

# Package ‘SpidermiR’

April 15, 2017

**Type** Package

**Title** SpidermiR: An R/Bioconductor package for integrative network analysis with miRNA data

**Version** 1.4.8

**Date** 04-11-2017

**Author** Claudia Cava,  
Antonio Colaprico,  
Alex Graudenzi,  
Gloria Bertoli,  
Tiago C. Silva,  
Catharina Olsen,  
Houtan Noushmehr,  
Gianluca Bontempi,  
Giancarlo Mauri,  
Isabella Castiglioni

**Maintainer** Claudia Cava <claudia.cava@ibfm.cnr.it>

**Depends** R (>= 3.0.0)

**Imports** networkD3, httr, igraph, utils, stats, miRNAatp, miRNAatp.db, AnnotationDbi, org.Hs.eg.db, ggplot2, gridExtra, gplots, grDevices, lattice, latticeExtra, visNetwork, TCGAbiolinks

**Description** The aims of SpidermiR are : i) facilitate the network open-access data retrieval from GeneMania data, ii) prepare the data using the appropriate gene nomenclature, iii) integration of miRNA data in a specific network, iv) provide different standard analyses and v) allow the user to visualize the results. In more detail, the package provides multiple methods for query, prepare and download network data (GeneMania), and the integration with validated and predicted miRNA data (mir-Walk, miR2Disease, miRTar, miRandola, Pharmaco-miR, DIANA, Miranda, PicTar and TargetScan) and the use of standard analysis (igraph) and visualization methods (networkD3).

**License** GPL (>= 3)

**biocViews** GeneRegulation, miRNA, Network

**Suggests** BiocStyle, knitr, rmarkdown, testthat, devtools, roxygen2

**VignetteBuilder** knitr

**LazyData** true

**URL** <https://github.com/claudiacava/SpidermiR>

**BugReports** <https://github.com/claudiacava/SpidermiR/issues>

**RoxygenNote** 6.0.1

**NeedsCompilation** no

## R topics documented:

Case_Study1_loading_1_network . . . . .	3
Case_Study1_loading_2_network . . . . .	3
Case_Study1_loading_3_network . . . . .	4
Case_Study1_loading_4_network . . . . .	4
Case_Study2_loading_1_network . . . . .	5
Case_Study2_loading_2_network . . . . .	6
Case_Study2_loading_3_network . . . . .	6
SpidermiR . . . . .	7
SpidermiRanalyze_Community_detection . . . . .	7
SpidermiRanalyze_Community_detection_bi . . . . .	8
SpidermiRanalyze_Community_detection_net . . . . .	8
SpidermiRanalyze_degree_centrality . . . . .	9
SpidermiRanalyze_DEnetworkTCGA . . . . .	9
SpidermiRanalyze_direct_net . . . . .	10
SpidermiRanalyze_direct_subnetwork . . . . .	11
SpidermiRanalyze_mirnanet_pharm . . . . .	11
SpidermiRanalyze_mirna_gene_complnet . . . . .	12
SpidermiRanalyze_mirna_network . . . . .	13
SpidermiRanalyze_subnetwork_neigh . . . . .	13
SpidermiRdownload_miRNAextra_cir . . . . .	14
SpidermiRdownload_miRNAprediction . . . . .	15
SpidermiRdownload_miRNAvalidate . . . . .	15
SpidermiRdownload_net . . . . .	16
SpidermiRdownload_pharmacomir . . . . .	16
SpidermiRprepare_NET . . . . .	17
SpidermiRquery_disease . . . . .	17
SpidermiRquery_networks_type . . . . .	18
SpidermiRquery_species . . . . .	18
SpidermiRquery_spec_networks . . . . .	19
SpidermiRvisualize_3Dbarplot . . . . .	19
SpidermiRvisualize_adj_matrix . . . . .	20
SpidermiRvisualize_BI . . . . .	21
SpidermiRvisualize_degree_dist . . . . .	21
SpidermiRvisualize_direction . . . . .	22
SpidermiRvisualize_mirnanet . . . . .	22
SpidermiRvisualize_plot_target . . . . .	23

