become popularly. The `ProNet` package integrates several functional modules and provides a simple way for network construction, visualization, topological analyses and comparison, clustering, as well as biological functions statistics. This tutorial illustrates how to use this `ProNet` package based on the dataset from the work of *Lai et al.* (Navratil V *et al.*, 2009; Yan-Hua Lai *et al.*, 2012).

Contents

1 Quick start

To install ProNet run the following command within R:

```
> install.packages("ProNet")
```

To load ProNet into your current R session:

```
> library("ProNet")
```

A network can then be constructed either from experimental PPI data or a set of gene products and the integrated PPI database.

```
> nodes<-data.frame(c("1855","1856","1857"))
> network<-construction(input=nodes,db="Biogrid",species="human",ID.type="Entrez Gene",hierarchy=1)
```

Next, operation on the network including sub-network extraction and assembling can be done.

```
> net1<-extraction(network, mode="sample", sample.number=20)
> net2<-extraction(network, mode="exact", nodes=1:20)
> net3<-assemble(net1, net2, mode="union")
```

Plot of the networks can be achieved by the visulization module.

```
> visualization(network, layout="fruchterman.reingold",node.size=8,node.fill.color="red",node.border
```

We then take topological analyses or comparison on the networks.

```
> topology(network, simple.parameters=TRUE, degree.distribution=TRUE,clustering.coefficient=TRUE)
> net.comparing(net1,net2,topology.parameters=TRUE)
```

Functional modules can be achieved by clustering.

```
> cluster(network, method="MCODE", plot=TRUE, layout="fruchterman.reingold")
```

GO annotation and comparison can be performed on networks.

```
> enrichment.annotation(network, onto="MF", pvalue=0.05)
> go.profiles(V(net1)$name, V(net2)$name, onto="MF", mode="frequency", plot=TRUE)
```

2 Example Session

To illustrate the package we will construct the network based on the dataset obtained from Lai et al.'s work (Yan-Hua Lai *et al.*, 2012). The original H1N1 IAV-human PPI data was revised, and finally contained direct interaction both between the 10 IVA proteins and between them and another 104 human proteins through an overall 179 PPI. This local network was constructed, and then expanded to include those proteins that interact with IAV proteins through at least two direct partners of IAV. Visualization, topological analyses, and graph based clustering, GO analyses were then performed.

2.1 Network construction and operation

When constructing a network, the input data must be a data frame.

```
> library("ProNet")
> iavPath <-file.path(system.file("example",package="ProNet"),"iav.txt")
> iav <- read.table(iavPath, header=TRUE, sep="\t")
> head(iav)
```

```
  Gene_name_1 Adscription_1 Interaction_type Gene_name_2 Adscription_2
1          M1   IAV protein               pp      GNB2L1    DHP of IAV
2          M1   IAV protein               pp       VPS28    DHP of IAV
3          M1   IAV protein               pp       CDC42    DHP of IAV
4          M1   IAV protein               pp        C1QA    DHP of IAV
5          M1   IAV protein               pp       PRKRA    DHP of IAV
6          M1   IAV protein               pp      SDCBP2    DHP of IAV
```

At first, the local network of the 179 PPIs between 114 IVA or host proteins was constructed.

```
> g1 <- construction(iav[,c("Gene_name_1","Gene_name_2")],local.net=TRUE)
> sp <- unique(cbind(c(as.vector(iav[,"Gene_name_1"]),as.vector(iav[,"Gene_name_2"])),
+                    c(as.vector(iav[,"Adscription_1"]),as.vector(iav[,"Adscription_2"]))))
> V(g1)$species <- sp[,2]
> summary(g1)
```

```
IGRAPH UN-- 114 179 --
+ attr: name (v/c), species (v/c)
```

Second, the non-local network between host proteins that interact with nodes having at least IAV protein partners was constructed based on the integrated Biogrid database.

```
> hostPath <-file.path(system.file("example",package="ProNet"),"host.txt")
> host <- read.table(hostPath, header=TRUE, sep="\t")
> g2 <- construction(input=as.data.frame(unique(host[,"Protein.name"])),
+                    hierarchy=1,db="HPRD",species="human",ID.type="Gene symbol")
> summary(g2)
```

```
IGRAPH UN-- 687 3273 --
+ attr: name (v/c), vertex.hierarchy (v/n)
```

