

Package: network.tools (via r-universe)

August 15, 2024

Title Tools to Analyse and Visualise Bipartite Networks

Version 0.0.5

Description Tools to analyse and visualise bipartite networks.

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

URL <https://github.com/Pakillo/network.tools>,
<https://pakillo.github.io/network.tools/>

BugReports <https://github.com/Pakillo/network.tools/issues>

Depends R (>= 4.1)

Imports dplyr, ggplot2, gtools, tidyr

LazyData true

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository <https://pakillo.r-universe.dev>

RemoteUrl <https://github.com/Pakillo/network.tools>

RemoteRef HEAD

RemoteSha b7e9ac6b63bcd7fc53f810b21c91545afbbcf30

Contents

drop_empty_cols	2
drop_empty_rows	2
expand_unobs	3
indiv_spec	4
long2wide	5
plot_web_heatmap	6
web	7
wide2long	7

Index**9**

drop_empty_cols	<i>Drop empty columns</i>
-----------------	---------------------------

Description

Remove columns from an interaction matrix where all values are zero.

Usage

```
drop_empty_cols(mat = NULL, na.rm = TRUE)
```

Arguments

mat	A matrix containing interaction data.
na.rm	Logical. If TRUE, NA values will be ignored for deciding if a column should be removed. If FALSE, columns having NA values will never be removed.

Value

A matrix

Examples

```
mat <- matrix(c(0,0,0, 1,1,0, 0,0,NA), ncol = 3, byrow = TRUE)
mat
drop_empty_cols(mat)
drop_empty_cols(mat, na.rm = FALSE)
```

drop_empty_rows	<i>Drop empty rows</i>
-----------------	------------------------

Description

Remove rows from an interaction matrix where all values are zero.

Usage

```
drop_empty_rows(mat = NULL, na.rm = TRUE)
```

Arguments

mat	A matrix containing interaction data.
na.rm	Logical. If TRUE, NA values will be ignored for deciding if a row should be removed. If FALSE, rows having NA values will never be removed.

Value

A matrix

Examples

```
mat <- matrix(c(0,0,0, 1,1,0, 0,0,NA), ncol = 3, byrow = TRUE)
mat
drop_empty_rows(mat)
drop_empty_rows(mat, na.rm = FALSE)
```

expand_unobs	<i>Expand interaction dataset to include zeroes (unobserved interactions)</i>
--------------	---

Description

Expand interaction dataset to include zeroes (unobserved interactions)

Usage

```
expand_unobs(
  df = NULL,
  plant.var = "Plant",
  animal.var = "Animal",
  int.var = "Visits",
  fill.value = 0
)
```

Arguments

df	A data frame with interaction presence or frequency data
plant.var	character. Name of the column representing plants.
animal.var	character. Name of the column representing animals.
int.var	character. Name of the column representing interaction presence or frequency.
fill.value	Value to use in int.var for unobserved interactions.

Value

A data frame

Examples

```
data(web)
df <- web[sample(1:nrow(web), size = 30), ]
df.expand <- expand_unobs(df)
```

indiv_spec

*Niche width and individual specialisation indices***Description**

Calculate indices of niche width and individual specialisation, following Bolnick et al. 2002 ([doi:10.1890/00129658\(2002\)083\[2936:MILRS\]2.0.CO;2](https://doi.org/10.1890/00129658(2002)083[2936:MILRS]2.0.CO;2)).

TNW calculates the Total Niche Width of the population, using Shannon diversity.

WIC calculates the Within-Individual Component of niche width as a weighted average of individuals' Shannon diversity (i.e. the Shannon diversity value of each individual plant is weighted by the proportion of visits or interactions received by that plant). WIC can also return the individual Shannon diversity values, if `indiv = TRUE`.

`indiv_spec` returns a data.frame with the following columns:

- WIC (Within-Individual Component)
- TNW (Total Niche Width)
- IndSpec (Individual Specialisation index, calculated as the ratio WIC / TNW).

Usage

```
indiv_spec(net)
```

```
TNW(net)
```

```
WIC(net, indiv = FALSE)
```

Arguments

`net` A matrix or data.frame containing interaction data. Plants in rows, Animals in columns. Numbers need not be integers (i.e. counts), can be relative abundances or interaction frequencies.