**Index**

**24**

---

`Case_Study1_loading_1_network`*Loading 1 network of shared protein domain*

---

**Description**

`Case_Study1_loading_1_network` loads shared protein domain in HomoSapiens

**Usage**

```
Case_Study1_loading_1_network(species)
```

**Arguments**

`species`            `variable`

**Value**

dataframe with interactions

**Examples**

```
## Not run:  
a<-Case_Study1_loading_1_network(species)  
  
## End(Not run)
```

---

`Case_Study1_loading_2_network`*Loading 2 network of shared protein domain*

---

**Description**

`Case_Study1_loading_2_network` loads shared protein domain in HomoSapiens including only miRNAs already found deregulated in PC,

**Usage**

```
Case_Study1_loading_2_network(data)
```

**Arguments**

`data`                the output of `Case_Study1_loading_1_network`

**Value**

dataframe with selected interactions

**Examples**

```
## Not run:  
b<-Case_Study1_loading_2_network(data=a)  
  
## End(Not run)
```

---

Case\_Study1\_loading\_3\_network

*Loading 3 network of shared protein domain*

---

**Description**

Case\_Study1\_loading\_3\_network loads shared protein domain in HomoSapiens including only the DEGs with a direct interaction among them

**Usage**

```
Case_Study1_loading_3_network(data, dataFilt, dataClin)
```

**Arguments**

data	the output of Case_Study1_loading_2_network
dataFilt	TCGA matrix
dataClin	clinical data matrix

**Value**

dataframe with selected interactions

**Examples**

```
## Not run:  
c<-Case_Study1_loading_3_network(data=b,dataFilt=dataFilt,dataClin=dataClin)  
  
## End(Not run)
```

---

Case\_Study1\_loading\_4\_network

*Loading 4 network of shared protein domain*

---

**Description**

Case\_Study1\_loading\_4\_network loads network community with the higher number of elements in the 3 network

**Usage**

```
Case_Study1_loading_4_network(TERZA_NET)
```

### Arguments

TERZA\_NET      the output of Case\_Study1\_loading\_3\_network

### Value

dataframe with selected interactions

### Examples

```
## Not run:  
d<-Case_Study1_loading_4_network(TERZA_NET=c)  
  
## End(Not run)
```

---

Case\_Study2\_loading\_1\_network

*Loading 1 network of Protein Interactions (PI)*

---

### Description

Case\_Study2\_loading\_1\_network loads PI in HomoSapiens

### Usage

```
Case_Study2_loading_1_network(species)
```

### Arguments

species      variable

### Value

dataframe with interactions

### Examples

```
## Not run:  
a2<-Case_Study2_loading_1_network(species)  
## End(Not run)
```

---

Case\_Study2\_loading\_2\_network

*Loading 2 network of Protein Interactions (PI) with miRNAs*

---

### Description

Case\_Study2\_loading\_2\_network loads PI in HomoSapiens with miRNAs already found as deregulated in BC (only interaction miRNA-gene)

### Usage

```
Case_Study2_loading_2_network(data)
```

### Arguments

data                    output of Case\_Study2\_loading\_1\_network

### Value

dataframe with interactions

### Examples

```
## Not run:  
b2<-Case_Study2_loading_2_network(data=a2)  
  
## End(Not run)
```

---

Case\_Study2\_loading\_3\_network

*Loading 3 network of Protein Interactions (PI) with miRNAs*

---

### Description

Case\_Study2\_loading\_2\_network loads PI in HomoSapiens with miRNAs already found as deregulated in BC (only interaction miRNA-gene)

### Usage

```
Case_Study2_loading_3_network(sdas, miRNA_NET)
```

### Arguments

sdas                    output of Case\_Study2\_loading\_1\_network  
miRNA\_NET               output of Case\_Study2\_loading\_2\_network

### Value

dataframe with interactions

**Examples**

```
## Not run:
c2<-Case_Study2_loading_3_network(sdas=a2,miRNA_NET=b2)

## End(Not run)
```

---

 SpidermiR

*Download data*


---

**Description**

SpidermiR allows you to Download data of samples from GeneMania

**Details**

The functions you're likely to need from **SpidermiR** is SpidermiR Otherwise refer to the vignettes to see how to format the documentation.

