

# Package ‘meconetcomp’

August 8, 2023

**Type** Package

**Title** Compare Microbial Networks of 'trans\_network' Class of 'microeco' Package

**Version** 0.3.0

**Author** Chi Liu [aut, cre],  
Minjie Yao [ctb],  
Xiangzhen Li [ctb]

**Maintainer** Chi Liu <liuchi0426@126.com>

**Description** Compare microbial co-occurrence networks created from 'trans\_network' class of 'microeco' package <<https://github.com/ChiLiubio/microeco>>. This package is the extension of 'trans\_network' class of 'microeco' package and especially useful when different networks are constructed and analyzed simultaneously.

**URL** <https://github.com/ChiLiubio/meconetcomp>

**Depends** R (>= 3.5.0)

**Imports** R6, microeco (>= 0.20.0), magrittr, dplyr, igraph, reshape2, ggpubr

**Suggests** rgexf, ape, file2meco, agricolae

**License** GPL-3

**LazyData** true

**Encoding** UTF-8

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-08-08 08:10:05 UTC

**RoxygenNote** 7.2.3

## R topics documented:

cal_module . . . . .	2
cal_network_attr . . . . .	3

edge_comp . . . . .	3
edge_node_distance . . . . .	4
edge_tax_comp . . . . .	7
get_edge_table . . . . .	8
get_node_table . . . . .	8
meconetcomp . . . . .	9
node_comp . . . . .	10
soil_amp . . . . .	10
soil_amp_network . . . . .	11
soil_measure_diversity . . . . .	11
stool_met . . . . .	11
subnet_property . . . . .	12
subset_network . . . . .	12

## Index 14

---

cal_module	<i>Assign modules to each network</i>
------------	---------------------------------------

---

## Description

Calculating modularity of networks and assign the modules to nodes for each network.

## Usage

```
cal_module(
  network_list,
  undirected_method = "cluster_fast_greedy",
  directed_method = "cluster_optimal",
  ...
)
```

## Arguments

network_list	a list with multiple networks; all the networks should be <code>trans_network</code> object created from <code>trans_network</code> class of <code>microeco</code> package.
undirected_method	default "cluster_fast_greedy"; the modularity algorithm for undirected network; see <code>cal_module</code> function of <code>trans_network</code> class for more algorithms.
directed_method	default 'cluster_optimal'; the modularity algorithm for directed network.
...	other parameters (except for method) passed to <code>cal_module</code> function of <code>trans_network</code> class.

## Value

list, with module attribute in nodes of each network

**Examples**

```
data(soil_amp_network)
soil_amp_network <- cal_module(soil_amp_network)
```

---

cal_network_attr	<i>Calculate network topological property for each network</i>
------------------	--

---

**Description**

Calculate the topological properties of all the networks and merge the results into one table.

**Usage**

```
cal_network_attr(network_list)
```

**Arguments**

network\_list a list with multiple networks; all the networks should be trans\_network object created from [trans\\_network](#) class of microeco package.

**Value**

data.frame

**Examples**

```
data(soil_amp_network)
test <- cal_network_attr(soil_amp_network)
```

---

edge_comp	<i>Generate a microtable object with paired nodes distributions of edges across networks</i>
-----------	--

---

**Description**

Generate a microtable object with paired nodes distributions of edges across networks. Useful for the edge comparisons across different networks. The return otu\_table in microtable object has the binary numbers in which 1 represents the presence of the edge in the corresponding network.

**Usage**

```
edge_comp(network_list)
```

**Arguments**

`network_list` a list with multiple networks; all the networks should be `trans_network` object created from `trans_network` class of `microeco` package.

**Value**

microtable object

**Examples**

```
data(soil_amp_network)
test <- edge_comp(soil_amp_network)
# test is a microtable object
```

---

<code>edge_node_distance</code>	<i>Perform the distance distribution of paired nodes in edges across networks.</i>
---------------------------------	--

---

**Description**

This class is a wrapper for a series of analysis on the distance values of paired nodes in edges across networks, including distance matrix conversion, the differential test and the visualization.