`indiv` Logical. If TRUE, return individual Shannon diversity values. If FALSE, return the weighted average of individual Shannon diversity, weighted by the proportion of interactions represented by each plant (i.e. the WIC).

Value

A numeric value or vector for WIC and TNW, a data.frame in the case of `indiv_spec`

Note

Ideally `net` should not contain missing data (NA). If present, they will be ignored.

References

Bolnick DI, LH Yang, JA Fordyce, JM Davis, and Svanback R. 2002. Measuring individual-level resource specialization. *Ecology* 83: 2936-2941.

See Also[RInSp::WTdMC\(\)](#)**Examples**

```
data(web)
net <- long2wide(web)

indiv_spec(net)
WIC(net)
WIC(net, indiv = TRUE)
TNW(net)
```

`long2wide`*Transform interaction data from long to wide format*

Description

Transform interaction data from long to wide format

Usage

```
long2wide(
  df = NULL,
  plant.var = "Plant",
  animal.var = "Animal",
  int.var = "Visits",
  exclude.noint = TRUE,
  fill.value = NA,
  sort = TRUE
)
```

Arguments

<code>df</code>	A data frame with interaction presence or frequency data
<code>plant.var</code>	character. Name of the column representing plants.
<code>animal.var</code>	character. Name of the column representing animals.
<code>int.var</code>	character. Name of the column representing interaction presence or frequency.
<code>exclude.noint</code>	Logical. Exclude plants/animals with no interactions? Default is TRUE.
<code>fill.value</code>	Value to fill empty cells in the matrix. Default is NA.
<code>sort</code>	Sort matrix rows and columns by frequency?

Value

A matrix with plants and animals as row and column names, respectively.

Examples

```
data(web)
long2wide(web)
```

plot_web_heatmap	<i>Plot bipartite interaction web as a heatmap</i>
------------------	--

Description

Plot bipartite interaction web as a heatmap

Usage

```
plot_web_heatmap(
  df,
  plant.var = "Plant",
  animal.var = "Animal",
  int.var = "Visits",
  binarize = FALSE,
  sort = TRUE,
  zero.na = TRUE,
  na.colour = "white"
)
```

Arguments

df	A data frame with interaction presence or frequency data
plant.var	character. Name of the column representing plants.
animal.var	character. Name of the column representing animals.
int.var	character. Name of the column representing interaction presence or frequency.
binarize	Logical. Discretize int.var into two categories? (Default is FALSE).
sort	Logical. If TRUE, sort rows and columns by prevalence to show nestedness.
zero.na	Logical. Show zeros as NA?
na.colour	Colour to be used for NA.

Value

A ggplot object.

Examples

```
data(web)
plot_web_heatmap(web)
plot_web_heatmap(web, zero.na = FALSE)
plot_web_heatmap(web, sort = FALSE)
plot_web_heatmap(web, binarize = TRUE)
```

web	<i>Bipartite network</i>
-----	--------------------------

Description

Example data frame containing data for a bipartite web of number of visits from 8 animals to 4 plants.

Usage

```
web
```

Format

web:

A data frame with 32 rows and 3 columns:

Plant Plant ID

Animal Animal ID

Visits Number of visits

Source

dataset simulated

wide2long	<i>Transform interaction data from wide to long format</i>
-----------	--

Description

Transform interaction data from wide to long format

Usage

```
wide2long(mat = NULL, int.name = "Visits")
```

Arguments

mat A matrix containing interaction data. Plants in rows, Animals in columns.

int.name character. Column name for the interaction values.

Value

A data frame with three columns: "Plant", "Animal", and interaction values.

Examples

```
data(web)
mat <- long2wide(web)
mat
df <- wide2long(mat)
df
```

Index

* datasets

web, [7](#)

drop_empty_cols, [2](#)

drop_empty_rows, [2](#)

expand_unobs, [3](#)

indiv_spec, [4](#)

long2wide, [5](#)

plot_web_heatmap, [6](#)

RInSp:WTdMC(), [5](#)

TNW (indiv_spec), [4](#)

web, [7](#)

WIC (indiv_spec), [4](#)

wide2long, [7](#)