---

 SpidermiRAnalyze\_Community\_detection

*Find community detection*


---

**Description**

SpidermiRAnalyze\_Community\_detection try to find dense subgraphs in directed or undirected graphs, by optimizing some criteria.

**Usage**

```
SpidermiRAnalyze_Community_detection(data, type)
```

**Arguments**

data	SpidermiRAnalyze_mirna_network output or SpidermiRAnalyze_mirna_gene_complnet
type	with the parameter type the user can choose the algorithm to calculate the community structure EB edge.betweenness.community FC fastgreedy.community WC walktrap.community SC spinglass.community LE leading.eigenvector.community LP label.propagation.community

**Value**

a list of clusters with their number of genes

**Examples**

```
miRNA_cN <-data.frame(gA=c('hsa-let-7a','hsa-miR-300'),gB=c('FOXM1','KPNA4'),stringsAsFactors=FALSE)
comm<- SpidermiRAnalyze_Community_detection(data=miRNA_cN,type="FC")
```

---

SpidermiRanalyze\_Community\_detection\_bi

*Community detection from biomarkers of interest*

---

### Description

SpidermiRanalyze\_Community\_detection\_bi find the cluster with biomarkers of interest

### Usage

```
SpidermiRanalyze_Community_detection_bi(data, BI)
```

### Arguments

data	SpidermiRanalyze_Community_detection output
BI	a set of biomarkers of interest

### Value

a list with the cluster for each biomarkers of interest

### Examples

```
miRNA_cN <-data.frame(gA=c('hsa-let-7a', 'hsa-miR-300'), gB=c('FOXO1', 'KPNA4'), stringsAsFactors=FALSE)
comm<- SpidermiRanalyze_Community_detection(data=miRNA_cN, type="FC")
biomark_of_interest<-c("hsa-let-7a", "CDK", "FOXO1", "hsa-miR-27a")
mol<-SpidermiRanalyze_Community_detection_bi(data=comm, BI=biomark_of_interest)
```

---

SpidermiRanalyze\_Community\_detection\_net

*Find the network of community detection and direct biomarker*

---

### Description

SpidermiRanalyze\_direct\_net find the direct interactions from a specific community

### Usage

```
SpidermiRanalyze_Community_detection_net(data, comm_det, size)
```

### Arguments

data	SpidermiRanalyze_mirna_network output or SpidermiRanalyze_mirna_gene_complnet
comm_det	SpidermiRanalyze_Community_detection
size	the index of community detection obtained from SpidermiRanalyze_Community_detection

### Value

dataframe with the interatcions



**Examples**

```
miRNA_cN <-data.frame(gA=c('hsa-let-7a','hsa-miR-300'),gB=c('FOXO1','KIP1'),stringsAsFactors=FALSE)
comm<- SpidermiRanalyze_Community_detection(data=miRNA_cN,type="FC")
cd_net<-SpidermiRanalyze_Community_detection_net(data=miRNA_cN,comm_det=comm,size=1)
```

---

SpidermiRanalyze\_degree\_centrality  
*Ranking degree centrality genes*

---

**Description**

SpidermiRanalyze\_degree\_centrality provides degree centrality, defined as the total number of direct neighbors for each gene.

**Usage**

```
SpidermiRanalyze_degree_centrality(data, cut = NULL)
```

**Arguments**

data	SpidermiRanalyze_mirna_network output or SpidermiRanalyze_mirna_gene_complnet
cut	parameter cut is able to cut off other genes

**Value**

dataframe with the ranked number of direct neighbors for each gene of the network

**Examples**

```
miRNA_cN <-data.frame(gA=c('hsa-let-7a','hsa-miR-300'),gB=c('FOXO1','KIP1'),stringsAsFactors=FALSE)
biomark_of_interest<-c("hsa-let-7a","CDK","FOXO1","hsa-miR-27a")
top10_cent<-SpidermiRanalyze_degree_centrality(miRNA_cN)
```

---

SpidermiRanalyze\_DEnetworkTCGA  
*Integration with TCGA data in order to obtain a network of differentially expressed (DE) genes or miRNAs.*

---

**Description**

SpidermiRanalyze\_DEnetworkTCGA integrates the information of differential analysis of TCGA data in the network. The final result will be a network with only DE genes or miRNAs depending whether the user chooses to mRNA or miRNA TCGA data.