**Methods****Public methods:**

- `edge_node_distance$new()`
- `edge_node_distance$cal_diff()`
- `edge_node_distance$plot()`
- `edge_node_distance$clone()`

**Method new():**

*Usage:*

```
edge_node_distance$new(
  network_list,
  dis_matrix = NULL,
  label = "+",
  with_module = FALSE,
  module_thres = 2
)
```

*Arguments:*

`network_list` a list with multiple networks; all the networks should be `trans_network` object created from `trans_network` class of `microeco` package.

`dis_matrix` default NULL; the distance matrix of nodes, used for the value extraction; must be a symmetrical matrix with both `colnames` and `rownames` (i.e. feature names).

label default "+"; "+" or "-" or c("+", "-"); the edge label used for the selection of edges.  
 with\_module default FALSE; whether show the module classification of nodes in the result.  
 module\_thres default 2; the threshold of the nodes number of modules remained when with\_module = TRUE.

*Returns:* data\_table, stored in the object

*Examples:*

```
\donttest{
data(soil_amp_network)
data(soil_amp)
# filter useless features to speed up the calculation
node_names <- unique(unlist(lapply(soil_amp_network, function(x){colnames(x$data_abund)})))
filter_soil_amp <- microeco::clone(soil_amp)
filter_soil_amp$otu_table <- filter_soil_amp$otu_table[node_names, ]
filter_soil_amp$tidy_dataset()
# obtain phylogenetic distance matrix
phylogenetic_distance <- as.matrix(cophenetic(filter_soil_amp$phylo_tree))
# choose the positive labels
t1 <- edge_node_distance$new(network_list = soil_amp_network,
  dis_matrix = phylogenetic_distance, label = "+")
}
```

**Method** cal\_diff(): Differential test across networks.

*Usage:*

```
edge_node_distance$cal_diff(
  method = c("anova", "KW", "KW_dunn", "wilcox", "t.test")[1],
  ...
)
```

*Arguments:*

method default "anova"; see the following available options:

**'anova'** Duncan's multiple range test for anova

**'KW'** KW: Kruskal-Wallis Rank Sum Test for all groups ( $\geq 2$ )

**'KW\_dunn'** Dunn's Kruskal-Wallis Multiple Comparisons, see dunnTest function in FSA package

**'wilcox'** Wilcoxon Rank Sum and Signed Rank Tests for all paired groups

**'t.test'** Student's t-Test for all paired groups

... parameters passed to cal\_diff function of trans\_alpha class of microeco package.

*Returns:* res\_diff in object. See the Return of cal\_diff function in trans\_alpha class of microeco package.

*Examples:*

```
\donttest{
t1$cal_diff(method = "wilcox")
}
```

**Method** plot(): Plot the distance.

*Usage:*

```
edge_node_distance$plot(...)
```

*Arguments:*

... parameters pass to plot\_alpha function of trans\_alpha class of microeco package.

*Returns:* ggplot.

*Examples:*

```
\donttest{
t1$plot(boxplot_add = "none", add_sig = TRUE)
}
```

**Method clone():** The objects of this class are cloneable with this method.

*Usage:*

```
edge_node_distance$clone(deep = FALSE)
```

*Arguments:*

deep Whether to make a deep clone.

## Examples

```
## -----
## Method `edge_node_distance$new`
## -----

data(soil_amp_network)
data(soil_amp)
# filter useless features to speed up the calculation
node_names <- unique(unlist(lapply(soil_amp_network, function(x){colnames(x$data_abund)})))
filter_soil_amp <- microeco::clone(soil_amp)
filter_soil_amp$otu_table <- filter_soil_amp$otu_table[node_names, ]
filter_soil_amp$tidy_dataset()
# obtain phylogenetic distance matrix
phylogenetic_distance <- as.matrix(cophenetic(filter_soil_amp$phylo_tree))
# choose the positive labels
t1 <- edge_node_distance$new(network_list = soil_amp_network,
  dis_matrix = phylogenetic_distance, label = "+")

## -----
## Method `edge_node_distance$cal_diff`
## -----

t1$cal_diff(method = "wilcox")

## -----
## Method `edge_node_distance$plot`
## -----
```

```
t1$plot(boxplot_add = "none", add_sig = TRUE)
```

---

edge\_tax\_comp

*Taxonomic sum of linked nodes in edges across networks*

---

### Description

Taxonomic sum of linked nodes in edges across networks.

