

ggNetView manual documentation

Yue Liu

2026-01-08

Contents

1 ggNetView manual documentation	7
1.1 Introduction	7
1.2 Installation	8
1.3 Citation	8
1.4 Source Code	8
1.5 Contact	8
2 Build graph object	9
2.1 Build graph from matrix	9
2.2 Build graph from data frame	12
2.3 Build graph from adjacency matrix	14
2.4 Build graph from double matrix	15
2.5 Build graph from adjacency matrix with module Information . .	15
2.6 Build graph from igraph	15
3 Random Matrix Theory (RMT)-based random network	17
3.1 RMT	17
4 Get network information	19
4.1 Full-network information	19
4.2 Sub-network (modularity) information	19
4.3 Sub-network (sample) information	19

5 Extract subgraph	21
5.1 Extract subgraph by module	21
5.2 Extract subgraph by sample	21
6 Network layout	23
6.1 Gephi layout	24
6.2 Fruchterman–Reingold force-directed layout	24
6.3 Diamond layout	24
6.4 Kk layout	24
6.5 Nicley layout	24
6.6 Multrings layout	24
6.7 Petal layout	24
6.8 Circle layout	24
6.9 Circle outline layout	24
6.10 Diamond outline layout	24
6.11 Grid layout	24
6.12 Heart_centered layout	24
6.13 Lgl layout	24
6.14 Randomly layout	24
6.15 Rectangle layout	24
6.16 Rightiso layout	24
6.17 Square layout	24
6.18 Square outline layout	24
6.19 Star layout	24
6.20 Star_concentric layout	24
6.21 Stress layout	24
7 Network topology information	25
7.1 Get network topology information	25
7.2 Get network topology information with matrix	25
7.3 Get network topology information by parallel	25
7.4 Get network topology information with matrix by parallel	25

CONTENTS	5
8 Network comparison	27
8.1 Subgraph comparisopn	27
8.2 Comparison of multi-sample networks	27
9 Network & Environment	29
9.1 Network Environment	29
10 Multi-omics network analysis	31
10.1 Multi-omics	31

Chapter 1

ggNetView manual documentation

1.1 Introduction

Network analysis has been widely applied in the **life sciences, microbiology, ecology, and agronomy** to quantify interactions among **genes, proteins, metabolites, microorganisms, and environmental factors**. These interactions collectively shape the functioning of biological and ecological systems. Despite its broad adoption, existing tools often face limitations in flexibility, customization, reproducibility, and the generation of publication-ready figures.

To address these challenges, **ggNetView** was developed as an **R package** that integrates **ggplot2**, **ggraph**, and **tidygraph** within the **Grammar of Graphics framework**, enabling fully reproducible and highly customizable network visualizations. The package provides deterministic layout algorithms, comprehensive topological analyses, modular coloring schemes, hierarchical annotations, and consistent theming, ensuring standardized and interpretable graphical output.

Although initially designed for applications in **soil science and microbial ecology**, **ggNetView** is broadly applicable to network analyses in **molecular biology**, including **WGCNA** and **protein–protein interaction (PPI) networks**. By lowering technical barriers in network construction and visualization, **ggNetView** enables researchers across disciplines to efficiently produce reproducible, publication-quality network figures.

1.2 Installation

First, install the required dependencies

```
# install.packages("BiocManager")
BiocManager::install("WGCNA")

# install.packages("remotes")
remotes::install_github("alserglab/mascarade")
remotes::install_github("zdk123/SpiecEasi")
```

and then install ggNetView.

```
# install.packages("devtools")
devtools::install_github("Jiawang1209/ggNetView")

# install.packages("pak")
pak::pak("Jiawang1209/ggNetView")
```

1.3 Citation

If you use ggNetView in your research, please cite:

Yue Liu, Chao Wang (2025). ggNetView: An R package for complex biological and ecological network visualization. <https://github.com/Jiawang1209/ggNetView>

1.4 Source Code

The source code for ggNetView is available in the ggNetView repository.

<https://github.com/Jiawang1209/ggNetView>

1.5 Contact

- Email: Jiawang1209@163.com

Chapter 2

Build graph object

Basic workflow of `ggNetView`

1. Build a **graph object**
 2. Understand and manipulate the **graph object**
 3. Visualize the network using **layout algorithms**
 4. Retrieve **topology** of network
-