**Usage**

```
SpidermiRanalyze_DEnetworkTCGA(data, TCGAmatrix, tumour, normal)
```

**Arguments**

data	network data (e.g. shared protein domains, co-expression,...)
TCGAmatrix	gene or miRNA expression matrix
tumour	barcode TCGA tumour data
normal	barcode TCGA normal data

**Value**

a network miRNA-gene differentially expressed as calculated by TCGAbiolinks package. The user can select the samples and cancer type from TCGA portal.

**Examples**

```
miRNA_cN <-data.frame(gA=c('IGFL3', 'GABRA1'), gB=c('IGFL2', 'KRT13'), stringsAsFactors=FALSE)
tumour<-c("TCGA-E9-A1RD-01A", "TCGA-E9-A1RC-01A")
normal<-c("TCGA-BH-A18P-11A", "TCGA-BH-A18L-11A")
de_int<-SpidermiRanalyze_DEnetworkTCGA(data=miRNA_cN,
                                         TCGAmatrix=Data_CANCER_normUQ_filt,
                                         tumour,
                                         normal
                                         )
```

---

SpidermiRanalyze\_direct\_net

*Searching by biomarkers of interest with direct interaction*

---

**Description**

SpidermiRanalyze\_direct\_net finds other biomarkers that are related to a set of biomarkers of interest (the input of user) with direct interactions.

**Usage**

```
SpidermiRanalyze_direct_net(data, BI)
```

**Arguments**

data	SpidermiRanalyze_mirna_network output or SpidermiRanalyze_mirna_gene_complnet
BI	a set of biomarkers of interest

**Value**

dataframe with direct interaction of biomarkers of interest

**Examples**

```
miRNA_cN <-data.frame(gA=c('hsa-let-7a', 'FOXM1'), gB=c('FOXM1', 'KPNA4'), stringsAsFactors=FALSE)
biomark_of_interest<-c("hsa-let-7a", "CDK", "FOXO1", "hsa-miR-27a")
GIdirect_net<-SpidermiRanalyze_direct_net(data=miRNA_cN, BI=biomark_of_interest)
```

---

SpidermiRanalyze\_direct\_subnetwork

*Searching by biomarkers of interest with direct interaction by ONLY the nodes in BI*

---

### Description

SpidermiRanalyze\_direct\_subnetwork creates a sub network composed by ONLY the nodes in genes of interest and the edges between them

### Usage

```
SpidermiRanalyze_direct_subnetwork(data, BI)
```

### Arguments

data	SpidermiRanalyze_mirna_network output or SpidermiRanalyze_mirna_gene_complnet
BI	a set of biomarkers of interest

### Value

dataframe with direct interaction of biomarkers of interest

### Examples

```
miRNA_cN <-data.frame(gA=c('hsa-let-7a', 'FOXO1'),gB=c('FOXO1', 'KPNA4'),stringsAsFactors=FALSE)
biomark_of_interest<-c("hsa-let-7a", "CDK", "FOXO1", "hsa-miR-27a")
subnet<-SpidermiRanalyze_direct_subnetwork(data=miRNA_cN,BI=biomark_of_interest)
```

---

SpidermiRanalyze\_mirnanet\_pharm

*Integration of pharmacomiR in the network*

---

### Description

SpidermiRanalyze\_mirnanet\_pharm integrates both miRNA targeting of the gene and the gene-drug interaction from PharmacomiR database in the network

### Usage

```
SpidermiRanalyze_mirnanet_pharm(mir_ph, net)
```

### Arguments

mir_ph	SpidermiRdownload_pharmacomir output
net	a network data (e.g. SpidermiRanalyze_mirna_network or SpidermiRanalyze_mirna_gene_complnet output)

**Value**

a dataframe with the integration of network and pharmacomiR data

**Examples**

```
## Not run:
mir_p <-data.frame(gA=c('hsa-let-7a', 'CASP3'),gB=c('CASP3', 'paclitaxel'),stringsAsFactors=FALSE)
net_p <-data.frame(gA=c('hsa-let-7a', 'hsa-miR-300'),gB=c('FOX11', 'KPNA4'),stringsAsFactors=FALSE)
mol<-SpidermiRanalyze_mirnanet_pharm(mir_ph=mir_p,net=net_p)
## End(Not run)
```

---

SpidermiRanalyze\_mirna\_gene\_complnet

*Integration of microRNA target gene networks.*

---

**Description**

SpidermiRanalyze\_mirna\_gene\_complnet creates a data frame with miRNA target gene interaction. The user can filter the search by disease.