The expanded IAV-host network was then abtained by integrating these two networks.

```
> hprd <- construction(db="HPRD",ID.type= c("Gene symbol"))
> id <- match(unique(c(V(g1)$name,V(g2)$name)),V(hprd)$name)
> gtemp <- induced.subgraph(hprd, id[!is.na(id)])
> g3 <- assemble(g1,gtemp,mode="union")
> summary(g3)
```

```
IGRAPH UN-- 761 3731 --
+ attr: name (v/c), species (v/c)
```

2.2 Visulization of the network

The network can be viewed either by simple visualization:

```
> color <- rep(1,vcount(g3))
> color[V(g3)$species=="DHP of IAV"] <- "red"
> color[V(g3)$species=="IAV protein"] <- "black"
> color[is.na(V(g3)$species)] <- "green"
> visualization(g3,node.size=3,node.fill.color=color,node.label="",edge.color="gray")
> legend("topleft",col=c("black","red","green"),
+        legend=c("virus","human_direct","human_indirect"),pch=19)
```

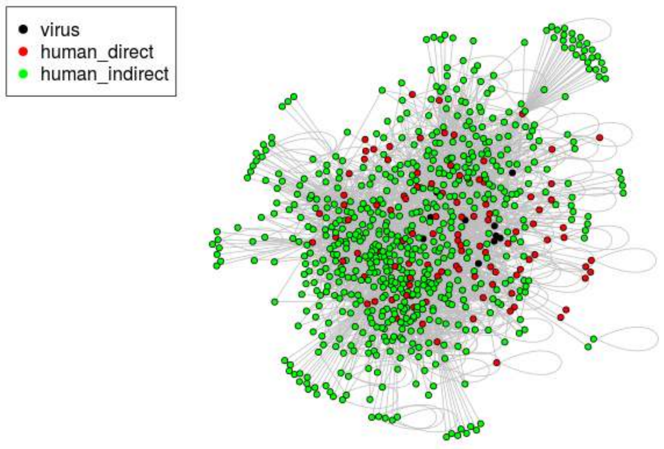

Figure 1: *Simple visualization of the constructed g3 network.*

Or subcellular localization based visualization:

```
> V(g3)$expression<-rexp(vcount(g3),1)
> location(g3,species=c("human"),vertex.size=3,vertex.label.cex=0.5,
+          vertex.color="expression",xlim=c(-1,1),ylim=c(-1,1))
```

2.3 Topological analyses

Overall statistics of the network's topology parameters can be retrieved by:

```
> topology(g3,simple.parameters=TRUE)
```

```
Simple statistics of the network:
 Number of nodes :  761 ;
 Number of edges :  3731 ;
 Connected components :  1 ;
 Isolated nodes :  0 ;
 Number of self-loops :  295 ;
 Average number of neighbors :  8.980289 ;
 Average path length :  3.192119 ;
 Network diameter :  5 ;
```

