

Package ‘ProNet’

October 22, 2015

Type Package

Title Biological Network Construction, Visualization and Analyses

Version 1.0.0

Date 2015-10-16

Author Xiang-Yun Wu and Xia-Yu Xia

Maintainer Xia-Yu Xia <xiaxiayu.thu@hotmail.com>

Description High-throughput experiments are now widely used in biological researches, which improves both the quality and quantity of omics data. Network-based presentation of these data has become a popular way in data analyses. This package mainly provides functions for biological network construction, visualization and analyses. Networks can be constructed either from experimental data or from a set of proteins and integrated PPI database. Based on them, users can perform traditional visualization, along with the subcellular localization based ones for Homo sapiens and Arabidopsis thaliana. Furthermore, analyses including topological statistics, functional module clustering and go profiling can also be achieved.

Depends R (>= 3.1.0), Rcpp (>= 0.10.4), igraph (>= 0.5.5-2), MCL, linkcomm

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-10-22 22:14:27

R topics documented:

ProNet-package	2
assemble	3
ath	4
cluster	4
comp.rand.subnet	6
comp.subnet	7
construction	8
construct_local	9

construct_nlocal	10
enrichment.annotation	11
extraction	11
go.profiles	12
GO_component	13
GO_function	13
GO_process	14
graph.add	15
HPRD	15
human	16
location	16
locDB	17
mcode	18
net.attack	19
net.comparing	20
net.robustness	21
topology	22
topology_anc	23
topology_betweenness	24
topology_closeness	24
topology_cluster_coefi	25
topology_degree	25
topology_ec	26
topology_shortest_paths	27
topology_simple	27
visualization	28
Index	30

Description

High-throughput experiments are now widely used in biological researches, which improves both the quality and quantity of omics data. Network-based presentation of these data has become a popular way in data analyses. This package mainly provides functions for biological network construction, visualization and analyses. Networks can be constructed either from experimental data or from a set of proteins and integrated PPI database. Based on them, users can perform traditional visualization, along with the subcellular location based ones for *Homo sapiens* and *Arabidopsis thaliana*. Furthermore, analyses including topological statistics, functional module clustering and go profiling can also be achieved.

Details

Package: ProNet
Type: Package
Version: 1.0.0
Date: 2015-10-16
License: GPL (>= 2)

Author(s)

Xiang-Yun Wu and Xia-Yu Xia
Maintainer: Xia-Yu Xia <xiaxiyu.thu@hotmail.com>

assemble *Operation of networks*

Description

Operation of networks including assemble them into a whole one, find their intersection, difference or complements.

Usage

```
assemble(..., mode = c("union", "intersection", "difference", "complementer"))
```

Arguments

... A list of igraph objects.
mode Operation mode, possible values are union, intersection, difference, and complementer.

Value

An integrated igraph object.

Examples

```
g1<-graph.ring(10)  
g2<-graph.tree(10,mode="undirected")  
assemble(g1,g2,mode="union")  
assemble(g1,g2,mode="intersection")  
assemble(g1,g2,mode="difference")  
assemble(g1,mode="complementer")
```

ath *Arabidopsis thaliana Biogrid PPI database*

Description

The Arabidopsis thaliana protein-protein interactions obtained from the Biogrid database.

Usage

```
data(ath)
```

Format

A data frame with 39901 rows and 4 columns.

Source

<http://thebiogrid.org/>

References

Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M (Jan 2006). BioGRID: A General Repository for Interaction Datasets. Nucleic Acids Research 1 (34): 535-539.

cluster *Network clustering*

Description

Network clustering based on different methods.