**Usage**

```
SpidermiRanalyze_mirna_gene_complnet(data, miR_trg, mirna_t = NULL,
  disease = NULL)
```

**Arguments**

data	SpidermiRprepare_NET output
miR_trg	a parameter to indicate miRNA target database used. The user can use: 1) validated database (val) or 2) predicted database (pred)
mirna_t	a list given by the user with miRNA list of interest
disease	miRNA target gene interaction can be filtered by disease using the parameters obtained from SpidermiRquery_disease

**Value**

dataframe with miRNA target gene interaction data

**Examples**

```
GS_net <- data.frame(gA=c('SMAD', 'MYC'),gB=c('FOX11', 'KRAS'),stringsAsFactors=FALSE)
miRNA_cNT<-SpidermiRanalyze_mirna_gene_complnet(data=GS_net,disease="prostate cancer",miR_trg="val")
```

---

 SpidermiRanalyze\_mirna\_network

*Integration of microRNA target networks.*


---

### Description

SpidermiRanalyze\_mirna\_network creates a data frame with miRNA gene interaction. The user can filter the search by disease.

### Usage

```
SpidermiRanalyze_mirna_network(data, miR_trg, mirna_t = NULL,
  disease = NULL)
```

### Arguments

data	SpidermiRprepare_NET output
miR_trg	a parameter to indicate miRNA target database used. The user can use: 1) validated database (val) or 2) predicted database (pred)
mirna_t	a list given by the user with miRNA list of interest
disease	miRNA gene interaction can be filtered by disease using the parameters obtained from SpidermiRquery_disease

### Value

dataframe with miRNA gene interaction data

### Examples

```
GS_net <- data.frame(gA=c('SMAD', 'MYC'), gB=c('FOXMI1', 'KRAS'), stringsAsFactors=FALSE)
miRNA_NET<-SpidermiRanalyze_mirna_network(data=GS_net, disease="prostate cancer", miR_trg="val")
```

---

 SpidermiRanalyze\_subnetwork\_neigh

*Searching by biomarkers of interest and all the edges among this bunch of nodes*


---

### Description

SpidermiRanalyze\_subnetwork\_neigh create a sub network composed by the nodes in BI and, if some of them are connected to other nodes (even if not in BI), take also them (include all the edges among this bunch of nodes).

### Usage

```
SpidermiRanalyze_subnetwork_neigh(data, BI)
```

**Arguments**

data SpidermiRanalyze\_mirna\_network output or SpidermiRanalyze\_mirna\_gene\_complnet  
BI a set of biomarkers of interest

**Value**

dataframe with interactions

**Examples**

```
miRNA_cN <-data.frame(gA=c('hsa-let-7a', 'hsa-miR-300'), gB=c('FOXO1', 'KPNA4'), stringsAsFactors=FALSE)
biomark_of_interest<-c("hsa-let-7a", "CDK", "FOXO1", "hsa-miR-27a")
GIdirect_net_neigh<-SpidermiRanalyze_subnetwork_neigh(data=miRNA_cN, BI=biomark_of_interest)
```

