

# Package ‘bipartite’

November 5, 2009

**Type** Package

**Title** Visualising bipartite networks and calculating some ecological indices.

**Version** 1.03

**Date** 2009-11-06

**Author** Carsten F. Dormann and Bernd Gruber, with additional code from Mariano Devoto, Jochen Freund, Jose Iriondo and Diego Vazquez, also based on C-code developed by Nils Bluethgen and by Miguel Rodriguez-Girones.

**Maintainer** Carsten F. Dormann <carsten.dormann@ufz.de>

**Depends** MASS, vegan

**Suggests** fields, sna, spam

**Description** Bipartite provides functions to visualise webs and calculate a series of indices commonly used to describe pattern in ecological webs. It focusses on webs consisting of only two trophic levels, e.g. pollination webs or predator-prey-webs. Visualisation is important to get an idea of what we are actually looking at, while the indices summarise different aspects of the webs topology.

**License** GPL

**Repository** CRAN

**Date/Publication** 2009-11-05 19:43:30

## R topics documented:

bipartite-package	3
as.one.mode	9
barrett1987	10
C.score	11
compart	12
degreedistr	14
dfun	15

discrepancy . . . . .	18
elberling1999 . . . . .	19
empty . . . . .	20
extinction . . . . .	21
functspec . . . . .	22
genweb . . . . .	24
H2fun . . . . .	25
inouye1988 . . . . .	26
kato1990 . . . . .	27
kevan1970 . . . . .	28
memmott1999 . . . . .	29
mosquin1967 . . . . .	29
motten1982 . . . . .	30
nested . . . . .	31
nestedness . . . . .	33
networklevel . . . . .	35
nodespec . . . . .	42
null.t.test . . . . .	44
nullmodel . . . . .	45
olesen2002aigrettes . . . . .	46
olesen2002flores . . . . .	47
ollerton2003 . . . . .	48
PAC . . . . .	49
plotweb . . . . .	50
plotweb2 . . . . .	54
robustness . . . . .	57
Safariland . . . . .	58
schemske1978 . . . . .	59
second.extinct . . . . .	60
shuffle.web . . . . .	61
slope.bipartite . . . . .	63
small1976 . . . . .	64
sortweb . . . . .	65
specieslevel . . . . .	66
swap.web . . . . .	69
togetherness . . . . .	71
V.ratio . . . . .	72
vazarr . . . . .	73
vazcer . . . . .	74
vazllao . . . . .	74
vazmasc . . . . .	75
vazmasnc . . . . .	75
vazquec . . . . .	76
vazquenc . . . . .	76
vazquez.example . . . . .	77
visweb . . . . .	79
wine . . . . .	81

## Description

Bipartite provides functions to visualise webs and calculate a series of indices commonly used to describe pattern in ecological webs. It focusses on webs consisting of only two trophic levels, e.g. pollination webs or predator-prey-webs. Visualisation is important to get an idea of what we are actually looking at, while the indices summarise different aspects of the webs topology.

## Details

We only had three types of bipartite webs in mind when writing this package: seed-disperser, plant-pollinator and host-parasitoid systems. In how far it makes sense to use these functionalities for other systems (or indeed for these systems) lies in the hands of the user. Please refer to the literature cited for details on the theory behind the indices.

Input for most analyses is an interaction matrix of  $m$  higher level species with  $n$  lower level species, i.e. an  $n \times m$  matrix, where higher trophic level species are in columns, lower level in rows. Column and row names can be provided. This is fundamentally different from multi-dimensional webs, which are organised as  $k \times k$  matrix, i.e. each species against each other. Such a format is incompatible with the functions we provide here.

The first step is to visualise the interaction web. Two functions are on offer here: one ([visweb](#)) simply plots the matrix in colours depicting the strength of an interaction and options for rearranging columns and rows (e.g. to identify compartments or nesting). The other function ([plotweb](#)) plots the actual web with participants (as two rows of rectangles) connected by lines (proportional to interaction strength).

The second step is to calculate various indices describing network topography. There are two different levels this can be achieved at: the entire web (using function [networklevel](#)) or the individual species (using function [specieslevel](#)). All other functions in the package are helpers, although some can be called on their own and return the respective result ([dfun](#), [H2fun](#) and [second.extinct](#) with [slope.bipartite](#)).

See function description for details and examples.

Package: bipartite  
Type: Package  
Version: 1.03  
Date: 2009-11-06  
License: GPL

## versionlog

- 1.03 (release date: 06-Nov-2009)

**Bug fix in [V.ratio](#):** A small mistake for a human, but a huge bug for mankind. Sorry. (Detail: I forgot to square  $\sigma$ , leading to strange results.) Thanks to Giorgio Mancinelli

for reporting!

**Error message for non-existent indices in `networklevel`:** So far, `networklevel` returned NULL when an index was selected that does not exist (e.g. "shannon diversity" instead of "diversity"). Now, a helpful (?) error message is returned.

**Change in defaults to index "interaction evenness" in `networklevel`:** After a fruitful discussion with Becky Morris and Jason Tylianakis, I changed the default to "sum". I also reproduce some of our communication in the help to this function, under details, to make the ecological assumptions behind either option a bit more transparent. There are good reasons for either option.

- 1.02 (release date: 11-Sep-2009)

**Function `plotweb`:** Now more trophic networks can be plotted by staggering bipartite networks on top of each other. See multitrophic examples in `plotweb`.

**Minor corrections to `dfun`:** This function did not return *exactly* the values of the website-version. Jochen Fründ corrected this. Please read the help of the function (final paragraph) for details.

**New function `nested`:** This convenience function collects the various ways to calculate nestedness of a network in order to facilitate comparison of nestedness analyses. To do so, it heavily borrows from `vegan`.

**Bug fix to `discrepancy`:** Would return a silly value (half of the number of rows) for empty matrices. This had no effect when called by `networklevel`, since the matrix would have been emptied. Thanks to Roberto Molowny for reporting!

**Bug fix to `networklevel`:** Due to a missing space, the option 'ALLBUTDD' did not work properly. Thanks to Etienne Laliberté for reporting!

- 1.00 (release date: 06-Aug-2009)

**Complete overhaul of `networklevel`:** After a workshop on bipartite networks in ecology, a few more indices were added (Fisher's alpha diversity of interactions, mean interaction diversity, mean number of predators) and the whole output reorganised. It now follows a gradient from less to more interesting (in our view) indices, and from indices for binary to those for quantitative networks. Also, I added several options for which indices to report ('index="info", "binary", "quantitative", "topology"). Most interestingly, perhaps, there is now a quantitative, Shannon-based series of indices. Starting with the "mean interaction diversity" (i.e. the Shannon-diversity of interactions of a species, averaged across all species in that trophic level), over "Shannon diversity" of interactions, to H2 (i.e. Shannon diversity scaled between max and min possible for this web characteristics).

**Additions to, and overhaul of `specieslevel`:** Similar to the above item, some indices were added, and the output simplified when calling the option 'index="ALLBUTD"' (only one D here!): a list with two matrices is now returned. Fisher's alpha for each species, vulnerability/generality and effective number of species for each target species are also now included. Index sequence has changed.

**New function `PAC`:** Calculates the Potential for Apparent Competition following the formula in Müller et al. (1999) and Morris et al. (2005) and the suggestion by Becky Morris and Owen Lewis. More than a theoretical concept, it was experimentally shown to be relevant (Morris et al. 2004).

**Bug fix in `H2fun` and `dfun`:** A line of code went missing at some point, so the maximum packing density was not optimal (but still good) in these functions. As a result, reported H2- and d-values were sometimes 0 when they should be only very close to 0.

**Switch for error reporting in `degredistr`:** By default now suppresses error reporting when `nls` fails to fit a degree distribution due to too few data points. This leaves the user of `networklevel` somewhat less confused. Warning message now also indicates for which trophic level there were too few data points.

**Bug fix in `C.score`:** Failed when the web was very dense or very sparse, because the maximum number of checkerboard patterns was 0 then.

- 0.94 (release date: 01-Aug-2009)

**New function `nullmodel`:** A convenience wrapper function to generate different types of null models.

**Small changes to `networklevel`:** This function returns a list of indices. If we exclude the computation of degree distribution fits, this would be coerced to a vector. We added the option `'index="ALLBUTDD"` to calculate all indices BUT degree distributions. The output is then returned as vector. This is much more convenient when using `networklevel` on many data sets (using `sapply`).

- 0.93 (release date: 30-Jun-2009)

**bug fix in `slope.bipartite` and `robustness`:** The function always selected column 3 of the object, instead of 2 for lower and 3 for higher trophic level; thanks to Antonio Rivera for spotting and reporting this error! Notice that this error must have slipped in somewhere around version 0.90/0.91, because I checked and the results reported in the Open Ecology Journal paper are valid! I seem to remember that I (CFD) modified `slope.bipartite` when `robustness` was added: never change a winning team!