Usage

```
cluster(graph, method = c("FN", "MCL", "LINKCOMM", "MCODE"), expansion = 2,  
inflation = 2, hcmethod = "average", directed = FALSE, outfile = NULL,  
plot = TRUE, layout = c("reingold.tilford", "circle", "random",  
"fruchterman.reingold", "sphere", "kamada.kawai", "lgl"), ...)
```

Arguments

graph	An igraph object.
method	Clustering method, possible values are FN, MCL, LINKCOMM and MCODE.
expansion	Numeric value > 1 for the expansion parameter, if method is MCL. See MCL for more information.
inflation	Numeric value > 0 for the inflation power coefficient, if method is MCL. See MCL for more information.
hcmethod	A character string naming the hierarchical clustering method to use. Default value is average. See linkcomm for more information.
directed	Logical value, indicating whether the network is directed (if TRUE) or not (if FALSE).
outfile	File to save the clustering result.
plot	Logical value, indicating whether to plot summary output (if TRUE) or not (if FALSE).
layout	Mode of the layout, possible values are fruchterman.reingold, reingold.tilford, random, circle, kamada.kawai, lgl and sphere. See igraph for more information.
...	Other arguments.

Value

A summary and visualization of the clustering.

References

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

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

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.

Bader GD, Hogue CW. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*. 2003 Jan 13;4(1):2.

See Also

[mcode](#)

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=nlocal, db="HPRD", species="human", ID.type="Gene symbol", hierarchy=1)
cluster(net, method="MCODE", layout="fruchterman.reingold")
cluster(net, method="FN", layout="fruchterman.reingold")
```

comp.rand.subnet *Comparing a sub network to the randomly simulated ones*

Description

Comparing a sub network with randomly simulated ones from the whole network.

Usage

```
comp.rand.subnet(subgraph, graph, nsim = 1000, degree = FALSE,
  betweenness = FALSE, ave.path.len = FALSE, eccentricity = FALSE,
  cc = FALSE, method = "utest", FDR = 0.05)
```

Arguments

subgraph	An igraph object.
graph	An igraph object. The whole one for random simulation.
nsim	Times for simulation. Default value is 1000.
degree	Logical value, indicating whether to do vertex degree comparing (if TRUE) or not (if FALSE).
betweenness	Logical value, indicating whether to do betweenness comparing (if TRUE) or not (if FALSE).
ave.path.len	Logical value, indicating whether to do average path comparing (if TRUE) or not (if FALSE).
eccentricity	Logical value, indicating whether to do eccentricity comparing (if TRUE) or not (if FALSE).
cc	Logical value, indicating whether to do clustering coefficient comparing (if TRUE) or not (if FALSE).
method	Test method, currently only utest is supported.
FDR	False discovery rate. Default value is 0.05.

Value

A matrix of compared parameters and plots.

References

Y Benjamini, Y Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 57, No. 1. (1995), pp. 289-300.

See Also

[net.comparing](#), [comp.subnet](#)

Examples

```
g<-barabasi.game(100,power=0.8,directed=FALSE)
subg<-induced.subgraph(g,sample(1:100,30))
comp.rand.subnet(subg,g)
comp.rand.subnet(subg,g,degree=TRUE)
```

 comp.subnet

Comparing a sub network to the whole one

Description

Comparing the topological parameters of a sub network and the whole one.

Usage

```
comp.subnet(x, graph, topology.parameters = TRUE, degree = FALSE,
  cc = FALSE, betweenness = FALSE, eccentricity = FALSE,
  ave.path.len = FALSE, figure.type = 1, method = "utest",
  legendname = c(substitute(x), substitute(graph)))
```

Arguments

x	Vertex of the sub network.
graph	An igraph object.
topology.parameters	Logical value, indicating whether to do basic comparing (if TRUE) or not (if FALSE).
degree	Logical value, indicating whether to do degree comparing (if TRUE) or not (if FALSE).
cc	Logical value, indicating whether to do clustering coefficient comparing (if TRUE) or not (if FALSE).
betweenness	Logical value, indicating whether to do betweenness comparing (if TRUE) or not (if FALSE).
eccentricity	Logical value, indicating whether to do eccentricity comparing (if TRUE) or not (if FALSE).
ave.path.len	Logical value, indicating whether to do average path comparing (if TRUE) or not (if FALSE).
figure.type	Type of the plot. See plot for more information.
method	Test method, currently only utest is supported.
legendname	Legend name for the plot.

Value

A list of compared parameters and plot.

See Also

[comp.rand.subnet](#), [net.comparing](#)

Examples