---

SpidermiRdownload\_miRNAextra\_cir

*Download miRNA validated database*

---

**Description**

SpidermiRdownload\_miRNAprediction will download miRNA validated target

**Usage**

```
SpidermiRdownload_miRNAextra_cir(miRNAextra_cir)
```

**Arguments**

miRNAextra\_cir parameter

**Value**

a dataframe with miRNA target validated interactions

**Examples**

```
list<-SpidermiRdownload_miRNAextra_cir(miRNAextra_cir)
```

---

SpidermiRdownload\_miRNAPrediction  
*Download miRNA predicted database*

---

**Description**

SpidermiRdownload\_miRNAPrediction will download miRNA predicted target

**Usage**

```
SpidermiRdownload_miRNAPrediction(mirna_list)
```

**Arguments**

mirna\_list      miRNA list of interest

**Value**

a dataframe with miRNA target validated interactions

**Examples**

```
mirna<-c('hsa-miR-567', 'hsa-miR-566')  
list<-SpidermiRdownload_miRNAPrediction(mirna_list=mirna)
```

---

SpidermiRdownload\_miRNAvalidate  
*Download miRNA validated database*

---

**Description**

SpidermiRdownload\_miRNAvalidate will download miRNA validated target

**Usage**

```
SpidermiRdownload_miRNAvalidate(validated)
```

**Arguments**

validated      parameter

**Value**

a dataframe with miRNA target validated interactions

**Examples**

```
list<-SpidermiRdownload_miRNAvalidate(validated)
```

---

SpidermiRdownload\_net *Download the network from GeneMania.*

---

**Description**

SpidermiRdownload\_net function will download the data

**Usage**

```
SpidermiRdownload_net(data)
```

**Arguments**

data            The SpidermiRquery\_spec\_networks output

**Value**

Download GeneMania network

**Examples**

```
org<-SpidermiRquery_species(species)
net_shar_prot<-SpidermiRquery_spec_networks(organismID = org[9,],
network = "SHpd")
out_net<-SpidermiRdownload_net(data=net_shar_prot)
```

---

SpidermiRdownload\_pharmacomir

*Download both miRNA target and the gene-drug interaction from  
PharmacomiR database*

---

**Description**

SpidermiRdownload\_pharmacomir will download miRNA Pharmacogenomic data

**Usage**

```
SpidermiRdownload_pharmacomir(pharmacomir)
```

**Arguments**

pharmacomir    variable

**Value**

a dataframe with gene-drug, and miR-gene associations

**Examples**

```
## Not run:
mir_pharmaco<-SpidermiRdownload_pharmacomir(pharmacomir=pharmacomir)
## End(Not run)
```



---

SpidermiRprepare\_NET    *Prepare matrix of gene network from Genamania with Ensembl Gene ID, and gene symbols*

---

### Description

The user in this step obtained a gene network matrix with the integration of gene symbols ID.

### Usage

```
SpidermiRprepare_NET(organismID, data)
```

### Arguments

organismID	is the index of SpidermiRquery_spec_networks output
data	is the output of function SpidermiRdownload_net

### Value

A list of tables.

### Examples

```
org<-SpidermiRquery_species(species)
net_shar_prot<-SpidermiRquery_spec_networks(organismID = org[9,],
network = "SHpd")
out_net<-SpidermiRdownload_net(data=net_shar_prot)
geneSymb_net<-SpidermiRprepare_NET(organismID = org[9,],
                                data = out_net)
```

---

SpidermiRquery\_disease  
*Visualize disease categories*

---

### Description

The user can visualize the disease supported by SpidermiR

### Usage

```
SpidermiRquery_disease(diseaseID)
```

### Arguments

diseaseID	variable name
-----------	---------------

### Value

a list of disease.

### Examples

```
disease<-SpidermiRquery_disease(diseaseID)
```

---

SpidermiRquery\_networks\_type  
*Network categories*

---

**Description**

The user can visualize the network types supported by GeneMania for a specific specie using SpidermiRquery\_networks\_type

**Usage**

```
SpidermiRquery_networks_type(organismID)
```

**Arguments**

organismID      describes index of a specific specie obtained by SpidermiRquery\_species output

**Value**

a list of network categories in a specie indicated.

**Examples**

```
org<-SpidermiRquery_species(species)
net_type<-SpidermiRquery_networks_type(organismID=org[9,])
```

---

SpidermiRquery\_species  
*Searching by network species*

---

**Description**

The user can visualize the species supported by GeneMania, using the function SpidermiRquery\_species .

**Usage**

```
SpidermiRquery_species(species)
```

**Arguments**

species      a variable parameter

**Value**

table of species

**Examples**

```
org<-SpidermiRquery_species(species)
```

---

SpidermiRquery\_spec\_networks

*Searching by network categories*

---

### Description

The user can visualize the database or reference where the information came from

### Usage

```
SpidermiRquery_spec_networks(organismID, network)
```

### Arguments

organismID describes index of a specific specie obtained by SpidermiRquery\_species output  
network The network type the user is interested in. Example:

COexp	Co-expression
PHint	Physical_interactions
COloc	Co-localization
GENint	Genetic_interactions
PATH	Pathway
SHpd	Shared_protein_domains
pred	Predicted

### Value

a list of the database or reference where the information came from.

