

# Package ‘SpidermiR’

March 30, 2021

**Type** Package

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

**Version** 1.20.0

**Date** 10-05-2020

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

**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 (mirWalk, miRTarBase, miRandola, Miranda, PicTar and TargetScan). Furthermore, we also present a statistical test to identify pharmaco-mir relationships using the gene-drug interactions derived by DGIdb and MATADOR database.

**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** 7.0.2

**NeedsCompilation** no

**git\_url** <https://git.bioconductor.org/packages/SpidermiR>

**git\_branch** RELEASE\_3\_12

**git\_last\_commit** f71b8e0

**git\_last\_commit\_date** 2020-10-27

**Date/Publication** 2021-03-29

## R topics documented:

SpidermiRanalyze_Community_detection . . . . .	3
SpidermiRanalyze_Community_detection_bi . . . . .	3
SpidermiRanalyze_Community_detection_net . . . . .	4
SpidermiRanalyze_degree centrality . . . . .	4
SpidermiRanalyze_direct_net . . . . .	5
SpidermiRanalyze_direct_subnetwork . . . . .	6
SpidermiRanalyze_subnetwork_neigh . . . . .	6
SpidermiRdownload_drug_gene . . . . .	7
SpidermiRdownload_miRNAextra_cir . . . . .	7
SpidermiRdownload_miRNAprediction . . . . .	8
SpidermiRdownload_miRNAprediction_mmu . . . . .	8
SpidermiRdownload_miRNAvalidate . . . . .	9
SpidermiRdownload_net . . . . .	9
SpidermiRdownload_pharmacomir . . . . .	10
SpidermiRprepare_NET . . . . .	10
SpidermiRquery_disease . . . . .	11
SpidermiRquery_networks_type . . . . .	11
SpidermiRquery_species . . . . .	12
SpidermiRquery_spec_networks . . . . .	12
SpidermiRvisualize_3Dbarplot . . . . .	13
SpidermiRvisualize_adj_matrix . . . . .	14
SpidermiRvisualize_BI . . . . .	15
SpidermiRvisualize_degree_dist . . . . .	15
SpidermiRvisualize_direction . . . . .	16
SpidermiRvisualize_mirnanet . . . . .	16
SpidermiRvisualize_plot_target . . . . .	17
<b>Index</b>	<b>18</b>

---

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', 'KPNA4'), 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', 'KPNA4'), stringsAsFactors=FALSE)
biomark_of_interest<-c("hsa-let-7a", "CDK", "FOXO1", "hsa-miR-27a")
top10_cent<-SpidermiRanalyze_degree centrality(miRNA_cN)
```

---

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', 'FOXO1'), gB=c('FOXO1', '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\_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\_drug\_gene

*Download drug-gene interactions in DGIdb*

---

**Description**

SpidermiRdownload\_drug\_gene will download drug gene interactions

**Usage**

```
SpidermiRdownload_drug_gene(drug_gene)
```

**Arguments**

drug\_gene      parameter

**Value**

a dataframe with miRNA target validated interactions

**Examples**

```
drug_genetarget<-SpidermiRdownload_drug_gene(drug_gene)
```

---

SpidermiRdownload\_miRNAextra\_cir

*Download miRNA validated database*

---

**Description**

SpidermiRdownload\_miRNAextra\_cir 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 human 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')
list<-SpidermiRdownload_miRNAprediction(mirna_list=mirna)
```

---

SpidermiRdownload\_miRNAprediction\_mmu

*Download mmu miRNA predicted database*

---

**Description**

SpidermiRdownload\_miRNAprediction\_mmu will download miRNA predicted target

**Usage**

```
SpidermiRdownload_miRNAprediction_mmu(mirna_list)
```

**Arguments**

mirna\_list      miRNA list of interest

**Value**

a dataframe with miRNA target validated interactions



**Examples**

```
mirna<-c('mmu-miR-708-3p')
list<-SpidermiRdownload_miRNAPrediction_mmu(mirna_list=mirna)
```

---

SpidermiRdownload\_miRNAvalidate

*Download miRNA validated database*

---

**Description**

SpidermiRdownload\_miRNAPrediction 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 drug-miRNA interactions with fisher test*

---

### Description

SpidermiRdownload\_pharmacomir will calculate miRNA- drug interactions

### Usage

```
SpidermiRdownload_pharmacomir(list, drug)
```

### Arguments

list	miRNA gene target as obtained from e.g.; SpidermiRdownload_miRNAvalidate
drug	parameter drug of interest

### Value

a dataframe with miRNA target validated interactions

### Examples

```
list1<-SpidermiRdownload_miRNAvalidate(validated)
drug="TAMOXIFEN"
drug_genetarget<-SpidermiRdownload_pharmacomir(list1[1:100,],drug="TAMOXIFEN")
```

---

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[,])
```

---

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('FOXM1','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('FOXM1','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 miRNA 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

SpidermiRanalyze\_Community\_detection, [3](#)  
SpidermiRanalyze\_Community\_detection\_bi, [3](#)  
SpidermiRanalyze\_Community\_detection\_net, [4](#)  
SpidermiRanalyze\_degree centrality, [4](#)  
SpidermiRanalyze\_direct\_net, [5](#)  
SpidermiRanalyze\_direct\_subnetwork, [6](#)  
SpidermiRanalyze\_subnetwork\_neigh, [6](#)  
SpidermiRdownload\_drug\_gene, [7](#)  
SpidermiRdownload\_miRNAextra\_cir, [7](#)  
SpidermiRdownload\_miRNAprediction, [8](#)  
SpidermiRdownload\_miRNAprediction\_mmu, [8](#)  
SpidermiRdownload\_miRNAvalidate, [9](#)  
SpidermiRdownload\_net, [9](#)  
SpidermiRdownload\_pharmacomir, [10](#)  
SpidermiRprepare\_NET, [10](#)  
SpidermiRquery\_disease, [11](#)  
SpidermiRquery\_networks\_type, [11](#)  
SpidermiRquery\_spec\_networks, [12](#)  
SpidermiRquery\_species, [12](#)  
SpidermiRvisualize\_3Dbarplot, [13](#)  
SpidermiRvisualize\_adj\_matrix, [14](#)  
SpidermiRvisualize\_BI, [15](#)  
SpidermiRvisualize\_degree\_dist, [15](#)  
SpidermiRvisualize\_direction, [16](#)  
SpidermiRvisualize\_mirnanet, [16](#)  
SpidermiRvisualize\_plot\_target, [17](#)