- 0.92 (release date: 02-Jun-2009)

**more colour options in `visweb`:** the arguments `'box.border'` and `'box.col'` now allow a specification of the colour of the boxes and their borders.

**bug fix `empty`:** the function returned 0 for a 1x1 matrix. Although not written for such a case, it should still do what it says on the tin. Thanks to Mariano Devoto for spotting and reporting!

**bug fix `wine`:** returned NA for square matrices.

**bug fix `plot.wine`:** gave decimal places for row and column names for very small networks.

- 0.91 (release date: 06-May-2009)

**new function `wine`:** This function replaces the (now deprecated) function `nestedness.corso` in calculating a (weighted) nestedness for bipartite networks. It was developed and implemented by Marcelino de la Cruz, Juan M. Pastor, Javier Galeano and Jose Iriondo. It is also called by `networklevel`. A plotting function is also available, depicting the contributions of each observed link to the web's nestedness. - `nestedness.corso` is now removed from the package. The main reason is that it served as an interim solution for `wine`, and the Corso-way of calculating nestedness is just one more of already too many. If you intend to use `wine` on binary data and interpret that as a Corso-equivalent, beware of the following two main differences (thanks to Jose Iriondo for summarising them): *First, the nestedness of Corso et al. varies between 0 and 1, with the highest nestedness is reached at 0 and 1 corresponds to random, whereas in `wine` is just the opposite (this is because the Manhattan distances are calculated with regard to opposite sides of the matrix). Secondly, the 'd' in the nestedness of Corso is the sums of the 'dij's whereas in `wine`, the 'd' (= 'win' value of the object produced by `wine`) is the average of the*

*dij*'s above 0. So, we recommend NOT to use `wine` for calculating Corso's nestedness, but rather download the source code for `nestedness.corso` from an older version of **bipartite**.

**new function `robustness`**: A better way to quantify the effect of species loss on the extinctions in the other trophic level; kindly provided by Mariano Devoto. This index is also part of `networklevel`.

**new data set `ollerton2003`**: Another quantitative pollination network from the NCEAS database (see `ollerton2003`).

- 0.90 (release date: 24-Mar-2009)

**example `vazquez.example`**: We added several new functionalities mainly to be able to analysis data and use network statistics as suggested by Vazquez et al. 2009. You can access this example by typing `?vazquez.example`.

**new feature in `visweb`**: Can now plot different sized circles to represent interactions, as proposed by Vazquez et al. (2009).

**new feature in `networklevel`**: We added an option to calculate interaction evenness either based on all possible links or just on realized links.

**new function `sortweb`**: Can be used to sort webs in different ways.

**data set `inouye1988`**: Another pollination network from the NCEAS database (see `inouye1988`).

**Function `compart`**: We replaced the CA-based approach to detecting compartments by a comprehensive and recursive approach. The latter is not affected by ties in the data set (i.e. species with the same number of links). In quantitative webs and for the networks included in `bipartite`, the old function was working fine, but in more recent trials it failed to detect 2-species compartments. In turn, we had to adapt `networklevel` and `plotweb`.

- 0.85 (release date: 10-Mar-2009)

**Function `plotweb`** New feature: text labels can now be printed in different colours. All colours can be passed as vectors and vectors are recycled if not of appropriate length

- 0.84 (release date: 25-Feb-2009)

**Function `plotweb`** New feature: `arrow="center.up"`, `arrow="center.down"`, `arrow="center.both"`: this results in the standard presentation of bipartite networks, where interactions are displayed as centered triangles.

- 0.83 (release date: 9-Feb-2009) pdf of OEJ-paper added as vignette

- 0.82 (release date: 27-Jan-2009)

**Function `plotweb`** Two new features: 1. colors of borders of boxes and interactions can be specified via setting `bor.col`. 2. Labels can now be rotated by specifying `text.rot=90`.

- 0.81 (release date: 06-Jan-2009)

**Function `nodespec` replaces the now obsolete function `functspec`**. The inventor of functional specialisation, Bo Dalsgaard, understands the term 'functional specialisation' to be restricted to the way in which plants are specialised to pollinators. Although I disagree (and think that 'functional' actually means very little until defined for a given problem), I renamed the 'functional specialisation index' into 'node specialisation index'. This is not a particularly clever name, but at least it indicates that the position of nodes in a network is important when it is calculated.

**Bug in `networklevel`**: Call to `nestedness.corso` overwrote results of `nestedness`.

**Various changes related to the analysis of very small webs.** Very small webs should probably not be used at all for testing theories! Indices are usually VERY sensitive to the exact number of species, number of observations etc. Still, sometimes we simply want to calculate some index, and then **bipartite** should handle such small networks, too. So, when going through several dozens of very small webs (sometimes only containing one species in one of the two trophic levels), several functions did not perform correctly (usually to such minor programming issues such as matrices being converted to vectors when `[]` was used with `'drop=TRUE'`, i.e. the default). Changes affected the functions `discrepancy`, `nestedness.corso`, `shuffle.web`, `compart` and `empty`. Their output remains identical, only they now also work for small webs.

**Changes to `nestedness.corso` with `'weighted=TRUE'`** Galeano et al. do not describe how to deal with ties, nor do they make clear if the packed matrix should be sorted by marginal sum of links or marginal sum of interactions! Previously, we used marginal sum of interactions (because it is a weighted index), but now we moved to marginal sum of links, because that is how I interpret their paper after a third (or forth) re-reading. Furthermore, because the real maximum chaos cannot be derived (to my knowledge) algorithmically, we use the 95% quantile of 500 randomisations as maximum. This will lead to a consistently overestimated nestedness, but it is less sensitive to the number of replicates than the max. (Also, there was an error in the description of the value returned: 0 is nested, 1 is chaos!)

- 0.8 (release date: 21-Dec-2008)

**bug fix in `shuffle.web`** Function didn't work correctly when the web contained more rows than columns. (Thanks to Anna Traveset for spotting and reporting!)

**minor modifications in `networklevel`** Sometimes additional information is available and species can be included in a network, although they have no interactions with other species. In this case, one might want to use these species in the network, too. A new option (`empty.web`), allows to keep empty rows and columns, although for some functions an emptied web must be used (e.g. degree distributions).

**bug fix in `H2fun`** As in the last `H2fun` bug fix, sometimes H2 became negative.

**bug (?) fix in `networklevel`** Shannon diversity is based on the log of interactions. If this value is 0, as it is for most network entries, an NA is produced. As a consequence, Shannon's H (now also given in output) is based not on all interactions, but only those > 0. In consequence, Shannon evenness should also only be SH divided by the number of realised interactions (`log(sum(web>0))`). That is now the case.

- 0.74 (release date: 24-Oct-2008)

**functional specialisation (`functspec`) bug fix** Paths were double the true length, hence minimum was 2, rather than 1.

**`H2fun` bug fix** Since the search for H2min is heuristic, H2uncorr can sometimes be lower than H2min; in that case, H2fun returned a value greater 1, while it should be one exactly.

**new function `nestedness.corso`** Calculates (weighted) nestedness according to Corso et al. (2008) and Galeano et al. (2008).

**new function `discrepancy`** Calculates discrepancy according to Brualdi & Sanderson (1999), deemed to be best ever measure of nestedness; also gives an example for a binary null model analysis based on **vegan**'s `commsimulator`.

**other** Correction of several minor typos on the help pages; removal of "~" in help files; same citation style throughout; new cross references (especially for the nestedness functions);

in `networklevel`, nestedness is now calculated using `vegan`'s `nestedtemp` due to matrix inversion problems reported for `binmatnest`.

- 0.73 (release date: 1-Sept-2008)
  - new feature** `plotweb` Named abundance-vector for each level can be used.
  - new function** `plotweb2 (not debugged!)` For plotting tripartite networks.
- 0.72 (release date: 12-June-2008)
  - new function: functional specialisation with** `functspec` See Dalgaard et al. (2008).
  - new function: interface to sna through** `as.one.mode` Allows calculation of path lengths, centrality, betweenness and other indices developed for one-mode networks.
  - bug-fix** Error in `plotweb` when no species labels were given.