Load R Package

```
library(tidyverse)
library(ggNetView)
```

2.1 Build graph from matrix

Example data

```
# Access built-in example datasets in ggNetView

# Raw ASV or OTU table
data("otu_tab")
dim(otu_tab)
```

```

## [1] 2859    18

otu_tab[1:5, 1:5]

##      K01   K02   K03   K04   K05
## ASV_1 1113 1968  816 1372 1062
## ASV_2 1922 1227 2355 2218 2885
## ASV_3  568  460  899  902 1226
## ASV_4 1433  400  535  759 1287
## ASV_6  882  673  819  888 1475

# Rarefied ASV or OTU table
data("otu_rare")
dim(otu_rare)

## [1] 2859    18

otu_rare[1:5, 1:5]

##      K01   K02   K03   K04   K05
## ASV_1  992 1636  604 1084  806
## ASV_2 1725 1018 1814 1743 2196
## ASV_3  520  389  687  701  932
## ASV_4 1280  328  425  580 1004
## ASV_6  794  557  633  706 1142

# Relative abundance table of rarefied ASVs or OTUs
data("otu_rare_relative")
dim(otu_rare_relative)

## [1] 2859    18

otu_rare_relative[1:5, 1:5]

##          K01          K02          K03          K04          K05
## ASV_1 0.03306667 0.05453333 0.02013333 0.03613333 0.02686667
## ASV_2 0.05750000 0.03393333 0.06046667 0.05810000 0.07320000
## ASV_3 0.01733333 0.01296667 0.02290000 0.02336667 0.03106667
## ASV_4 0.04266667 0.01093333 0.01416667 0.01933333 0.03346667
## ASV_6 0.02646667 0.01856667 0.02110000 0.02353333 0.03806667

```

```
# Taxonomic annotation table of ASVs or OTUs
data("tax_tab")
tax_tab[1:5, 1:5]

## # A tibble: 5 x 5
##   OTUID Kingdom Phylum      Class      Order
##   <chr>  <chr>  <chr>      <chr>      <chr>
## 1 ASV_2  Archaea Thaumarchaeota Unassigned Nitrosphaerales
## 2 ASV_3  Bacteria Verrucomicrobia Spartobacteria Unassigned
## 3 ASV_31 Bacteria Actinobacteria Actinobacteria Actinomycetales
## 4 ASV_27 Archaea Thaumarchaeota Unassigned Nitrosphaerales
## 5 ASV_9  Bacteria Unassigned     Unassigned    Unassigned
```

Build graph object

```
graph_obj <- build_graph_from_mat(
  mat = otu_rare_relative,
  transfrom.method = "none", # based your input data
  r.threshold = 0.7,
  p.threshold = 0.05,
  method = "WGCNA",
  cor.method = "pearson",
  proc = "Bonferroni",
  module.method = "Fast_greedy",
  node_annotation = tax_tab,
  top_modules = 15,
  seed = 1115
)

graph_obj

## # A tbl_graph: 213 nodes and 844 edges
## #
## # An undirected simple graph with 29 components
## #
## # Node Data: 213 x 14 (active)
##   name    modularity modularity2 modularity3 Modularity Degree Strength Kingdom
##   <chr>    <fct>      <ord>      <chr>      <ord>      <dbl>    <dbl> <chr>
## 1 ASV_649 5          5          5          5          27       26.5 Bacter-
## 2 ASV_705 5          5          5          5          27       26.5 Bacter-
## 3 ASV_12~ 5          5          5          5          27       26.5 Bacter-
## 4 ASV_13~ 5          5          5          5          27       26.5 Bacter-
## 5 ASV_14~ 5          5          5          5          27       26.5 Bacter-
## 6 ASV_14~ 5          5          5          5          27       26.5 Bacter-
```

```

## 7 ASV_24~ 5      5      5      5      27 26.5 Bacter-
## 8 ASV_25~ 5      5      5      5      27 26.4 Bacter-
## 9 ASV_28~ 5      5      5      5      27 26.5 Bacter-
## 10 ASV_28~ 5     5      5      5      27 26.5 Bacter-
## # i 203 more rows
## # i 6 more variables: Phylum <chr>, Class <chr>, Order <chr>, Family <chr>,
## #   Genus <chr>, Species <chr>
## #
## # Edge Data: 844 x 5
##   from      to weight correlation corr_direction
##   <int> <int> <dbl>      <dbl> <chr>
## 1    194    195  0.959      0.959 Positive
## 2    185    208  0.954      0.954 Positive
## 3    185    213  0.957      0.957 Positive
## # i 841 more rows

```

2.2 Build graph from data frame

Example data

```
# Access built-in example datasets in ggNetView
data("ppi_example")
df = ppi_example$ppi
head(df)
```

```

##   from      to weight
## 1 A1 D40 9.306533
## 2 A2 D39 11.783920
## 3 A3 D38 23.005025
## 4 A4 D37 7.412060
## 5 A5 D36 18.778894
## 6 A6 D35 16.592965
```

```
node_annotation = ppi_example$annotation
head(node_annotation)
```

```

##   node group
## 1 A1    A
## 2 A2    A
## 3 A3    A
## 4 A4    A
```

```
## 5   A5     A
## 6   A6     A
```