### Usage

```
edge_tax_comp(network_list, taxrank = "Phylum", label = "+", rel = TRUE)
```

### Arguments

network_list	a list with multiple networks; all the networks should be trans_network object created from <a href="#">trans_network</a> class of microeco package.
taxrank	default "Phylum"; Which taxonomic level is used for the sum of nodes in edges.
label	default "+"; "+" or "-" or c("+", "-"); the edge label used for the selection of edges for the sum.
rel	default TRUE; TRUE represents using ratio, the denominator is the number of selected edges; FALSE represents the absolute number of the sum of edges.

### Value

data.frame

### Examples

```
data(soil_amp_network)
test <- edge_tax_comp(soil_amp_network)
# test is a microtable object
```

---

get_edge_table	<i>Get edge property table for each network</i>
----------------	---

---

**Description**

Get edge property table for each network in the list with multiple networks.

**Usage**

```
get_edge_table(network_list)
```

**Arguments**

`network_list` a list with multiple networks; all the networks should be `trans_network` object created from `trans_network` class of `microeco` package.

**Value**

list, with `res_edge_table` in each network

**Examples**

```
data(soil_amp_network)
soil_amp_network <- get_edge_table(soil_amp_network)
```

---

get_node_table	<i>Get node property table for each network</i>
----------------	---

---

**Description**

Get node property table for each network in the list with multiple networks.

**Usage**

```
get_node_table(network_list, ...)
```

**Arguments**

`network_list` a list with multiple networks; all the networks should be `trans_network` object created from `trans_network` class of `microeco` package.  
`...` parameter passed to `get_node_table` function of `trans_network` class.

**Value**

list, with `res_node_table` in each network

## Examples

```
data(soil_amp_network)
soil_amp_network <- get_node_table(soil_amp_network, node_roles = FALSE)
```

---

meconetcomp                      *Introduction to meconetcomp package*  
([Rhrefhttps://github.com/ChiLiubio/meconetcomp](https://github.com/ChiLiubio/meconetcomp)<https://github.com/ChiLiubio/meconetcomp>)

---

## Description

For the detailed tutorial on meconetcomp package, please follow the links:

Online tutorial website: [https://chiliubio.github.io/microeco\\_tutorial/meconetcomp-package.html](https://chiliubio.github.io/microeco_tutorial/meconetcomp-package.html)

Download tutorial: [https://github.com/ChiLiubio/microeco\\_tutorial/releases](https://github.com/ChiLiubio/microeco_tutorial/releases)

Please open the help document by using `help` function or by clicking the following links collected:

[cal\\_module](#)

[cal\\_network\\_attr](#)

[get\\_node\\_table](#)

[get\\_edge\\_table](#)

[node\\_comp](#)

[edge\\_comp](#)

[edge\\_node\\_distance](#)

[edge\\_tax\\_comp](#)

[subset\\_network](#)

[subnet\\_property](#)

To report bugs or discuss questions, please use Github Issues (<https://github.com/ChiLiubio/meconetcomp/issues>).

Before creating a new issue, please read the guideline ([https://chiliubio.github.io/microeco\\_tutorial/notes.html#github-issues](https://chiliubio.github.io/microeco_tutorial/notes.html#github-issues)).

To cite meconetcomp package in publications, please run the following command to get the reference: `citation("meconetcomp")`

Reference:

Chi Liu, Chaonan Li, Yanqiong Jiang, Raymond Jianxiong Zeng, Minjie Yao, and Xiangzhen Li. 2023. A guide for comparing microbial co-occurrence networks. *iMeta*. 2(1): e71.

---

node_comp	<i>Generate a microtable object with node distributions across networks</i>
-----------	---

---

**Description**

Generate a microtable object with node distributions across networks. Useful for the node information comparisons across different networks.