```
g<-barabasi.game(100,power=0.8,directed = FALSE)
id<-sample(1:100, 20)
res<-comp.subnet(id,g)
res<-comp.subnet(id,g,topology.parameters=TRUE)
```

construction

Network construction

Description

Construction a network from experimental data or integrated PPI database.

Usage

```
construction(input, local.net = FALSE, node.attribute = NULL,
  db = c("Biogrid", "HPRD"), species = c("human", "ath"),
  ID.type = c("Gene symbol", "Entrez Gene"), hierarchy = 1)
```

Arguments

input	A data frame containing the experimental data.
local.net	Logical value, indicating whether to construct a network from experimental data (if TRUE) or not (if FALSE). Default value is FALSE.
node.attribute	A data frame containing node attributes. Default value is NULL.
db	Integrated PPI database, either Biogrid or HPRD .
species	This parameter indicates the biological species to which analyzable PPI data is related; currently human for "Homo sapiens" and ath for "Arabidopsis thaliana" are available.
ID.type	The ID type of the biological genes or proteins, possible values are Entrez gene and Gene symbol when db is Biogrid, or Gene symbol when db is HPRD .
hierarchy	This parameter indicates how many hierarchy are included in the network, currently it can be 0, 1 or 2. Default value is 1.

Value

A network in igraph format.

See Also

[construct_local](#), [construct_nlocal](#)

Examples

```
## Construction a local network.
local<-data.frame(1:5,2:6)
attribute<-data.frame(1:6,c(2.2,5.3,1.2,4.5,6.2,0.6))
net<-construction(input=local,local.net=TRUE,node.attribute=attribute)
## Construction a network from the human HPRD database.
nlocal<-data.frame(c("DVL1","DVL2","DVL3"))
net<-construction(input=nlocal,db="HPRD",species="human",ID.type="Gene symbol",hierarchy=1)
```

construct_local	<i>Local network construction</i>
-----------------	-----------------------------------

Description

Construction a network from experimental data.

Usage

```
construct_local(input, node.attribute)
```

Arguments

input A data frame containing the experimental data.
node.attribute A data frame containing node attributes.

Value

A network in igraph format.

See Also

[construction](#), [construct_nlocal](#).

Examples

```
local<-data.frame(1:5,2:6)
attribute<-data.frame(c(0.2,0.3,0.2,0.5,0.1))
net<-construct_local(input=local,node.attribute=attribute)
```

construct_nlocal	<i>Non-local network construction</i>
------------------	---------------------------------------

Description

Construction a network from integrated PPI database.

Usage

```
construct_nlocal(input, db = c("Biogrid", "HPRD"), species = c("human",
  "ath"), ID.type = c("Gene symbol", "Entrez Gene"),
  hierarchy = 1)
```

Arguments

input	A data frame containing the experimental data.
db	Integrated PPI database, either Biogrid or HPRD .
species	This parameter indicates the biological species to which analyzable PPI data is related; currently human for "Homo sapiens" and ath for "Arabidopsis thaliana" are available.
ID.type	The ID type of the biological genes or proteins, possible values are Entrez gene and Gene symbol when db is Biogrid, or Gene symbol when db is HPRD .
hierarchy	This parameter indicates how many hierarchy are included in the network, currently it can be 0, 1 or 2. Default value is 1.

Value

A network in igraph format.

See Also

[construction](#), [construct_local](#).

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construct_nlocal(input=nlocal, db="HPRD", species="human", ID.type="Gene symbol", hierarchy=1)
```

enrichment.annotation *GO enrichment annotation*

Description

GO enrichment annotation of the functional molecules or networks.

Usage

```
enrichment.annotation(data, onto = c("MF", "BP", "CC"), pvalue = 0.05)
```

Arguments

data	An igraph object or vector of vertex names.
onto	GO categories, three possible values are MF for GO function, BP for GO process, and CC for GO componet
pvalue	Significant level. Default value is 0.05.

Value

Numeric vector, pvalue with the length as the size of data.

References

The Gene Ontology Consortium (January 2008). The Gene Ontology project in 2008. *Nucleic Acids Res.* 36 (Database issue): D440?C4.