Build graph object

```
graph_obj_from_df <- build_graph_from_df(
  df = df,
  node_annotation = node_annotation,
  directed = F,
  module.method = "Fast_greedy",
  top_modules = 15,
  seed = 1115
)

graph_obj_from_df

## # A tbl_graph: 100 nodes and 50 edges
## #
## # An unrooted forest with 50 trees
## #
## # Node Data: 100 x 9 (active)
##    name group modularity modularity2 modularity3 Modularity Degreee Segree
##    <chr> <chr> <fct>      <fct>      <chr>      <fct>      <dbl> <dbl>
## 1 C13   C     1          1          1          1          1          1
## 2 C28   C     1          1          1          1          1          1
## 3 C2    C    10         10         10         10          1          1
## 4 D9    D    10         10         10         10          1          1
## 5 A3    A    11         11         11         11          1          1
## 6 D38   D    11         11         11         11          1          1
## 7 B12   B    12         12         12         12          1          1
## 8 D19   D    12         12         12         12          1          1
## 9 A1    A    13         13         13         13          1          1
## 10 D40  D   13         13         13         13          1          1
## # i 90 more rows
## # i 1 more variable: Strength <dbl>
## #
## # Edge Data: 50 x 4
##    from      to weight correlation
##    <int> <int> <dbl>      <dbl>
## 1     9     10   45.2      45.2
## 2    15     16   50.6      50.6
## 3     5      6   37.8      37.8
## # i 47 more rows
```

```

graph_obj_from_df2 <- build_graph_from_df(
  df = df,
  node_annotation = NULL,
  directed = F,
  module.method = "Fast_greedy",
  top_modules = 15,
  seed = 1115
)

graph_obj_from_df2

## # A tbl_graph: 100 nodes and 50 edges
## #
## # An unrooted forest with 50 trees
## #
## # Node Data: 100 x 8 (active)
##   name modularity modularity2 modularity3 Modularity Degree Segree Strength
##   <chr> <fct>      <fct>      <chr>      <fct>      <dbl> <dbl> <dbl>
## 1 C13  1          1          1          1          1     1     1    26.7
## 2 C28  1          1          1          1          1     1     1    26.7
## 3 C2   10         10         10         10         10    1     1    37.4
## 4 D9   10         10         10         10         10    1     1    37.4
## 5 A3   11         11         11         11         11    1     1    37.8
## 6 D38  11         11         11         11         11    1     1    37.8
## 7 B12  12         12         12         12         12    1     1    41.3
## 8 D19  12         12         12         12         12    1     1    41.3
## 9 A1   13         13         13         13         13    1     1    45.2
## 10 D40 13         13         13         13         13   1     1    45.2
## # i 90 more rows
## #
## # Edge Data: 50 x 4
##   from   to weight correlation
##   <int> <int> <dbl>      <dbl>
## 1 9     10    45.2      45.2
## 2 15   16    50.6      50.6
## 3 5     6     37.8      37.8
## # i 47 more rows

```

2.3 Build graph from adjacency matrix

Example data

Build graph object

2.4 Build graph from double matrix

Example data

Build graph object

2.5 Build graph from adjacency matrix with module Information

Example data

Build graph object

2.6 Build graph from igraph

Example data

Build graph object

Chapter 3

Random Matrix Theory (RMT)–based random network

3.1 RMT

Chapter 4

Get network information

4.1 Full-network information

4.2 Sub-network (modularity) information

4.3 Sub-network (sample) information

Chapter 5

Extract subgraph

5.1 Extract subgraph by module

5.2 Extract subgraph by sample

Chapter 6

Network layout

6.1 Gephi layout

6.2 Fruchterman–Reingold force-directed layout

6.3 Diamond layout

6.4 Kk layout

6.5 Nicley layout

6.6 Multrings layout

6.7 Petal layout

6.8 Circle layout

6.9 Circle outline layout

6.10 Diamond outline layout

6.11 Grid layout

6.12 Heart_centered layout

6.13 Lgl layout

6.14 Randomly layout

6.15 Rectangle layout

Chapter 7

Network topology information

- 7.1 Get network topology information**
- 7.2 Get network topology information with matrix**
- 7.3 Get network topology information by parallel**
- 7.4 Get network topology information with matrix by parallel**

Chapter 8

Network comparison

8.1 Subgraph comparisopn

8.2 Comparison of multi-sample networks

Chapter 9

Network & Environment

9.1 Network Environment

Chapter 10

Multi-omics network analysis

10.1 Multi-omics