### Author(s)

Carsten F. Dormann and Bernd Gruber, with additional code from Jochen Fründ

Maintainer: Carsten Dormann <carsten.dormann@ufz.de> and Bernd Gruber <bernd.gruber@ufz.de>

### References

- Bascompte, J., Jordano, P. and Olesen, J. M. (2006) Asymmetric coevolutionary networks facilitate biodiversity maintenance. *Science* **312**, 431–433
- Bersier, L. F., Banasek-Richter, C. and Cattin, M. F. (2002) Quantitative descriptors of food-web matrices. *Ecology* **83**, 2394–2407
- Blüthgen, N., Menzel, F. and Blüthgen, N. (2006) Measuring specialization in species interaction networks. *BMC Ecology* **6**, 12
- Blüthgen, N., Menzel, F., Hovestadt, T., Fiala, B. and Blüthgen, N. (2007) Specialization, constraints, and conflicting interests in mutualistic networks. *Current Biology* **17**, 1–6
- Corso G., de Araújo A.I.L. and de Almeida A.M. (2008) A new nestedness estimator in community networks. *arXiv*, 0803.0007v1 [physics.bio-ph]
- Dalsgaard, B., A. M. Martín González, J. M. Olesen, A. Timmermann, L. H. Andersen, and J. Ollerton. (2008) Pollination networks and functional specialization: a test using Lesser Antillean plant-hummingbird assemblages. *Oikos* **117**, 789–793
- Dormann, C.F., Fründ, J., Blüthgen, N., and Gruber, B. (2009) Indices, graphs and null models: analysing bipartite ecological networks. *The Open Ecology Journal* **2**, 7–24.
- Galeano J., Pastor J.M. and Iriondo J.M. (2008) Weighted-Interaction Nestedness Estimator (WINE): A new estimator to calculate over frequency matrices. *arXiv* 0808.3397v1 [physics.bio-ph]
- Memmott, J., Waser, N. M. and Price, M. V. (2004) Tolerance of pollination networks to species extinctions. *Proceedings of the Royal Society B* **271**, 2605–2611
- Morris, R. J., Lewis, O. T. and Godfray, H. C. J. 2004. Experimental evidence for apparent competition in a tropical forest food web. *Nature* **428**, 310–313.
- Morris, R. J., Lewis, O. T. and Godfray, H. C. J. 2005. Apparent competition and insect community structure: towards a spatial perspective. *Annales Zoologica Fennici* **42**, 449–462.
- Müller, C. B., Adriaanse, I. C. T., Belshaw, R. and Godfray, H. C. J. 1999 The structure of an aphid-parasitoid community. *Journal of Animal Ecology* **68**, 346–370.

Tylianakis, J. M., Tscharntke, T. and Lewis, O. T. (2007) Habitat modification alters the structure of tropical host-parasitoid food webs. *Nature* **445**, 202–205

Vazquez, D. P. and Aizen, M. A. (2004) Asymmetric specialization: A pervasive feature of plant-pollinator interactions. *Ecology* **85**, 1251–1257

Vazquez, P.D., Chacoff, N.,P. and Cagnolo, L. (2009) Evaluating multiple determinants of the structure of plant-animal mutualistic networks. *Ecology* in press.

## Examples

```
data(Safariland)
plotweb(Safariland)
visweb(Safariland)
networklevel(Safariland)
specieslevel(Safariland)
```

---

as.one.mode

*Conversion of a network matrix*

---

## Description

This helper function converts a bipartite matrix into a one-mode matrix.

## Usage

```
as.one.mode(web, fill = 0)
```

## Arguments

<code>web</code>	A matrix with lower trophic level species as rows, higher trophic level species as columns and number of interactions as entries.
<code>fill</code>	What shall unobserved combinations be represented as in the one-mode matrix (see below)? Defaults to 0.

## Details

In bipartite (or: two-mode) networks, participants are of different types (e.g. pollinators and plants, actors and parties in social research). Hence, a party cannot connect to another party except through actors. A pollinator interacts with another pollinator only through the host plant.

Much network theory, however, is based on one-mode networks, where all participants are listed in one vector, i.e. plants and pollinators alike, actors together with events. This function here transforms the more condensed bipartite representation into a one-mode-representation, filling the unobserved type of interactions ( i.e. plants with plants and pollinators with pollinators) with 0 (unless you specify it differently in ‘fill’).

The lower trophic level (e.g. plants or rows) is listed first, then the higher trophic level (e.g. pollinators or columns). Hence, pollinator 2 becomes species number  $r+2$ , where  $r$  is the number of rows of the network matrix.

The benefit of this conversion is access to the wonderful R-package Social Network Analysis (**sna**), with its many one-mode indices (such as `betweenness`, `closeness`, `centralization`, `degree`, `kpath.census` and so forth). Furthermore, `gplot` in that package also provides cool network depictions well worth checking out.

With respect to **bipartite**, `as.one.mode` is employed in the function `functspec`, which itself uses the **sna**-function `geodist`.

### Value

A matrix of dimension  $(n+k) \times (n+k)$ , where  $n$  and  $k$  are the dimensions of the input web. Both dimensions are given the names of the original web (first the lower, then the higher trophic level).

### Note

This function is not at all clever or complex. In fact, most users may not consider it worth providing at all.

### Author(s)

Carsten F. Dormann <carsten.dormann@ufz.de>

### Examples

```
data(Safariland)
image(Safariland)
image(as.one.mode(Safariland))
```

---

barrett1987

*Individuals caught in a pollination web in boreal Canada.*

---

### Description

This study took place in the boreal forest of central New Brunswick, Canada, from May to September of 1978, 1979, and 1980. The objective was to investigate the role of animals in pollination and seed dispersal. The study was designed to provide basic descriptive information on breeding systems, pollination biology, and phenology of understory herbs.

The authors recorded their data by counting the number of individual flower visitors caught on each plant species. The total number of individuals collected on each plant species provide a rough estimate of the level of visitation that each species received. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

### Usage

```
data(barrett1987)
```

## References

Barrett, S. C. H. and Helenurm, K. (1987) The Reproductive-Biology Of Boreal Forest Herbs. 1. Breeding Systems And Pollination. *Canadian Journal Of Botany-Revue Canadienne De Botanique* **65**, 2036–2046

## Examples

```
data(barrett1987)
```

---

<code>C.score</code>	<i>Calculates the (normalised) mean number of checkerboard combinations (C-score) in a matrix</i>
----------------------	---

---

## Description

Calculates the C-score for all pollinator species; the C-score represents the average number of checkerboard units for each unique species pair.

## Usage

```
C.score(web, normalise = TRUE, FUN = mean, ...)
```

## Arguments

<code>web</code>	A matrix with pollinators as columns and plants as rows. Alternatively, when used on e.g. species occurrences across islands, rows are islands.
<code>normalise</code>	Logical; if TRUE (default), the C-score is ranged between 0 (no checkerboards) and 1 (only checkerboards). For FALSE the standard value of mean number of checkerboard pairs is returned. This is somewhat awkward for comparing different data sets, that's what the normalisation is for.
<code>FUN</code>	Function to use when summarising the C-scores for each pairwise comparison. Defaults to <code>mean</code> , but other useful functions could be <code>median</code> (because C-scores are rather skewed) or <code>hist</code> (for a nice graph).
<code>...</code>	Options to be passed on to FUN, e.g. <code>'na.rm=T'</code> for matrices with many zeros and <code>'normalise=TRUE'</code> .

## Details

As a first step, any quantitative matrix is converted to a binary matrix of presences and absences.

Then, the formula given in Stone and Roberts (1990) is calculated for all species combinations, by calling `designdist` from the package **vegan**. See code for details.

## Value

Returns whatever the 'FUN' produces as output. Default would be a single value, i.e.~the mean C-score of the web.

**Note**

The normalisation uses the following logic: let A and B be the number of occurrences of each of the two species, respectively. Then J is the number of sites where both occur and P the total number of sites. The maximum value for C in each comparison is  $A*B$ , when they are exactly complementary and hence  $J=0$ . However, if  $(A+B) > \text{length of vector}(P)$ , then there will be some co-occurrences and hence  $J > 0 = (A+B-P)$ . The general maximum then becomes  $(A-A-B+P)(B-A-B+P) = (P-B)(P-A)$ . For  $(A+B) < P$ , maximum is  $AB$ .

The minimum is set to 0.

**Author(s)**

Carsten F. Dormann

**References**

- Gotelli, N.J. and Rohde, K. (2002) Co-occurrence of ectoparasites of marine fishes: a null model analysis. *Ecology Letters* **5**, 86–94
- Stone, L. and Roberts, A. (1990) The checkerboard score and species distributions. *Oecologia* **85**, 74–79

**See Also**

[V.ratio](#) for another measure of species associations, `nestedchecker` in **vegan**, which also calculates the C-score (even yielding the same value), but not its normalised version

**Examples**

```
m <- matrix(c(1,0,0, 1,1,0, 1,1,0, 0,1,1, 0,0,1), 5,3,TRUE)
C.score(m)
C.score(m, normalise=FALSE)
C.score(m, normalise=FALSE, FUN=print)
```

---

compart

*Detects compartments*

---

**Description**

Finds number of compartments, based on multivariate ordination techniques, and labels interactions according to the compartment they belong to.

**Usage**

```
compart(web)
```

**Arguments**

`web` A bipartite interaction web, i.e.~a matrix with higher (cols) and lower (rows) trophic levels.

**Details**

Internal function, to be called by [networklevel](#).

**Value**

Returns a list with two entries:

`cweb` A matrix similar to `web`, but now with compartment numbers instead of interaction values.

`ncompart` The number of compartments.

**Note**

Note that up to (and including) version 0.85 we used a code based on correspondence analysis (see Lewinsohn et al. 2006). This is, however, faulty for webs with many same-linked species. Hence we resort to a brute-force search for compartments, which is orders of magnitude slower, but at least works correctly.