### Examples

```
org<-SpidermiRquery_species(species)
net_shar_prot<-SpidermiRquery_spec_networks(organismID = org[9,],
network = "SHpd")
```

---

SpidermiRvisualize\_3Dbarplot

*plots the 3D barplot*

---

### Description

It shows a barplot of 5 networks given by the user with a summary representation of number of nodes, edges, and miRNAs (log values)

### Usage

```
SpidermiRvisualize_3Dbarplot(Edges_1net, Edges_2net, Edges_3net, Edges_4net,
Edges_5net, NODES_1net, NODES_2net, NODES_3net, NODES_4net, NODES_5net,
nmiRNAs_1net, nmiRNAs_2net, nmiRNAs_3net, nmiRNAs_4net, nmiRNAs_5net)
```

**Arguments**

Edges_1net	int number of edges in the 1 net
Edges_2net	int number of edges in the 2 net
Edges_3net	int number of edges in the 3 net
Edges_4net	int number of edges in the 4 net
Edges_5net	int number of edges in the 5 net
NODES_1net	int number of nodes in the 1 net
NODES_2net	int number of nodes in the 2 net
NODES_3net	int number of nodes in the 3 net
NODES_4net	int number of nodes in the 4 net
NODES_5net	int number of nodes in the 5 net
nmiRNAs_1net	int number of miRNAs in the 1 net
nmiRNAs_2net	int number of miRNAs in the 2 net
nmiRNAs_3net	int number of miRNAs in the 3 net
nmiRNAs_4net	int number of miRNAs in the 4 net
nmiRNAs_5net	int number of miRNAs in the 5 net

**Value**

barplot

**Examples**

```
SpidermiRvisualize_3Dbarplot(Edges_1net=1041003,Edges_2net=100016,Edges_3net=3008,
Edges_4net=1493,Edges_5net=1598,NODES_1net=16502,NODES_2net=13338,NODES_3net=1429,NODES_4net=675,
NODES_5net=712,nmiRNAs_1net=0,nmiRNAs_2net=74,nmiRNAs_3net=0,nmiRNAs_4net=0,nmiRNAs_5net=37)
```

---

```
SpidermiRvisualize_adj_matrix
```

*plots the adjacency matrix of the network*

---

**Description**

It shows a plot OF the adjacency matrix of the network

**Usage**

```
SpidermiRvisualize_adj_matrix(data)
```

**Arguments**

data	The input data is a network
------	-----------------------------

**Value**

plot

**Examples**

```
cd<-data.frame(gA=c('hsa-let-7a','hsa-miR-141'),gB=c('FOXm1','CDK'),stringsAsFactors=FALSE)
SpidermiRvisualize_adj_matrix(data=cd)
```

---

SpidermiRvisualize\_BI *Visualize results obtained by SpidermiR analysis starting from a set of biomarker of interest*

---

**Description**

Visualize miRNA-target interaction and miRNA-target-gene starting from a set of biomarker of interest

**Usage**

```
SpidermiRvisualize_BI(data, BI)
```

**Arguments**

data            The input data is a dataframe containing network data.  
BI              a set of biomarkers of interest

**Value**

3D graphic

**Examples**

```
miRNA_cNET <-data.frame(gA=c('hsa-let-7a','hsa-miR-141'),gB=c('FOXO1','CDK'),stringsAsFactors=FALSE)
biomark_of_interest<-c("hsa-let-7a","CDK","FOXO1","hsa-miR-27a")
SpidermiRvisualize_BI(data=miRNA_cNET,BI=biomark_of_interest)
```

---

SpidermiRvisualize\_degree\_dist  
*plots the degree distribution of the network*

---

**Description**

It shows a plot of the degree distribution of the network

**Usage**

```
SpidermiRvisualize_degree_dist(data)
```

**Arguments**

data            The input data is a network

**Value**

plot

**Examples**

```
cd<-data.frame(gA=c('hsa-let-7a','hsa-miR-141'),gB=c('FOXO1','CDK'),stringsAsFactors=FALSE)
SpidermiRvisualize_degree_dist(data=cd)
```

SpidermiRvisualize\_direction

*Visualize results obtained by SpidermiR analysis with the direction of the interaction (pharmaco-gene and miRNA-gene)*

---

### **Description**

Visualize the network

### **Usage**

```
SpidermiRvisualize_direction(data)
```

### **Arguments**

data            The input data is a dataframe containing network data.