Examples

```
entrez<-data.frame(c("121549", "51160", "83878", "11338", "196477", "9319", "608", "7015"))
net<-construction(input=entrez, hierarchy=0, species="human", db="Biogrid", ID.type="Entrez Gene")
res<-enrichment.annotation(net, pvalue=0.05, onto="CC")
```

extraction *Subnetwork extraction*

Description

Extract a sub network from the original one.

Usage

```
extraction(graph, mode = c("exact", "sample"), sample.number = 10,
  nodes = NULL, vertex.name = NULL, vertex.expression = NULL, ...)
```

Arguments

graph	An igraph object.
mode	Extraction mode, either be exact or sample.
sample.number	Sampled nodes if mode is sample, no more than the total nodes in the network. Default value is 10.
nodes	A vector of vertex id to extract the subnetwork if mode is exact.
vertex.name	A vector of vertex name to extract the subnetwork if mode is exact.
vertex.expression	Attribute used to choose the vertex and extract the subnetwork.
...	Other vertex or edge attributes.

Value

An igraph object.

Examples

```
g<-graph.ring(100)
g1<-extraction(g,mode="sample",sample.number=5)
```

go.profiles

GO enrichment comparison

Description

GO enrichment comparison between networks.

Usage

```
go.profiles(g_ref, g_comp, color = c("red", "white"), onto = c("MF", "BP",
  "CC"), mode = c("number", "frequency"), demonstration.number = 10,
  plot = TRUE, ...)
```

Arguments

g_ref	Vector of the first set of molecules.
g_comp	Vector of the second set of molecules.
color	Color represent of the two sets.
onto	GO categories, three possible values are MF for GO function, BP for GO process, and CC for GO componet
mode	Mode of the GO vlues, either Frequency or Number.
demonstration.number	Number of the top GOs to display.
plot	Logical value, whether to plot the profiling result (if TRUE) or not (if FALSE).
...	other arguments.

Value

A data frame of the profiling result and plots.

Examples

```
entrez1<-c("11067", "414157", "196477", "147339", "642")
entrez2<-c("121549", "51160", "83878", "11338", "196477", "9319", "608", "7015")
go.profiles(entrez1, onto="MF", main="Only Network 1")
go.profiles(entrez1, entrez2, onto="MF", main=c("Network 1 vs 2"))
```

GO_component

GO component annotation database

Description

The gene products' GO component annotation data retrieved from the Gene Ontology Consortium.

Usage

```
data(GO_component)
```

Format

A data frame with 92428 rows and 3 columns.

Source

<http://geneontology.org/>

References

The Gene Ontology Consortium (January 2008). The Gene Ontology project in 2008. *Nucleic Acids Res.* 36 (Database issue): 440-4.

GO_function

GO function annotation database

Description

The gene products' GO function annotation data retrieved from the Gene Ontology Consortium.

Usage

```
data(GO_function)
```

Format

A data frame with 100475 rows and 3 columns.

Source

<http://geneontology.org/>

References

The Gene Ontology Consortium (January 2008). The Gene Ontology project in 2008. *Nucleic Acids Res.* 36 (Database issue): 440-4.

GO_process

GO process annotation database

Description

The gene products' GO process annotation data retrieved from the Gene Ontology Consortium.

Usage

```
data(GO_process)
```

Format

A data frame with 128103 rows and 3 columns.

Source

<http://geneontology.org/>

References

The Gene Ontology Consortium (January 2008). The Gene Ontology project in 2008. *Nucleic Acids Res.* 36 (Database issue): 440-4.

graph.add	<i>Network union</i>
-----------	----------------------

Description

Assemble a list of networks into a whole one.

Usage

```
graph.add(...)
```

Arguments

... A list of igraph objects.

Value

An integrated igraph object.

Examples

```
g1<-graph.ring(10)
g2<-graph.tree(10,mode="undirected")
g<-graph.add(g1,g2)
```

HPRD	<i>HPRD PPI database</i>
------	--------------------------

Description

The human protein-protein interactions obtained from the HPRD database.

Usage

```
data(HPRD)
```

Format

A data frame with 39240 rows and 2 columns.