For large networks, the recursive call of compartment may cause an error. Use `options(expressions=50000)` at the beginning of the script to allow for deeper nesting.

**Author(s)**

Carsten F. Dormann <carsten.dormann@ufz.de>

**References**

Lewinsohn, T. M., P. I. Prado, P. Jordano, J. Bascompte, and J. M. Olesen (2006) Structure in plant-animal interaction assemblages. *Oikos* **113**, 174–184

**See Also**

See also [networklevel](#).

**Examples**

```
# make a nicely compartmented web:
web <- matrix(0, 10,10)
web[1,1:3] <- 1
web[2,4:5] <- 1
web[3:7, 6:8] <- 1
web[8:10, 9:10] <- 1
web <- web[-c(4:5),] #oh, and make it asymmetric!
web <- web[,c(1:5, 9,10, 6:8)] #oh, and make it non-diagonal
compart(web)
```

```
# or, standard, use Safariland as example:
data(Safariland)
compart(Safariland)
```

---

degreedistr	<i>Fits functions to cumulative degree distributions of both trophic levels of a network.</i>
-------------	---

---

### Description

This function first calculates degrees for each species, then constructs a cumulative distribution with them, and finally fits three different functions to these distributions: exponential, power law and truncated power law. Coefficients and fits are returned.

### Usage

```
degreedistr(web, plot.it=TRUE, pure.call=TRUE, silent=TRUE)
```

### Arguments

web	A bipartite network matrix.
plot.it	Logical; returns graphs of fits when set to TRUE (default).
pure.call	Logical; adjusts par for two panels (for TRUE) or leaves this to the wrapper function (FALSE).
silent	Logical; suppresses error reporting in the <code>try</code> -function around <code>nls</code> ; defaults to TRUE.

### Details

Jordano et al. (2003) proposed that plant-animal networks may show scale invariance, as indicated by the presence of a power law in species degrees. They report on consistently better fits of the truncated power law, hypothesising that such patterns may arise from morphological mismatch or phenological uncoupling. Most problematic with the use of this particular approach is the extreme demand for data. The example web `Safariland` in this package is large (1130 interactions), but it provides only 5 different degree levels (for plants, only 4 for pollinators). Hence fitting three different non-linear functions to these few points is stretching it a bit.

### Value

For both trophic levels, a table:

```
... trophic level dd fits
      Contains coefficient estimates, estimate's standard error and P-value, R2 and
      AIC for each of the three model fits, for the respective trophic level.
```

**Note**

The truncated power law fits two coefficients: slope and cut-off. The function only returns the slope. R2-values for non-linear fits are not well liked among statisticians! See the discussion the R-help list (e.g. <http://finzi.psych.upenn.edu/R/Rhelp02a/archive/90168.html>). Finally, often data are too few to yield any fit. In this case the error message “singular gradient” is returned to signalise this problem!

Post finally, yes, I am aware that degrees are integers and unlikely to be normally distributed, and that thus the `nls` procedure is not really a good idea. My (poor) excuse: I followed the implementation of the above-cited paper and do not believe enough in degree distributions (and power laws, for that matter) to implement a proper likelihood-based approach. Check out the **statnet** bundle for alternative approaches to this problem.

**Author(s)**

Carsten F. Dormann <carsten.dormann@ufz.de>

**References**

Jordano, P., Bascompte, J. and Olesen, J. M. (2003) Invariant properties in coevolutionary networks of plant-animal interactions. *Ecology Letters* **6**, 69-81

**See Also**

[networklevel](#), where `degredistr` is called (without picturing the results)

**Examples**

```
data(Safariland)
degredistr(Safariland)
```

---

dfun

*Calculates standardised specialisation index  $d'$  ( $d$  prime) for each species in the lower trophic level of a bipartite network.*

---

**Description**

This function returns the specialisation index  $d'$ , which expresses how specialised a given species is in relation to what resources are on offer.

**Usage**

```
dfun(web, abuns=NULL)
```

## Arguments

web	Web is a matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
abuns	A vector of abundances for the HIGHER trophic level, usually from additional information. If none is given (default) marginal sums are used.

## Details

The  $d'$  index is derived from Kulback-Leibler distance (as is Shannon's diversity index), and calculates how strongly a species deviates from a random sampling of interacting partners available. It ranges from 0 (no specialisation) to 1 (perfect specialist). In the case of a pollination web, a pollinator may be occurring only on one plant species, but if this species is the most dominant one, there is limited evidence for specialisation. Hence this pollinator would receive a low value. In contrast, a pollinator that occurs only on the two rarest plants would have a very high value of  $d'$ .

The idea of this index is laid out in Blüthgen et al. (2006). It basically calculates the Shannon-diversity for each column (delivering the raw  $d$ -values) and re-ranges them between the theoretical maximum and minimum (yielding values between 0 and 1).  $d_{max}$  is given analytically (see paper or code), but  $d_{min}$  must be found 'heuristically', since the web can only contain integers. The idea behind the heuristic minimum is that  $d$  will be minimal when observed values differ least from expected values based on marginal distributions.

The way this function is implemented, it calculates expected values for each cell based on the product of observed marginal sums (i.e. column and row sums) times  $\text{sum}(\text{web})$ . Then it rounds off to integers and allocates the remaining interactions in two steps: First, all columns and rows with marginal sums of 0 obtain one interaction into the cell with the highest expected value. Secondly, all remaining interactions are distributed according to difference between present and expected value: those cells with highest discrepancy receive an interaction until the sum of all entries in the new web equals those in the original web. Now the  $d$ -values for this web are calculated and used as  $d_{min}$ .

Simple rounding of expected values would lead to empty columns or rows, i.e. the  $d_{min}$ -web would be of lower dimension than the original web.

`dfun` returns the  $d'$  values for the lower trophic level. Use `fun(t(web))` to get the  $d'$ -values for the higher trophic level (as does 'specieslevel').

$d'$  is one of several species-level network indices. It's generalisation to the entire interaction web is called  $H2'$ .

The `abundances` vector allows to incorporate independent estimates of the abundances of the HIGHER trophic level. In a pollination web, pollinator abundances may be very different from those estimated by the interaction matrix column sums. This has also, obviously, large consequences for the specialisation: A plant being pollinated by a bee that is common on this plant, but very rare in general, will show a low specialisation unless bee abundances are corrected for. Data given in the abundance vector are here used in replacement for the row sums, both in the  $d$ -function itself, as well as in the calculation of the minimum  $d$ s.

In contrast to `H2fun`, finding the minimum value of  $d$  violates marginal totals. The idea is that we look at each species in turn. Then, we estimate how its observed number of interactions can be distributed, given the marginal totals (i.e. if 5 interactions were observed, they cannot be put into a

link that only has 3 interactions across all species). So, for each species the number of interactions never exceeds the total across all species, but if we would put the web together from this sequential scan, it may well do so. In our view, this is irrelevant, because we are interested in the potential of each species separately to be perfectly specialised (given the marginal totals), not for the entire web. We leave this to [H2fun](#).

### Value

dprime	d'-value for each species
d	Raw d-value for each species, i.e. before ranging it between 0 and 1.
dmin	Minimum d-value for each species, based on a perfect nesting of the matrix; see details.
dmax	Maximum d-value theoretically possible given the observed number of interactions and the observed marginal distributions.

### Note

As detailed above, deriving the dmin-values 'heuristically' leaves room to some variation. We are very happy with this implementation, but you may want to program something yourself ...

### Author(s)

Jochen Fründ and Carsten F. Dormann

### References

Blüthgen, N., Menzel, F. and Blüthgen, N. (2006) Measuring specialization in species interaction networks. *BMC Ecology* **6**, 12

### See Also

[H2fun](#) for a similar function for the entire network. [specieslevel](#) for a method that, amongst other indices, calls dfun.

### Examples

```
data(Safariland)
dfun(Safariland) # gives d-values for the lower trophic level
# now using independent abundance estimates for higher trophic level:
dfun(Safariland, abuns=runif(ncol(Safariland)))

dfun(t(Safariland)) #gives d-values for the higher trophic level
```

---

 discrepancy

*Calculates discrepancy of a matrix*


---

### Description

Discrepancy is the number of mismatches between a packed (binary) matrix and the maximally packed matrix (with same row sums)

### Usage

```
discrepancy(mat)
```

### Arguments

mat	A matrix (or something that can be transformed into a matrix when <code>as.matrix</code> is called within the function) of species (in columns) on islands (in rows). If quantitative data are given (e.g. in a quantitative pollination network), these are internally transformed into a binary matrix.
-----	---

### Details

Discrepancy is a way to measure the nestedness of a matrix. In a comparative study, Ulrich & Gotelli (2007) showed discrepancy to outperform all other measures and hence recommend its use (together with a fixed-columns, fixed-rows null model, such as implemented in `commsimulator` in **vegan**, see example).