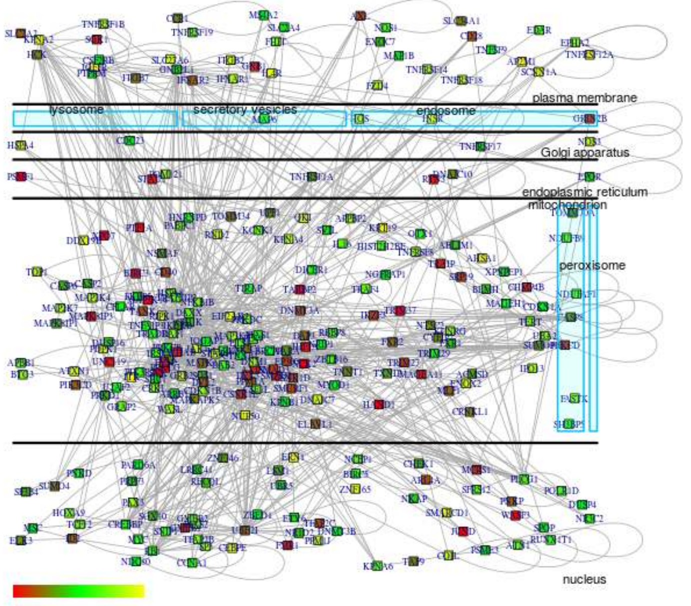

Figure 2: *Subcellular localization based visualization of the constructed g3 network.*

```
 Density :  0.012902 ;
 Cluster coefficient :  0.1231016 ;
$simple
             Number of nodes              Number of edges         Connected components
                 761.0000000                 3731.0000000                    1.0000000
              Isolated nodes         Number of self-loops Avgerage number of neighbors
                   0.0000000                  295.0000000                    8.9802891
         Average path length             Network diameter                      Density
                   3.1921191                    5.0000000                    0.0129020
         Cluster coefficient
                   0.1231016
```

Specific statistics of the degree distribution.

```
> tp <- topology(g2,degree.distribution=TRUE)
> head(as.data.frame(tp))
```

```
       degree.Node.name degree.Degree degree.Degree.Distribution
PIK3R2           PIK3R2            32        0.00291120815138282
HDAC1             HDAC1            34        0.00582241630276565
CBL                 CBL            28        0.00582241630276565
PLCG1             PLCG1            29        0.00291120815138282
TYK2               TYK2            14         0.0218340611353712
MAPK1             MAPK1            45        0.00145560407569141
```

Other topological parameters like clustering coefficient, betweeness, shortest path, eigenvector centrality, connectivity and closeness can be obtained similarly by changing the default setting of the parameters to be TRUE.

```
> tp <- topology(g2,shortest.paths=TRUE)
> head(as.data.frame(tp))
```

```
  shortest.paths.Var1 shortest.paths.Freq
1                   1                2984
```

```
2                   2          38960
3                   3         101589
4                   4          77729
5                   5          13978
6                   6            401
```

Along with the returned list of statistics value, a plot is also provided.

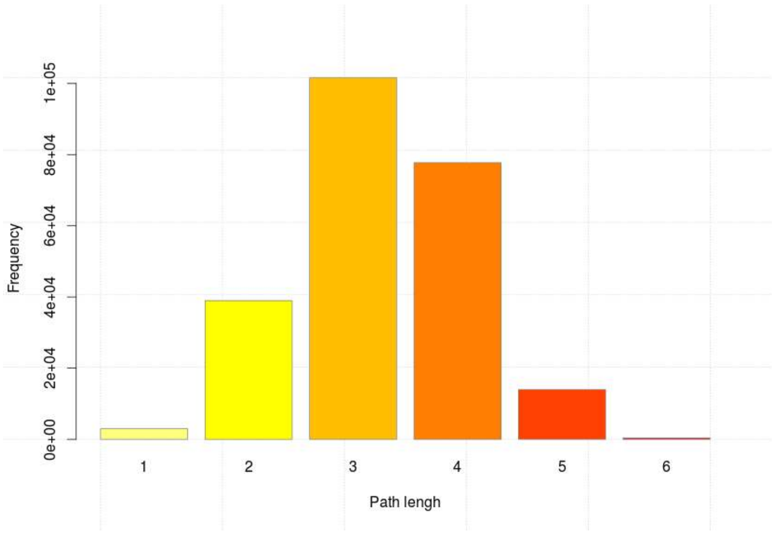

Figure 3: *Shortest path length distribution of the g2 network.*

2.4 Topological comparison of networks

It was also able to compare two networks' topological networks by either overall or topological parameter specific statistics.

```
> net.comparing(g3,hprd,topology.parameters=TRUE)
```

```
$topology
                                   g3        hprd
Number of nodes              761.0000   9617.0000
Number of edges             3731.0000  39240.0000
Isolated nodes                 0.0000      0.0000
Connected components           1.0000    262.0000
Network diameter               5.0000     14.0000
Average path length            3.1921      4.2093
Avg. number of neighbors       8.9803      7.7028
Ave. degree                    9.8055      8.1605
Avg. clustering coefficient    0.2220      0.1381
Avg. betweenness             833.0053  14179.5208
```

Mann-Whitney U-test was performed on the degree distribution of the two networks, the p-value and a plot was returned.

```
> net.comparing(g3,hprd,topology.parameters=FALSE,degree=TRUE)
```

```
$dg.p.value
[1] 3.117568e-19
```

The same procedure can be performed by setting the other topological parameters as TRUE.