Source

<http://www.hprd.org/>

References

Peri S, et al. (2003). Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Research* 13: 2363-71.

human	<i>Homo sapiens Biogrid PPI database</i>
-------	--

Description

The Homo sapien protein-protein interactions obtained from the Biogrid database.

Usage

```
data(human)
```

Format

A data frame with 250802 rows and 4 columns.

Source

<http://thebiogrid.org/>

References

Stark C, Breitkreutz BJ, Reguly T, Boucher L, Breitkreutz A, Tyers M (Jan 2006). BioGRID: A General Repository for Interaction Datasets. *Nucleic Acids Research* 1 (34): 535-539.

location	<i>Hierarchical plot of biological network</i>
----------	--

Description

Hierarchical plot of biological network according to the elements' subcellular location.

Usage

```
location(graph, species = c("human", "ath"), vertex.color = 1,  
  vertex.size = 10, vertex.shape = "square", edge.color = 1,  
  label.add = TRUE, colorbar.add = TRUE, xlim = c(-1, 1), ylim = c(-1,  
  1), ...)
```


Arguments

<code>graph</code>	An igraph object.
<code>species</code>	The species name, currently only human and ath are available.
<code>vertex.color</code>	Color of the vertex. Default value is 1.
<code>vertex.size</code>	Size of the vertex. Default value is 10.
<code>vertex.shape</code>	Shape of the vertex. Default value is square.
<code>edge.color</code>	Color of the edge. Default value is 1.
<code>label.add</code>	Boolean value, whether to add label to the plot (if TRUE) or not (if FALSE).
<code>colorbar.add</code>	Boolean value, whether to add colorbar to the plot (if TRUE) or not (if FALSE).
<code>xlim</code>	A vector indicating the range of x axis.
<code>ylim</code>	A vector indicating the range of y axis.
<code>...</code>	Other arguments.

Value

A hierarchical plot of biological network.

References

Barsky A, Gardy JL, Hancock REW, and Munzner T. (2007) Cerebral: a Cytoscape plugin for layout of and interaction with biological networks using subcellular localization annotation. *Bioinformatics* 23(8):1040-2.

Examples

```
gene<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=gene,db="HPRD",species="human",ID.type="Gene symbol",hierarchy=1)
location(net,species="human",vertex.color="vertex.hierarchy")
```

locDB

Gene products' location database

Description

The human's and ath's gene products' subcellular location data.

Usage

```
data(locDB)
```

Format

A list of 2 data frames.

Source

<https://www.rostlab.org/services/locDB/>

References

Rastogi, Shruti; Rost Burkhard (Jan 2011). LocDB: experimental annotations of localization for Homo sapiens and Arabidopsis thaliana. Nucleic Acids Res. 39 (Database issue): 230-4.

mcode

MCODE network clustering

Description

Clustering of the network using the MCODE method.

Usage

```
mcode(graph, vwp = 0.5, haircut = FALSE, fluff = FALSE, fdt = 0.8,
       loops = TRUE)
```

Arguments

graph	An igraph object.
vwp	Vertex weight percentage. Default value is 0.5.
haircut	Boolean value, whether to remove singly-connected nodes from clusters (TRUE) or not (FALSE).
fluff	Boolean value, whether to spand cluster cores by one neighbour shell outwards (TRUE) or not (FALSE).
fdt	Cluster density cutoff. Default value is 0.8.
loops	Boolean value, whether to include self-loops (TRUE) or not (FALSE).

Value

A list of clusters.

References

Bader GD, Hogue CW. An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics. 2003 Jan 13;4(1):2.