### **Value**

3D graphic

### **Examples**

```
miRNA_cNET <-data.frame(gA=c('hsa-let-7a','hsa-miR-141'),gB=c('FOXM1','CDK'),stringsAsFactors=FALSE)
SpidermiRvisualize_direction(data=miRNA_cNET)
```

---

SpidermiRvisualize\_mirnanet

*Visualize results obtained by SpidermiR analysis*

---

### **Description**

Visualize the network

### **Usage**

```
SpidermiRvisualize_mirnanet(data)
```

### **Arguments**

data            The input data is a dataframe containing network data.

### **Value**

3D graphic

### **Examples**

```
miRNA_cNET <-data.frame(gA=c('hsa-let-7a','hsa-miR-141'),gB=c('FOXM1','CDK'),stringsAsFactors=FALSE)
SpidermiRvisualize_mirnanet(data=miRNA_cNET)
```

---

`SpidermiRvisualize_plot_target`*Visualize results obtained by SpidermiRanalyze\_mirna\_network*

---

**Description**

It shows a plot with miRNAs and the number of their targets in the network

**Usage**

```
SpidermiRvisualize_plot_target(data)
```

**Arguments**

<code>data</code>	The input data is a dataframe containing miRNA network data (e.g. output of SpidermiRanalyze_mirna_network).
-------------------	--

**Value**

plot

**Examples**

```
cd<-data.frame(gA=c('hsa-let-7a', 'hsa-miR-141'), gB=c('FOXM1', 'CDK'), stringsAsFactors=FALSE)
SpidermiRvisualize_plot_target(data=cd)
```

# Index

Case\_Study1\_loading\_1\_network, 3  
Case\_Study1\_loading\_2\_network, 3  
Case\_Study1\_loading\_3\_network, 4  
Case\_Study1\_loading\_4\_network, 4  
Case\_Study2\_loading\_1\_network, 5  
Case\_Study2\_loading\_2\_network, 6  
Case\_Study2\_loading\_3\_network, 6

SpidermiR, 7  
SpidermiR-package (SpidermiR), 7  
SpidermiRanalyze\_Community\_detection,  
7  
SpidermiRanalyze\_Community\_detection\_bi,  
8  
SpidermiRanalyze\_Community\_detection\_net,  
8  
SpidermiRanalyze\_degree\_centrality, 9  
SpidermiRanalyze\_DEnetworkTCGA, 9  
SpidermiRanalyze\_direct\_net, 10  
SpidermiRanalyze\_direct\_subnetwork, 11  
SpidermiRanalyze\_mirna\_gene\_complnet,  
12  
SpidermiRanalyze\_mirna\_network, 13  
SpidermiRanalyze\_mirnanet\_pharm, 11  
SpidermiRanalyze\_subnetwork\_neigh, 13  
SpidermiRdownload\_miRNAextra\_cir, 14  
SpidermiRdownload\_miRNAprediction, 15  
SpidermiRdownload\_miRNAvalidate, 15  
SpidermiRdownload\_net, 16  
SpidermiRdownload\_pharmacomir, 16  
SpidermiRprepare\_NET, 17  
SpidermiRquery\_disease, 17  
SpidermiRquery\_networks\_type, 18  
SpidermiRquery\_spec\_networks, 19  
SpidermiRquery\_species, 18  
SpidermiRvisualize\_3Dbarplot, 19  
SpidermiRvisualize\_adj\_matrix, 20  
SpidermiRvisualize\_BI, 21  
SpidermiRvisualize\_degree\_dist, 21  
SpidermiRvisualize\_direction, 22  
SpidermiRvisualize\_mirnanet, 22  
SpidermiRvisualize\_plot\_target, 23