**Usage**

```
node_comp(network_list, property = "name")
```

**Arguments**

network_list	a list with multiple networks; all the networks should be trans_network object created from <a href="#">trans_network</a> class of microeco package.
property	default "name"; a colname of res_node_table in each network; the default "name" represents using node presence/absence information in the otu_table of final output, in which 1 represents presence of the node in the corresponding network; For other options (such as degree), the results in the output otu_table are the actual values of res_node_table.

**Value**

microtable object

**Examples**

```
data(soil_amp_network)
test <- node_comp(soil_amp_network)
# test is a microtable object
```

---

soil_amp	<i>The soil_amp data</i>
----------	--------------------------

---

**Description**

The soil\_amp data is the 16S rRNA gene amplicon sequencing dataset of Chinese wetland soils. Reference: An et al. 2019 <doi:10.1016/j.geoderma.2018.09.035>; Liu et al. 2022 <10.1016/j.geoderma.2022.115866>

**Usage**

```
data(soil_amp)
```

---

soil_amp_network	<i>The soil_amp_network data</i>
------------------	----------------------------------

---

**Description**

The soil\_amp\_network data is a list storing three trans\_network objects created based on soil\_amp data. Three networks are created for IW, CW and TW groups, respectively.

**Usage**

```
data(soil_amp_network)
```

---

soil_measure_diversity	<i>The soil_measure_diversity data</i>
------------------------	--

---

**Description**

The soil\_measure\_diversity data is a table storing all the abiotic factors and functional diversity based on the metagenomic sequencing and MetaCyc pathway analysis.

**Usage**

```
data(soil_measure_diversity)
```

---

stool_met	<i>The stool_met data</i>
-----------	---------------------------

---

**Description**

The stool\_met data is the metagenomic species abundance dataset of stool samples selected from R ExperimentHub package. It has 198 samples, collected from the people with alcohol drinking habit, and 92 species.

**Usage**

```
data(stool_met)
```

---

subset_property	<i>Calculate properties of sub-networks selected according to features in samples</i>
-----------------	---

---

### Description

Extracting sub-network according to the presence of features in each sample across networks and calculate the sub-network properties.

### Usage

```
subset_property(network_list)
```

### Arguments

network_list	a list with multiple networks; all the networks should be trans_network object created from <a href="#">trans_network</a> class of microeco package.
--------------	--

### Value

data.frame

### Examples

```
data(soil_amp_network)
test <- subset_property(soil_amp_network)
```

---

subset_network	<i>Extract subset of network according to the edge intersection of networks</i>
----------------	---

---

### Description

Extracting a network according to the edge intersection of networks.

### Usage

```
subset_network(network_list, venn = NULL, name = NULL)
```

**Arguments**

network_list	a list with multiple networks; all the networks should be trans_network object created from <code>trans_network</code> class of microeco package.
venn	default NULL; a microtable object which must be converted by trans_comm function of trans_venn class.
name	default NULL; integer or character; must be a number or one of colnames of the otu_table in the input venn parameter.

**Value**

trans\_network object, with only the extracted edges in the network

**Examples**

```
data(soil_amp_network)
# first obtain edge distribution
tmp <- edge_comp(soil_amp_network)
# obtain edge intersection using trans_venn class
tmp1 <- microeco::trans_venn$new(tmp)
# convert intersection result to microtable object
tmp2 <- tmp1$trans_comm()
# extract the intersection of all the three networks ("IW", "TW" and "CW")
test <- subset_network(soil_amp_network, venn = tmp2, name = "IW&TW&CW")
# test is a trans_network object
```

# Index

## \* **Description**

meconetcomp, 9

## \* **data.frame**

soil\_measure\_diversity, 11

## \* **list**

soil\_amp\_network, 11

## \* **microtable**

soil\_amp, 10

stool\_met, 11

cal\_module, 2, 9

cal\_network\_attr, 3, 9

edge\_comp, 3, 9

edge\_node\_distance, 4, 9

edge\_tax\_comp, 7, 9

get\_edge\_table, 8, 9

get\_node\_table, 8, 9

meconetcomp, 9

node\_comp, 9, 10

soil\_amp, 10

soil\_amp\_network, 11

soil\_measure\_diversity, 11

stool\_met, 11

subnet\_property, 9, 12

subset\_network, 9, 12

trans\_network, 2–4, 7, 8, 10, 12, 13