See Also

[cluster](#)

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=nlocal,db="HPRD",species="human",ID.type="Gene symbol",hierarchy=1)
mcode(net,vwp=0.9,haicut=TRUE,fluff=TRUE,fdt=0.1)
```

net.attack	<i>Network attack</i>
------------	-----------------------

Description

Retrieve the network after vertex attack.

Usage

```
net.attack(graph, mode = c("exact", "sample"), sample.number = 10,
  nodes = NULL, vertex.name = NULL, ...)
```

Arguments

graph	An igraph object.
mode	Attack mode, possible values are exact and sample.
sample.number	Number of attacked vertex, if mode is sample.
nodes	Index of attacked vertex, if mode is exact.
vertex.name	Name of attacked vertex, if mode is exact.
...	Other arguments.

Value

An igraph object.

Examples

```
g<-barabasi.game(100,power=0.8,directed=FALSE)
g1<-net.attack(g,mode="sample",sample.number=10)
g1<-net.attack(g,mode="exact",nodes=sample(1:100,20))
```

net.comparing *Comparing two networks*

Description

Comparing two networks' topological parameters.

Usage

```
net.comparing(graph1, graph2, topology.parameters = FALSE, degree = FALSE,
              cc = FALSE, betweenness = FALSE, eccentricity = FALSE,
              ave.path.len = FALSE, figure.type = 1, method = "utest", legendname)
```

Arguments

graph1	An igraph object.
graph2	An igraph object.
topology.parameters	Logical value, indicating whether to do basic comparing (if TRUE) or not (if FALSE).
degree	Logical value, indicating whether to do degree comparing (if TRUE) or not (if FALSE).
cc	Logical value, indicating whether to do clustering coefficient comparing (if TRUE) or not (if FALSE).
betweenness	Logical value, indicating whether to do betweenness comparing (if TRUE) or not (if FALSE).
eccentricity	Logical value, indicating whether to do eccentricity comparing (if TRUE) or not (if FALSE).
ave.path.len	Logical value, indicating whether to do average path comparing (if TRUE) or not (if FALSE).
figure.type	Type of the plot. See plot for more information.
method	Test method, only utest is supported.
legendname	Legend name for graph.

Value

A list of compared results and plot.

References

Dehmer,M. et al. A large scale analysis of information-theoretic network complexity measures using chemical structures. PLoS ONE, (2009a), 4, e8057.

See Also

[comp.subnet](#) and [comp.rand.subnet](#).

Examples

```
g1<-barabasi.game(100,power=0.8,directed = FALSE)
g2<-erdos.renyi.game(100,p=0.01,directed = FALSE)
res<-net.comparing(g1,g2,degree=TRUE,cc=TRUE,betweenness=FALSE)
```

net.robustness	<i>Network robustness</i>
----------------	---------------------------

Description

Test the network robustness after attack.

Usage

```
net.robustness(graph, n = 1000, vertex.name = NULL, ...)
```

Arguments

graph	An igraph object.
n	Number of sampling times. Default value is 1000.
vertex.name	A vector of the vertex name.
...	Other arguments.

Value

A matrix of the network parameters by exact and the simulated random attack.

See Also

[net.attack](#).

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=nlocal,db="HPRD",species="human",ID.type="Gene symbol",hierarchy=1)
net.robustness(net,n=1000,vertex.name=c("DVL1", "DVL2", "DVL3"))
```

topology

Network topology analysis

Description

Calculate the network or graph's topological parameters like degree distribution, clustering coefficient, betweenness, closeness, shortest paths, eigenvector centrality and connectivity.

Usage

```
topology(graph, simple.parameters = FALSE, degree.distribution = FALSE,
power.law = TRUE, fit.line = FALSE, clustering.coefficient = FALSE,
betweenness = FALSE, shortest.paths = FALSE, closeness = FALSE,
eigenvector.centrality = FALSE, connectivity = FALSE)
```

Arguments

graph	An igraph object.
simple.parameters	Logical value, indicating whether to do basic statistics (if TRUE) or not (if FALSE).
degree.distribution	Logical value, indicating whether to do degree distribution statistics (if TRUE) or not (if FALSE).
power.law	Logical value, indicating whether the log ratio would be calculated in degree distribution statistics (if TRUE) or not (if FALSE).
fit.line	Logical value, indicating whether to do line fitting in degree distribution statistics (if TRUE) or not (if FALSE).
clustering.coefficient	Logical value, indicating whether to do clustering.coefficient statistics (if TRUE) or not (if FALSE).
betweenness	Logical value, indicating whether to do betweenness statistics (if TRUE) or not (if FALSE).
shortest.paths	Logical value, indicating whether to do shortest.paths statistics (if TRUE) or not (if FALSE).
closeness	Logical value, indicating whether to do closeness statistics (if TRUE) or not (if FALSE).
eigenvector.centrality	Logical value, indicating whether to do eigenvector.centrality statistics (if TRUE) or not (if FALSE).
connectivity	Logical value, indicating whether to do connectivity statistics (if TRUE) or not (if FALSE).