This function follows the logic laid out by Brualdi & Sanderson (1999), although, admittedly, I find their mathematical description highly confusing. Another implementation is given by the function `nsteddisc` in **vegan**. The reason to write a new function is simple: I wasn't aware of `nsteddisc`! (I was sitting on a train and I wanted to use this measure later on, so I put it into a function consulting only the original paper. When looking for the swap algorithm to create null models, which I somehow knew to exist in **vegan**, I stumbled across `nsteddisc`. If you are interested in the swap algorithm and come across this help page, let me re-direct you to `oecosimu` in **vegan**.)

Now that this function exists, too, I found it to differ in output from `nsteddisc`. Jari Oksanen was quick to point out, that our two implementations differ in the way they handle ties in column totals. This function is, I think, closer to the results given in Brualdi & Sanderson. Jari also went on to implement different strategies to deal with ties, so my guess is that his version may be (slightly) superior to this one. Having said that, values don't differ much between the two implementations.

So what does it do: The matrix is sorted by marginal totals, yielding a matrix **A**. Then, all 1s in **A** are "pushed" to the left to maximally compact the matrix, yielding **P**. Discrepancy is now simply the number of disagreements between **A** and **P**, divided by two (to correct for the fact that every "wrong" 1 will necessarily generate a "wrong" 0).

### Value

Returns the number of mismatches, i.e. the discrepancy of the matrix from perfect nestedness.

**Note**

Discrepancy is well-defined only for matrices that can be sorted uniquely. For matrices with ties no foolproof way to handle them has been proposed. For small matrices, or large matrices with many ties, this will lead to different discrepancy values. See also how `nsteddisc` in **vegan** handles this issue! (Thanks to Jari Oksanen for pointing this out!)

**Author(s)**

Carsten F. Dormann

**References**

Brualdi, R.A. and Sanderson, J.G. (1999) Nested species subsets, gaps, and discrepancy. *Oecologia* **119**, 256–264

Ulrich, W. and Gotelli, N.J. (2007) Disentangling community patterns of nestedness and species co-occurrence. *Oikos* **116**, 2053–2061

**See Also**

`nestedness` for the most commonly used method to calculate nestedness, `wine` for a new, un-evaluated but very fast way to calculate nestedness; `nestedtemp` (another implementation of the same method used in our `nestedness`) and `nestedn0` (calculating the number of missing species, which has been shown to be a poor measure of nestedness) in **vegan**

**Examples**

```
data(Safariland)
require(vegan)
nulls <- replicate(1000, discrepancy(commsimulator(Safariland, method="quasiswap")))
hist(nulls)
obs <- discrepancy(Safariland)
abline(v=obs, lwd=3, col="grey")
c("p value"=min(sum(nulls>obs), sum(nulls<obs))/length(nulls))
# calculate Brualdi & Sanderson's Na-value (i.e. the z-score):
c("N_a"=(unname(obs)-mean(nulls))/sd(nulls))
```

---

elberling1999

*No. of visits in a pollination web of arctic-alpine Sweden.*

---

**Description**

This study took place in the subarctic alpine zone of Latnjajaure, in northern Sweden. Field work was conducted from May 21 to August 23, 1994. The objective was to describe the plant-flower visitor interaction matrix of this area and compare it with the characteristics of other subarctic alpine systems and with pollination systems of lower latitudes, especially in relation to the role of flies as flower visitors at high latitudes.

The authors recorded their data by counting the number of visits of each flower visitor species to each plant species. Regardless of whether insects were observed to forage for nectar and pollen or to perform sun-basking, they were all classified as flower visitors and potential pollinators and the plant species visited were recorded. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

See also <http://www.nceas.ucsb.edu/interactionweb/html/datasets.html>

### Usage

```
data(elberling1999)
```

### Format

A data frame with 12 plant species (in rows) and 102 pollinators (columns).

### References

Elberling H. and Olesen J.M. (1999) The structure of a high latitude plant-flower visitor system: the dominance of flies. *Ecography* **22**, 314–323

### Examples

```
data(barrett1987)
## maybe str(barrett1987) ; plot(barrett1987) ...
```

---

empty

*Deletes empty rows and columns from a matrix.*

---

### Description

Gets rid of empty columns and rows in a matrix. Optionally counts removed rows and columns, and returns these values as attribute.

### Usage

```
empty(web, count=FALSE)
```

### Arguments

web	A matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
count	Logical. Shall be counted how many columns and rows were removed? Numbers are returned in attribute. Defaults to FALSE.

## Details

Helper function to remove empty (i.e. all-zero or all-NA) rows and columns, thereby concentrating the matrix. This function is also invoked for its side effect by `extinction` to investigate the effect of removing a species from the network.

## Value

Returns matrix without empty rows or columns. Its attribute 'out' (if `count=TRUE`) contains a named vector with the number of rows removed and the number of columns removed.

## Author(s)

Carsten F. Dormann

## See Also

`extinction` and `second.extinct`, which repeatedly call `empty`.

## Examples

```
data(Safariland)
web <- Safariland
web[,5] <- 0
empty(web, count=TRUE)
attr(empty(web), "empty")
```

---

extinction

*Simulates extinction of a species from a bipartite network*

---

## Description

Following (how I remember) the paper of Memmott et al. (2004), this function deletes a column (e.g. pollinator) or row (e.g. plant). Only a helper function for `second.extinct`, really.

## Usage

```
extinction(web, participant = "both", method = "random")
```

## Arguments

<code>web</code>	A matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
<code>participant</code>	Which level of participant to remove: 'lower' removes a row, 'higher' removes a row, 'both' randomly picks either row or column. Partial matching of strings implemented.

method Determines sequence of extinctions: ‘random’ removes a random participant, while ‘abundance’ removes the least abundant species first. Partial matching of strings implemented.

### Details

In itself rather useless. Called repeatedly by `second.extinct` to build an extinction sequence and accordingly a sequence of secondary extinctions.

### Value

Returns the same matrix that was given as object, just with one row or column being turned into zeros.

### Author(s)

Carsten F. Dormann

### References

Memmott, J., Waser, N. M. and Price, M. V. 2004 Tolerance of pollination networks to species extinctions. *Proceedings of the Royal Society B* **271**, 2605–2611

### See Also

`second.extinct`

### Examples

```
data(Safariland)
(w <- extinction(Safariland, participant="low", method="abun"))
empty(w, count=TRUE)
```

---

functspec

*Calculates the functional specialisation index*

---

### Description

Calculates the functional specialisation index for all species in a bipartite network, separately for the higher and lower trophic level.

### Usage

```
functspec(web, inf.replace = NA)
```

**Arguments**

<code>web</code>	A matrix with lower trophic level species as rows, higher trophic level species as columns and number of interactions as entries.
<code>inf.replace</code>	What should infinite geodesic distances (e.g. between compartments) be represented as? Defaults to 'NA'; only currently implemented alternative is 'Inf', which replaces infinite distances by the maximum path length plus 1.

**Details**

This index aims to describe the functional specialisation of pollinators and was proposed by Dalgaard et al. (2008). It is a purely qualitative measure.

After calculating the geodesic distances between species, i.e. the minimum number of steps from one species to another, these values are averaged for each species. This mean geodesic distance is interpreted as functional specialisation (Dalgaard et al. 2008).

**Value**

A list with two components, names “higher” and “lower”, both containing the functional specialisation index for each species.

**Note**

This index is as yet unevaluated. We don't know how it responds to true specialisation at all. In fact, it is a rather good example of how to get a new thing published without even having demonstrated in which way it differs from existing indices of specialisation (such as standardised `d` included in the function `dfun`), or how it performs on artificial data with known properties.

One major disadvantage of any index based on path lengths is its difficulty with compartments, i.e. species not linked to the rest of the network. There are, generally speaking, three ways to handle this: Firstly, ignore it (that is, set infinite distances to NA; our default). Secondly, leave it as it is (that is, leave infinite distances as infinite). This is not really an option, since then ALL species would have infinite specialisation values. Thirdly, replace infinite by the largest distance plus one (see comments in `geodist` in `sna`). That would probably be a plausible thing to do, since we could argue that with a little bit extra observation we might have found a species linking a compartment to the rest of the network. However, this solution is “not canonical”, as put in `geodist` and hence biased to an unknown extent. To use this option, specify `inf.replace=Inf`.

**Author(s)**

Carsten F. Dormann <carsten.dormann@ufz.de>

**References**

Dalgaard, B., Martín González, A. M., Olesen, J. M., Timmermann, A., Andersen, L. H. and Ollerton, J. (2008) Pollination networks and functional specialization: a test using Lesser Antillean plant-hummingbird assemblages. *Oikos* **117**, 789–793

**See Also**

See also as [specieslevel](#), which calls `functspec`.

**Examples**

```
data(Safariland)
functspec(Safariland, inf.replace=Inf)
```

---

genweb	<i>Generate a random bipartite web</i>
--------	--

---

**Description**

Generates a random bipartite web, based on `r2dtable` and lognormal marginal distributions.