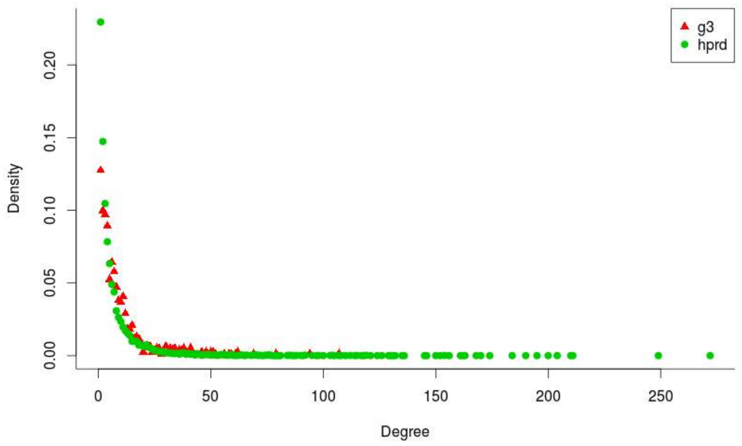

Figure 4: *Degree distribution comparison between the g3 and HPRD networks.*

```
> net.comparing(g3,hprd,topology.parameters=FALSE,degree=TRUE)

$dg.p.value
[1] 3.117568e-19
```

To test the significance of the IAV-host interaction network, g3 was compared with randomly selected ones from the whole HPRD network.

```
> comp.rand.subnet(g3,hprd,nsim=10000,ave.path.len=TRUE)

                    g3     hprd pvalue
Ave.path.len 3.192119 5.822729      1
```

The p-value and a plot of the mean degree distribution of sampled sub-networks would return. Similar results can also be obtained from other parameters comparison.

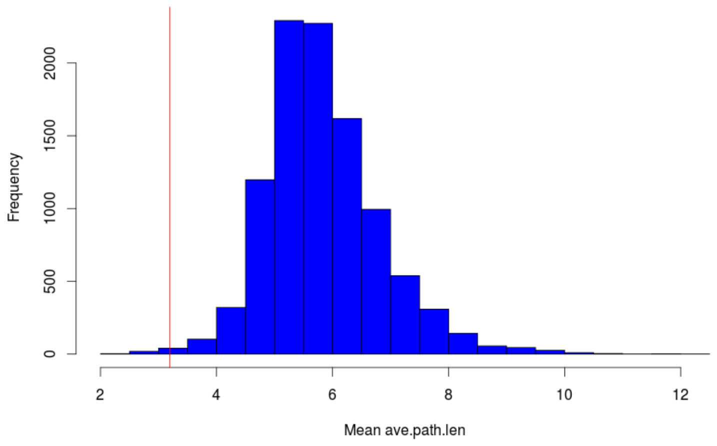

Figure 5: *Mean average path length distribution comparison between the g3 and randomly sampled HPRD networks.*

2.5 Network clustering

Several graph based network clustering algorithms were integrated into the package, such as the FN (A Clauset *et al.*, 2004), linkcomm (Kalinka *et al.*, 2011), MCL (van Dongen SM, 2000) and MCODE (Bader GD *et al.*, 2003) methods.

There are 7 clusters found by the FN method, and the number of nodes in each cluster is also shown.

```
> result <- cluster(g3, method="FN")
> clusters <- rep(1, vcount(g3))
> for(i in 1:vcount(g3)){clusters[i] <- result[[i]]}
> clusters <- as.factor(clusters)
> table(clusters)
```

```
clusters
  1   2   3   4   5   6   7
104 295  72  35 191  56   8
```