Value

A list of topological parameters and plots.

References

Y Benjamini, Y Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 57, No. 1. (1995), pp. 289-300.

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=nlocal,db="HPRD",species="human",ID.type="Gene symbol",hierarchy=1)
tp<-topology(net,simple.parameters=TRUE)
tp<-topology(net,degree.distribution=TRUE)
tp<-topology(net,simple.parameters=TRUE,degree.distribution=TRUE)
```

topology_anc

Average neighborhood connectivity statistics

Description

Average neighborhood connectivity statistics of the network.

Usage

```
topology_anc(graph)
```

Arguments

graph An igraph object.

Value

A data frame containing the average neighborhood connectivity information and plots.

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=nlocal,db="HPRD",species="human",ID.type="Gene symbol",hierarchy=1)
anc<-topology_anc(net)
```

topology_betweenness *Betweenness statistics*

Description

Betweenness statistics of the network.

Usage

```
topology_betweenness(graph)
```

Arguments

graph An igraph object.

Value

A data frame containing the betweenness distribution and plots.

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=nlocal, db="HPRD", species="human", ID.type="Gene symbol", hierarchy=1)
b.g<-topology_betweenness(net)
```

topology_closeness *Closeness statistics*

Description

Closeness statistics of the network.

Usage

```
topology_closeness(graph)
```

Arguments

graph An igraph object.

Value

A data frame containing the closeness distribution and plots.

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=nlocal, db="HPRD", species="human", ID.type="Gene symbol", hierarchy=1)
c<-topology_closeness(net)
```

topology_cluster_coeffi
Clustering coefficient statistics

Description

Clustering coefficient statistics of the network.

Usage

```
topology_cluster_coeffi(graph)
```

Arguments

graph An igraph object.

Value

A data frame containing the clustering coefficient information and plots.

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=nlocal, db="HPRD", species="human", ID.type="Gene symbol", hierarchy=1)
cc<-topology_cluster_coeffi(net)
```

topology_degree *Degree statistics*

Description

Degree distribution statistics of the network.

Usage

```
topology_degree(graph, power.law = FALSE, fit.line = TRUE)
```

Arguments

graph An igraph object.

power.law Logical value indicating whether the log ratio would be calculated in degree distribution statistics (TRUE) or not (FALSE).

fit.line Logical value indicating whether to do line fitting in degree distribution statistics (TRUE) or not (FALSE).

Value

A data frame containing the vertex and degree information and plots.

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=nlocal,db="HPRD",species="human",ID.type="Gene symbol",hierarchy=1)
d<-topology_degree(net)
d<-topology_degree(net,power.law=TRUE)
```

topology_ec

Eigenvector centrality statistics

Description

Eigenvector centrality statistics of the network.

Usage

```
topology_ec(graph)
```

Arguments

graph An igraph object.

Value

A data frame containing the vertex eigenvector centrality information and plots.

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=nlocal,db="HPRD",species="human",ID.type="Gene symbol",hierarchy=1)
ec<-topology_ec(net)
```

topology_shortest_paths
Shortest path statistics

Description

Shortest path statistics of the network.

Usage

```
topology_shortest_paths(graph)
```

Arguments

graph An igraph object.

Value

An array containing the vertex path information and plots.

Examples

```
nlocal<-data.frame(c("DVL1", "DVL2", "DVL3"))
net<-construction(input=nlocal, db="HPRD", species="human", ID.type="Gene symbol", hierarchy=1)
p<-topology_shortest_paths(net)
```

topology_simple *Basic statistics*

Description

Basic statistics of the network's topological parameters.

Usage