**Usage**

```
genweb(N1 = 10, N2 = 30, dens = 2)
```

**Arguments**

N1	Number of species in the lower trophic level; or a vector of length 2 giving number of lower and higher trophic level species. Defaults to 10.
N2	Number of species in the higher trophic level. Ignored if N1 a vector of length 2. Defaults to 30.
dens	Interaction density, i.e.~how many interactions there shall be, on average, for each species. Defaults to 2 (the median observed interaction density in the NCEAS pollination webs).

**Details**

This function can be used to create simple, but not necessarily realistic, bipartite webs for given dimensionality and interaction density. Marginal distributions are assumed to be lognormal, mean and standard deviation are calculated from ‘N1’, ‘N2’ and ‘dens’ (see code for details).

**Value**

A matrix with N1 x N2 species.

**Note**

Can be a bit time-consuming for large webs, because the absolute values for both dimensions have to match perfectly. This involves a rather unelegant `while`-loop.

**Author(s)**

Jochen Fründ and Carsten F. Dormann

**Examples**

```
genweb()
```

H2fun

*Specialisation of a bipartite web.***Description**

Calculates the overall level of specialisation of all partners in a bipartite web.

**Usage**

```
H2fun(web)
```

**Arguments**

web	A matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
-----	---

**Details**

H2' is an index describing the level of specialisation, or should one say: selectiveness, of an entire bipartite network (Blüthgen et al. 2006). It describes to which extent observed interactions deviate from those that would be expected given the species marginal sums. The more selective a species, the larger is H2' for the web.

H2' is an extension of d' (see [dfun](#)) for the entire network. Check there for details on finding maximum and minimum H2 values for a given web.

For internet implementation see also: <http://nils.mib.man.ac.uk/~nils/stat/>

**Value**

H2	The H2'-value for the web matrix.
H2min	Heuristic minimum H2-value for the web matrix.
H2max	Heuristic maximum H2-value for the web matrix.
H2uncorr	Uncorrected H2-values (before ranging between min and max), rounded to three digits.

**Author(s)**

Carsten F. Dormann and Jochen Fründ

**References**

Blüthgen, N., Menzel, F. and Blüthgen, N. (2006) Measuring specialization in species interaction networks. *BMC Ecology* **6**, 9

**See Also**

`dfun` doing the same for each species in the web matrix.

**Examples**

```
data(Safariland)
H2fun(Safariland)
```

---

inouye1988

*A pollination network.*

---

**Description**

This data set reports a community-level study of the pollination biology of alpine plants in Kosciusko National Park in the Snowy Mountains of south-eastern New South Wales, Australia. The flora and their associated insect pollinators were observed from December 1983 until March 1984.

**Usage**

```
data(inouye1988)
```

**Format**

A data frame with 41 observations on the following 83 variables, with plant species in rows and pollinators in columns.

**Details**

The authors recorded their data by counting the number of individual flower visitors caught on each plant species. The total number of individuals collected on each plant species provide a rough estimate of the level of visitation that each species received. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

**Note**

Male and female pollinators were summed when moving this data set from NCEAS to bipartite.

**Source**

NCEAS data base on interaction webs: <http://www.nceas.ucsb.edu/interactionweb/html/datasets.html>

**References**

Inouye, D.W. and G.H. Pyke (1988) Pollination biology in the Snowy Mountains of Australia: comparisons with montane Colorado, USA. *Australian Journal of Ecology* **13**: 191–210.

## Examples

```
data(inouye1988)
plotweb(inouye1988)
```

---

kato1990	<i>No. of individuals caught in a pollination web of a Japanese beech forest.</i>
----------	---

---

## Description

The study took place at the Kyoto University Forest of Ashu, at the northeastern boundary of the Kyoto Prefecture in Japan, between 1984 and 1987. The paper deals with the flowering phenology of 91 plant species, the community structure of flower-visiting insects, and the spectrum of floral hosts for flower visitors. The emphasis is laid on the pattern of community organization of flower-visiting insects in a primary forest ecosystem of western Japan.

The authors recorded their data by counting the number of individual flower visitors caught on each plant species. The total number of individuals collected on each plant species provide a rough estimate of the level of visitation that each species received. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction. For details and data see <http://www.nceas.ucsb.edu/interactionweb/html/datasets.html>

## Usage

```
data(kato1990)
```

## References

Kato, M., T. Makutani, T. Inoue, and T. Itino. 1990. Insect-flower relationship in the primary beech forest of Ashu, Kyoto: an overview of the flowering phenology and seasonal pattern of insect visits. *Contr. Biol. Lab. Kyoto Univ.* 27:309-375.

## Examples

```
data(kato1990)
## maybe str(kato1990) ; plot(kato1990) ...
```

---

`kevan1970`*A pollination network.*

---

## Description

The total number of individuals collected on each plant species provide a rough estimate of the level of visitation that each species received.

## Usage

```
data(kevan1970)
```

## Details

### General information

This study sought to determine the importance of insect-flower relations to both plants and insects in a high arctic community as well as the degree to which some of the more common arctic plants are dependent on insects for pollination and reproduction. The research was conducted in 1967 at Hazen Camp (81° 49' N, 71° 18' W) near Lake Hazen on Northern Ellesmere Island, the most northerly island of the Canadian Arctic Archipelago.

### Data type

The authors recorded their data by counting the number of individual flower visitors caught on each plant species. The total number of individuals collected on each plant species provide a rough estimate of the level of visitation that each species received. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

## References

Kevan, P. G. 1970. High Arctic Insect-Flower Visitor Relations: The Inter-Relationships of Arthropods and Flowers at Lake Hazen, Ellesmere Island, Northwest Territories. University of Alberta, Canada.

## Examples

```
data(kevan1970)
```

---

`memmott1999`*Pollination network.*

---

**Description**

This study was conducted in a 150 x 250 m meadow plot in the vicinity of Bristol, U.K. in July 1997. The objective was to describe the plant-flower visitor interaction web of this area, taking into account species abundances and their frequency of interaction. Twenty five plant species were studied, and 79 flower visitor species were recorded visiting them.

**Usage**

```
data(memmott1999)
```

**Details**

The author recorded her data by counting the number of visits of each flower visitor species to each plant species, and by independently measuring the abundance of plant and animal taxa. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

**Source**

NCEAS

**References**

Memmott, J. 1999. The structure of a plant-pollinator food web. *Ecology Letters* 2:276-280.

**Examples**

```
data(memmott1999)
## maybe str(memmott1999) ; plot(memmott1999) ...
```

---

`mosquin1967`*Pollination network.*

---

**Description**

This study took place on Melville Island, N.W.T., Canada from July 19 to July 31 1965. While collecting plants the authors made some observations on the occurrence and behavior of flower visiting insects as well as on the scent and other target characteristics of flowers.

**Usage**

```
data(mosquin1967)
```

**Details**

The authors recorded their data by counting the number of individual flower visitors caught on each plant species. The total number of individuals collected on each plant species provide a rough estimate of the level of visitation that each species received. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

**Source**

NCEAS

**References**

Mosquin, T., and J. E. H. Martin. 1967. Observations on the pollination biology of plants on Melville Island, N.W.T., Canada. *Canadian Field Naturalist* 81:201-205.

**Examples**

```
data(mosquin1967)
## maybe str(mosquin1967) ; plot(mosquin1967) ...
```

---

motten1982

*A pollination network.*

---

**Description**

This is a study of the interactions between insects visitors and spring wildflowers in piedmont North Carolina. Spring flowering, entomophilous herbs, shrubs, and understory trees were included in the study.

**Usage**

```
data(motten1982)
```

**Details**

The author recorded his data by counting the number of visits of each flower visitor species to each plant species. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

**References**

Motten, A. F. 1982. *Pollination Ecology of the Spring Wildflower Community in the Deciduous Forests of Piedmont North Carolina*. Doctoral Dissertation thesis, Duke University, Durham, North Carolina, USA.

Motten, A. F. 1986. Pollination ecology of the spring wildflower community of a temperate deciduous forest. *Ecological Monographs* 56:21-42.

**Examples**

```
data(motten1982)
## maybe str(motten1982) ; plot(motten1982) ...
```

---

nested	<i>Calculates any of several measures of nestedness</i>
--------	---

---

**Description**

Wrapper function calling one, several or all currently implemented nestedness measures

**Usage**

```
nested(web, method = "binmatnest2", ..., rescale = FALSE)
```

**Arguments**

web	A matrix with elements of a set (e.g., plants) as rows, elements of a second set (e.g., pollinators) as columns and number of interactions as entries.
method	One or more of the following: discrepancy, discrepancy2, binmatnest, binmatnest2, NODF, NODF2, C.score, checker, wine, ALL. See details for details on each method
...	Arguments passed on to other nestedness functions. Options need to be specified (i.e. no positional parsing).
rescale	Should all measures be rescaled so that higher values mean higher nestedness? Defaults to FALSE, i.e. the standard interpretation of each measure is maintained.