MCODE method can be perfomed using the individual mcode module. 11 clusters were found, with the largest containing 77 elements. Scores of each cluster were also shown.

```
> result <- mcode(g3,vwp=0.05,haircut=TRUE,fluff=FALSE,fdt=0.8,loops=FALSE)
> summary(result$COMPLEX)
```

```
      Length Class  Mode
 [1,] 41     -none- numeric
 [2,] 77     -none- numeric
 [3,]  5     -none- numeric
 [4,] 21     -none- numeric
 [5,]  4     -none- numeric
 [6,]  4     -none- numeric
 [7,]  3     -none- numeric
 [8,]  3     -none- numeric
 [9,]  3     -none- numeric
[10,]  3     -none- numeric
[11,]  4     -none- numeric
```

```
> result$score
```

```
 [1] 7.250000 5.184211 4.500000 4.300000 4.000000 3.333333 3.000000 3.000000 3.000000 3.000000
[11] 2.666667
```

The first cluster with the highest clustering score is shown.

```
> cluster1<-induced.subgraph(g3,result$COMPLEX[[1]])
> summary(cluster1)
```

```
IGRAPH UN-- 41 179 --
+ attr: name (v/c), species (v/c), expression (v/n)
```

```
> visualization(cluster1,node.size=4,node.label=V(cluster1)$name,node.label.color="blue")
```

2.6 GO enrichment and profiling for clusters

At first, the HPRD id table should be read and prepared for the following conversion.

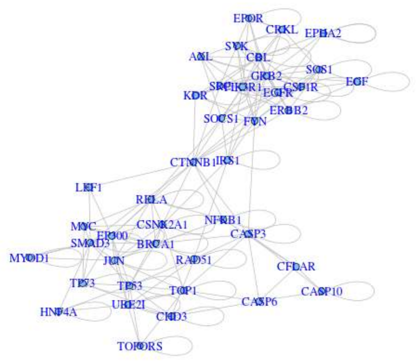

Figure 6: *The cluster with the highest clustering score.*

```
> idPath <-file.path(system.file("example",package="ProNet"),"hprd.id.txt")
> id <- read.table(idPath, header=FALSE, sep="\t")
> colnames(id) <- c("geneSymbol","entrezgene_id")
> head(id)
```

```
  geneSymbol entrezgene_id
1    ALDH1A1           216
2        FAU          2197
3      ALDH2           217
4    ALDH3A1           218
5    ALDH1B1           219
6      ACAT2            39
```

Node labels of the networks should be converted to Entrez Gene ID and then GO enrichment is performed.

```
> index1 <- match(V(cluster1)$name, as.vector(id$geneSymbol), nomatch=0)
> entrez1 <- as.vector(id$entrezgene_id[index1])
> go.mf <- enrichment.annotation(entrez1, onto="MF", pvalue=0.05)
> head(go.mf[,c("GO_ID","GO_term","p.value")])
```

```
       GO_ID                                         GO_term      p.value
1 GO:0004713                 protein tyrosine kinase activity 0.000000e+00
2 GO:0005515                                  protein binding 0.000000e+00
3 GO:0008134                      transcription factor binding 0.000000e+00
4 GO:0044212          transcription regulatory region DNA binding 0.000000e+00
5 GO:0004714 transmembrane receptor protein tyrosine kinase activity 8.659740e-15
6 GO:0003677                                      DNA binding 1.498801e-14
```

GO profiling can be performed either on a single network.

```
> go.profiles(entrez1, onto="MF",main="cluster1")
```

```
                                                     GO_term       GOID entrez1
1                                            protein binding GO:0005515      39
2                                                DNA binding GO:0003677      17
3                                                ATP binding GO:0005524      16
4                                         nucleotide binding GO:0000166      14
5  sequence-specific DNA binding transcription factor activity GO:0003700      14
```

```
6                              protein tyrosine kinase activity GO:0004713      11
7                                  transcription factor binding GO:0008134      10
8                   transcription regulatory region DNA binding GO:0044212      10
9                                      identical protein binding GO:0042802       9
10                                             metal ion binding GO:0046872       9
```