```
topology_simple(graph)
```

Arguments

graph An igraph object.

Value

A vector containing the simple statistics.

Examples

```
nlocal<-data.frame(c("DVL1","DVL2","DVL3"))
net<-construction(input=nlocal,db="HPRD",species="human",ID.type="Gene symbol",hierarchy=1)
s<-topology_simple(net)
```

visualization

Network visualization

Description

This function takes an igraph formatted network as input and exports a picture of the visualization. Mode of the layout, and size, color, shape of the nodes, edges and labels can be set according to the experiment data.

Usage

```
visualization(graph, layout = c("reingold.tilford", "circle", "random",
  "fruchterman.reingold", "sphere", "kamada.kawai", "lgl"),
  node.fill.color = "SkyBlue2", node.border.color = "Black",
  node.shape = c("circle", "square", "sphere", "csquare", "rectangle",
  "crectangle", "vrectangle", "none", "pie", "raster"), node.size = 15,
  node.label = "name", node.label.color = 1, node.label.size = 0.8,
  node.label.position = 0, edge.shape = 1, edge.width = 1,
  edge.color = "gray1", ...)
```

Arguments

graph	An igraph object.
layout	Mode of the layout, possible values are fruchterman.reingold, reingold.tilford, random, circle, kamada.kawai, lgl and sphere. See igraph for more information.
node.fill.color	The fill color of the vertex. Default value is SkyBlue2.
node.border.color	Border color the the vertex. Default value is Black.
node.shape	Shape of the vertex. currently circle, square, csquare, rectangle, crectangl, vrectangle and none are supported.
node.size	Size of the vertex. Default value is 15.
node.label	Label of the vertex. Specify NA to omit vertex labels. Default value is the vertex name.
node.label.color	Color of labels of the vertex. Default value is 1.
node.label.size	Font size of the vertex label. Default value is 0.8.
node.label.position	Positon of the vertex label. Default value is 0.

<code>edge.shape</code>	Shape of the edge. Default value is 1.
<code>edge.width</code>	Width of the edge. Default value is 1.
<code>edge.color</code>	Color of the edge. Default value is gray1.
<code>...</code>	Other arguments.

Value

A plot of the network.

Examples

```
local<-data.frame(1:5,2:6)
attribute<-data.frame(node=1:6,value=c(2.2,5.3,1.2,4.5,6.2,0.6))
net<-construction(input=local,local.net=TRUE,node.attribute=attribute)
visualization(net,layout="reingold.tilford")
visualization(net,layout="reingold.tilford",node.size=V(net)$value)
```

Index

*Topic **datasets**

- ath, [4](#)
- GO_component, [13](#)
- GO_function, [13](#)
- GO_process, [14](#)
- HPRD, [15](#)
- human, [16](#)
- locDB, [17](#)

*Topic **package**

- ProNet-package, [2](#)

assemble, [3](#)

ath, [4](#)

cluster, [4](#), [18](#)

comp.rand.subnet, [6](#), [8](#), [21](#)

comp.subnet, [6](#), [7](#), [21](#)

construct_local, [8](#), [9](#), [10](#)

construct_nlocal, [8](#), [9](#), [10](#)

construction, [8](#), [9](#), [10](#)

enrichment.annotation, [11](#)

extraction, [11](#)

go.profiles, [12](#)

GO_component, [13](#)

GO_function, [13](#)

GO_process, [14](#)

graph.add, [15](#)

HPRD, [8](#), [10](#), [15](#)

human, [16](#)

location, [16](#)

locDB, [17](#)

mcode, [5](#), [18](#)

net.attack, [19](#), [21](#)

net.comparing, [6](#), [8](#), [20](#)

net.robustness, [21](#)

plot, [7](#), [20](#)

ProNet (ProNet-package), [2](#)

ProNet-package, [2](#)

topology, [22](#)

topology_anc, [23](#)

topology_betweenness, [24](#)

topology_closeness, [24](#)

topology_cluster_coeffi, [25](#)

topology_degree, [25](#)

topology_ec, [26](#)

topology_shortest_paths, [27](#)

topology_simple, [27](#)

visualization, [28](#)