**Details**

There are five different measures currently available:

1. binmatnest calculates nestedness temperature following the function `nestedness` (0 = cold = highly nested; 100 = hot = not nested at all). It uses the original program of Miguel Rodríguez-Gironés, only called from R; binmatnest2, in contrast, is the implementation in `nestedtemp` of the same algorithm by Jari Oksanen. Because binmatnest sometimes (and to us unexplicably) invert the matrix, we prefer the binmatnest2 option.
2. Discrepancy calculates the number of non-nested 0s and 1s. While `discrepancy` calls the function with the same name, `discrepancy2` calls `nesteddisc`, which handles ties differently. Most of the time, these two should deliver very, very similar results. Higher values indicate lower nestedness.
3. NODF is the nestedness measure proposed by Almeida-Neto et al., correcting for matrix fill and matrix dimensions. Values of 0 indicate non-nestedness, those of 100 perfect nesting. NODF2 sorts the matrix before calculating the measure. NODF is, I understand, closer to the version presented in the paper, while NODF2 seems to make more sense for comparisons across different networks (because it is independent of the initial presentation of the matrix).

Both call `nestednodf` in **vegan**. (Yes, I initially programmed NODF myself, only to find that it was there already. Luckily, there was a perfect agreement between my (depricated) version and `nestednodf`.)

4. `C.score` calculates the number of checkerboard pattern in the matrix. As default, it normalises this value between min and max, so that values of 0 indicate no checkerboards (i.e. nesting), while a value of 1 indicates a perfect checkerboard. `checker` is the non-normalised version, based on `nestedchecker`.
5. `wine` is the only nestedness measure using the information on the weight of a link. See `wine` for details.

### Value

A vector with values for each of the selected nestedness measures.

### Note

The idea behind this function is to encourage the comparison of different nestedness measures. That does not mean, we necessarily see much ecological sense in them (see, e.g., the paper by Blüthgen et al. 2008).

### Author(s)

Carsten F. Dormann <carsten.dormann@ufz.de>

### References

- Almeida-Neto, M., Gumaraes, P., Gumaraes, P.R., Loyola, R.D. and Ulrich, W. 2008. A consistent metric for nestedness analysis in ecological systems: reconciling concept and measurement. *Oikos* **117**, 1227–1239.
- Blüthgen, N., J. Fründ, D. P. Vázquez, and F. Menzel. 2008. What do interaction network metrics tell us about specialisation and biological traits? *Ecology* **89**, 3387–3399.
- Brualdi, R.A. and Sanderson, J.G. 1999. Nested species subsets, gaps, and discrepancy. *Oecologia* **119**, 256–264.
- Galeano, J., Pastor, J.M., Iriondo and J.M. 2008. Weighted-Interaction Nestedness Estimator (WINE): A new estimator to calculate over frequency matrices. *arXiv* 0808.3397v2 [physics.bio-ph]
- Rodriguez-Girones, M.A. and Santamaria, L. 2006. A new algorithm to calculate the nestedness temperature of presence-absence matrices. *J. Biogeogr.* **33**, 924–935.
- Stone, L. and Roberts, A. 1990. The checkerboard score and species distributions. *Oecologia* **85**, 74–79.

### See Also

`C.score`, `wine`, `nestedness`, `discrepancy`; and, within **vegan**: `nestedtemp`, `nestedchecker`, `nesteddisc`, `nestednodf`

**Examples**

```

data(Safariland)
nested(Safariland, "ALL")
nested(Safariland, "ALL", rescale=TRUE)
# illustration that non-normalised C.score and checker are the same:
nested(Safariland, c("C.score", "checker"), normalise=FALSE)

```

---

 nestedness

*Calculates nestedness temperature of presence/absence matrices*


---

**Description**

Calculates matrix temperature using the binmatnest programm of Miguel Rodríguez-Gironés by calling a tweaked version of the C++ program binmatnest. For a full description what it does please refer to the paper of Miguel. In principle nestedness temperature is calculated by using a line of perfect order (using a genetic algorithm) to determine the reordering of rows and columns that leads to minimum matrix temperature of given size and fills. The deviation from this minimum temperature is the matrix temperature. In addition nestedness uses there different null models to check for statistical significance of the matrix temperature.

**Usage**

```

nestedness(m, null.models = TRUE, n.nulls = 100, popsize = 30, n.ind = 7,
n.gen = 2000, binmatnestout=FALSE)

```

**Arguments**

<code>m</code>	<code>m</code> is the matrix object for which the temperature is calculated. <code>m</code> will be converted to a binary matrix as temperature is only based on binary data
<code>null.models</code>	Logical; shall the three different null models to check for significance of the matrix temperature be calculated? The null models procedure is quite time consuming and therefore we added this switch. Defaults to <code>null.models=TRUE</code> .
<code>n.nulls</code>	How many null models should be calculated. Defaults to <code>n.nulls=100</code> .
<code>popsize</code>	For the genetic algorithm some parameters have to be initialised. First is <code>popsize</code> , default is 30
<code>n.ind</code>	Second is number of individuals picked for the next generation. Default of <code>n.ind</code> is 7.
<code>n.gen</code>	Third is the number of generations until the genetic algorithm stops. Default of <code>n.gen</code> is 2000.
<code>binmatnestout</code>	if set to <code>TRUE</code> a file "binmat.out" is saved in the current working directory , which stores the original binmatnest output

## Details

There are several implementations of nestedness-calculators, most noticeably NTC (nestedness temperature calculator), BINMATNEST and aninhado (check Wikipedia's entry on the subject: <http://en.wikipedia.org/wiki/Nestedness>). While we here use BINMATNEST, this does not disqualify any of the others. Miguel was simply the first we contacted and he was readily willing to share his code (applause).

For details on what BINMATNEST does different, and better, than the original NTC see reference below.

Notice also that the original software BINMATNEST is available as a stand-alone application, too. Check out Miguel's homepage: <http://www.eeza.csic.es/eeza/personales/rgirones.aspx> or download directly: <http://www.eeza.csic.es/eeza/personales/rgirones/File/BINMATNEST3.zip>.

## Value

returns a list of matrix descriptors, such as

```
temperature  the matrix temperature
parameters of genetic algorithms
                Parameters used for the genetic algorithm
nullmodels   switch if null models have been calculated, 1 for yes, 0 for no
p, mean, var  probability, mean temperature and variance of temperature for the three different
                null models
packing order
                the packing order of the most packed matrix (minimum temperature of a perfectly
                nested matrix using given size and fills.
```

## Note

Make sure matrix *m* is valid, as error proofing in the C++ function does not fully work, and therefore it is possible that R may crash when using strange types of matrices, such as matrices with only one entry.

Also, while this function returns exactly the same values as `binmatnest`, there seems to be the occasional glitch in the software, in that in unexplicable circumstances the matrix is inverted before calculating nestedness. We are currently trying to find the cause of this problem. Sorry!

In the meantime, use `nestedtemp` in **vegan** (as we do in `networklevel`)!

## Author(s)

Bernd Gruber, based on C++-code by Miguel Rodríguez-Gironés.

## References

Rodríguez-Gironés M.A., and Santamaría L. 2006. A new algorithm to calculate the nestedness temperature of presence-absence matrices. *Journal of Biogeography* **33**, 924–935

## Examples

```
data(vazarr)
nestedness(vazarr) # null models are calculated
nestedness(vazarr, null.models=FALSE) # no null models, much faster for bigger matrices
nestedness(vazarr, n.nulls=300, n.gen=300)
```

---

networklevel

*Analysis of bipartite webs at the level of the entire network*

---

## Description

Calculates a variety of indices and values for a bipartite network.

## Usage

```
networklevel(web, index="ALL", ISAmethod="Bluethgen",
             SAmethod = "Bluethgen", extinctmethod = "r", nrep = 100,
             plot.it.extinction = FALSE, plot.it.dd=FALSE, CCfun=median,
             dist="horn", normalise=TRUE, empty.web=TRUE, logbase="e",
             intereven="sum")
```

## Arguments

web	Web is a matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
index	One or more of the following (exact match only!): “number of species”, “connectance”, “web asymmetry”, “links per species”, “number of compartments”, “compartment diversity”, “cluster coefficient”, “degree distribution”, “mean number of shared partners”, “togetherness”, “C score”, “V ratio”, “discrepancy”, “nestedness”, “weighted nestedness”, “extinction slope”, “robustness”, “ISA”, “SA”, “niche overlap”, “generality”, “vulnerability”, “linkage density”, “Fisher alpha”, “mean interaction diversity”, “interaction evenness”, “Alatalo interaction evenness”, “diversity”, “H2”; “ALL” (default) calculates all indices (returning a list (since degree distribution fits are a table within this list and cannot be simplified)), “ALLBUTDD” calculates all indices except degree distribution fits. This latter has the advantage that the output can be simplified to a vector; “info” returns more general information on the network, “binary” returns a best-of selection of indices based on a binary network, “quantitative” returns a best-of selection of indices based on quantitative networks, “topology” returns indices more abstractly describing network properties. Also CHECK details below!
ISAmethod	Method to use for calculating interaction strength (= dependence) asymmetry; original by “Bascompte” is biased for singletons and few interactions (range 0 to infy); “Bluethgen” (default) excludes singletons and corrects for low number of interactions (range -1 to 1); see Bascompte et al. (2006) and Blüthgen et al. (2007).