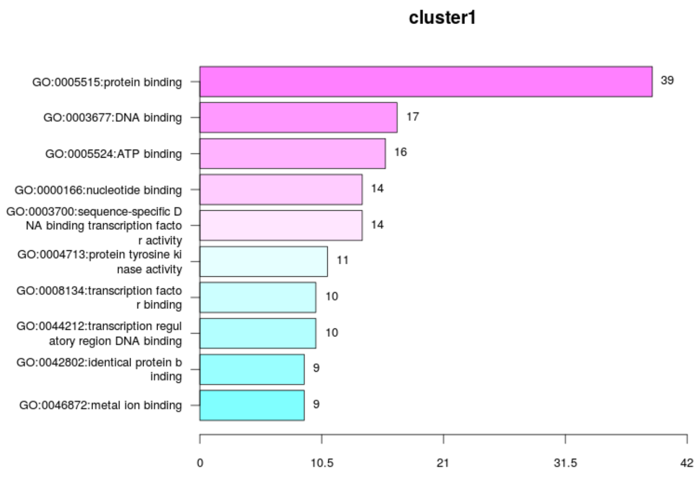

Figure 7: *GO profiling for a single cluster.*

Or comparising between two networks.

```
> cluster2<-induced.subgraph(g3,result$COMPLEX[[2]])
> index2 <- match(V(cluster2)$name, as.vector(id$geneSymbol), nomatch=0)
> entrez2 <- as.vector(id$entrezgene_id[index2])
> go.profiles(entrez1,entrez2,onto="MF",main=c("cluster1 vs 2"))
```

```
                                                         GO_term       GOID entrez1 entrez2
1                                               protein binding GO:0005515      39      61
2                                                   DNA binding GO:0003677      17      16
3                                                   ATP binding GO:0005524      16      21
4                                            nucleotide binding GO:0000166      14      20
5  sequence-specific DNA binding transcription factor activity GO:0003700      14      12
6                              protein tyrosine kinase activity GO:0004713      11      10
7                                  transcription factor binding GO:0008134      10       6
8                   transcription regulatory region DNA binding GO:0044212      10       2
9                                      identical protein binding GO:0042802       9       7
10                                             metal ion binding GO:0046872       9      24
```

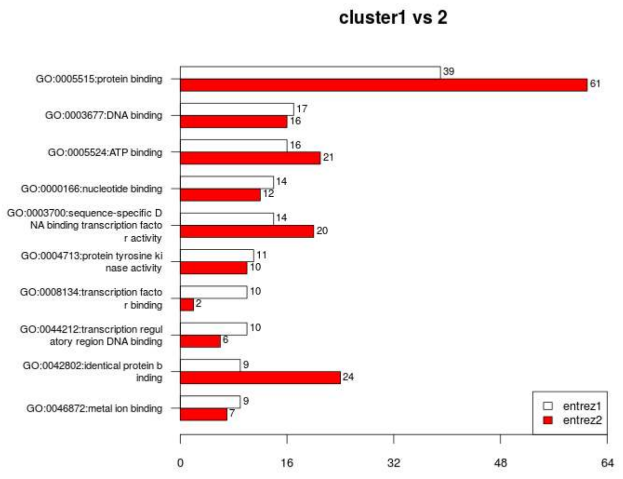

Figure 8: *GO comparing for a single cluster.*

References

A Clauset, MEJ Newman, C (2004) Moore: Finding community structure in very large networks. `http://www.arxiv.org/abs/cond-mat/0408187`.

Bader GD, Hogue CW. (2003) An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, **4(1)**: 2.

Kalinka, A.T. and Tomancak, P. (2011) linkcomm: an R package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type. *Bioinformatics*, **27(14)**: 2011-2012.

Navratil V, de Chassey B, Meyniel L, Delmotte S, Gautier C, Andre P, et al. (2009) VirHostNet: a knowledge base for the management and the analysis of proteome-wide virus–host interaction networks. *Nucleic Acids Res*, **37**: 661-8.

van Dongen, S.M. (2000) Graph Clustering by Flow Simulation. Ph.D. thesis, Universtiy of Utrecht.

Yan-Hua Lai, Zhan-Chao Li, Li-Li Chen, Zong Dai, Xiao-Yong Zou. (2012) Identification of potential host proteins for influenza A virus based on topological and biological characteristics by proteome-wide network approach. *Journal of Proteomics*, **75(5)**: 2500-2513.