<code>SAmethod</code>	Which method to use to calculate the specification asymmetry: mean of log-transformed dependencies (“log”) or Blüthgen’s abundance-weighted mean (default); see Blüthgen et al. (2007).
<code>extinctmethod</code>	Specifies how species are removed from matrix: “random” or “abundance” (partial matching), where abundance removes species in the order of increasing abundance (i.e. rarest first); see Memmott et al. (2004).
<code>nrep</code>	Number of replicates for the extinction sequence analysis.
<code>plot.it.extinction</code>	logical; plot the extinction sequence graph; defaults to FALSE.
<code>plot.it.dd</code>	logical; plot the degree distribution fits?; defaults to FALSE.
<code>CCfun</code>	Method to use when calculating the clustering coefficient. Originally proposed as mean of cluster coefficients for each node. Defaults to “median”, because cluster coefficients are strongly skewed.
<code>dist</code>	Distance metric to be used to calculate niche overlap; defaults to Horn’s index, which is the recommendation of Krebs (Ecological Methodology); for other options see <code>vegdist</code> in <b>vegan</b> .
<code>normalise</code>	Logical; shall the C-score and togetherness metrics be normalised to a range of 0 to 1? Defaults to TRUE.
<code>empty.web</code>	Shall the empty columns and rows be deleted? Defaults to TRUE.
<code>logbase</code>	Shall the various diversity indices be calculated to the base of e (default) or 2? Log2 is the proposal for generality and vulnerability by Bersier et al. (2002), while Shannon uses ln. The choice of the base will not affect the results qualitatively, only by a scaling factor.
<code>intereven</code>	Shall all cells of the matrix be used to calculate the interaction evenness ( <code>intereven="prod"</code> )? Or, as given by Bersier et al. (2002) and Tylianakis et al. (2007), should only the realised links be used ( <code>intereven="sum"</code> ; default)? Prod and sum refer to using the log of the product of matrix dimensions (i.e. all cells) or the log of the sum of non-zero cells (i.e. number of links) as denominator in the evenness formula. See last paragraph of the details-section for views on these two options!

## Details

This function implements a variety of the many (and still procreating) indices describing network topography. Some are embaracingly simple and mere descriptors of a network’s outer appearance (such as number of species in each trophic level or the number of links (= non-zero cells) in the web). Others are variations on Shannon’s diversity index applied to within column or within rows. Only extinction slope is newly implemented here, and hence described in a bit more detail.

Note that Bersier et al. (2002) have three levels of values for some of their indices: qualitative (i.e. based on binary networks), quantitative (based on networks with information on the number of interactions observed for each link), and weighted-quantitative (where also independent information on the abundances of the participants is fed into the formula). At present, we implement only qualitative and quantitative indices! One exception is `d'`, which can be called with an abundance vector (in function `dfun`). You can also get the qualitative version of quantitative indices (such as vulnerability) by simply calling `networklevel` on “binarised” data: `networklevel(Safariland>0,`

`index="vulnerability")`. (Why you would want to do that, however, is currently beyond me.)(Thanks to Jason Tylianakis for proposing to put this clarification into the help!)

Extinction slope works on a repeated random sequence of species extinctions (within one trophic level), and calculates the number of secondary extinctions (in the other level). These values are then averaged (over the `nrep` runs) and plotted against the number of species exterminated. The proportion still recent (on the y-axis) regressed against the proportion exterminated (on the x-axis) is hence standardised to values between 0 and 1 each. Through this plot, a hyperbolic regression is fitted, and the slope of this regression line is returned as an index of extinction sensitivity. The larger the slope, the later the extinction takes its toll on the other trophic level, and hence the higher the redundancy in the trophic level under consideration. Using `plot.it=F` also returns the graphs (set history to recording in the plotting window). Changing the `extinctionmethod` to "abundance" will always result in the same sequence (by increasing abundance) and hence does not require replication.

Most indices are straightforward, one-line formulae; some, such as H2', also require a re-arranging of the matrix; and one, secondary extinction slope, internally requires iterative runs, making the function relatively slow. If you are not interested in the secondary extinction slopes, simply set `nrep=1` to make it much faster.

Sum or Prod: How to calculate interaction evenness? I shall first put down my argument for "prod" and then Jason Tylianakis' arguments for "sum". Carsten: "I do not want to defend a position I cannot hold against the flood of qualified criticism, and shall be happy to change the default to option "sum" (i.e. Jason's proposal). Nevertheless, I shall make a very brief attempt to defend my (and Nico's point of view). Imagine a completely different situation: I have "counted" birds in a landscape. From a more meticulous colleague I know that there are 27 bird species breeding at the moment, but on that two morning that I went out, I could only hear 15. Now I want to calculate the Shannon diversity (and evenness) of birds in that landscape. The "normal" (in the sense of established) approach to use the data from my 15 species. But hold on: I KNOW there are more species out there. I don't know how many (i.e. there may be more than the 27 my colleague has found), but there are at least 27. If I only use the data from my 15 species, I will get a higher evenness value than when I also include the 12 zeros. My conclusion would be: I don't want to overestimate evenness only because I couldn't look long enough, thus I use all 27 values." Jason: "I would disagree because what you "know" is based on your meticulous colleague's 'sampling', which will also have its limits. If all you wanted was to know the total number of species there (assuming none have gone extinct), then what you propose is fine. However, the problem comes when you want to compare sites, and then sampling effort should be standardised. In most cases we know we don't have a full representation of the diversity (or food web) of an area, but we know for a given spatial or temporal sampling scale that one site differs from another in certain ways, and to me that is the most important. Anyway, it is all a question of scale and the precise question being asked. So what about making it an option in bipartite that you can either choose to divide by the realised links (give our 2007 paper as a ref, so people know it's comparable to that) or divide by the number of potential links, if that's the question people want to ask?" There you go: it's your choice!

## Value

Depending on the selected indices, some or all of the below (returned as vector is "degree distribution" was not requested, otherwise as list):

number of higher trophic species

Number of species in the higher trophic level, i.e. `ncol(web)`.

number of lower trophic species	Number of species in the lower trophic level, i.e. <code>nrow(web)</code> .
connectance	Realised proportion of possible links (Dunne et al. 2002): sum of links divided by number of cells in the matrix (= number of higher times number of lower trophic level species). This is the standardised number of species combinations often used in co-occurrence analyses (Gotelli & Graves 1996)
web asymmetry	Balance between numbers in the two levels: positive numbers indicate more lower-trophic level species, negative more higher-trophic level species; implemented as $(nrow(web) - ncol(web)) / \text{sum}(\text{dim}(web))$ ; web asymmetry is a null-model for what one might expect in dependence asymmetry: see Blüthgen et al. (2007).
links per species	Mean number of links per species (qualitative): sum of links divided by number of species.
number of compartments	Compartments are sub-sets of the web which are not connected (through either higher or lower trophic level) to another compartment. Mathematically, they are Jordan blocks, but this implementation is rule-based (and fast). They are also nicely visualised in the <code>visweb</code> function.
compartment diversity C.D.	Shannon's diversity index across compartment sizes (i.e. number of participants); see Tylianakis et al. (2007).
cluster coefficient	The CC for a network is the average CC of its members. CC for each node, in turn, is simply the number of realised links divided by the number of possible links. Introduced by Watts & Strogatz (1998) and described in Wikipedia under <a href="http://en.wikipedia.org/w/index.php?title=Clustering_coefficient">http://en.wikipedia.org/w/index.php?title=Clustering_coefficient</a> .
degree distribution	Coefficients and fits for three different functions to degree distributions: exponential, power law and truncated power law. See <code>degreedistr</code> for details and references.
mean number of shared hosts	The simplest measure of co-occurrence and hence similarity in host preferences of the higher trophic level; based on Roberts & Stone (1990) and Stone & Roberts (1992).
mean number of shared predators	Same as before, only for the lower trophic level.
togetherness	Mean number of co-occupancies across all species-host-combinations; the whole matrix is scanned for submatrices of the form <code>c(0,0,1,1)</code> , representing perfect matches of co-presences and co-absences. These are counted for each pairwise species combination, and averaged. Based on Stone & Roberts (1992).
C score	Mean (normalised) number of checkerboard combinations across all higher trophic level species. Values close to 1 indicate that there is evidence for disaggregation, e.g.~through competition. Value close to 0 indicate aggregation of species (i.e.~no repelling forces between species). See Stone and Roberts (1990) for details.

V ratio	Variance-ratio of species numbers to individual numbers within species for the higher trophic level. Values larger than 1 indicate positive aggregation, values between 0 and 1 indicate disaggregation of species. See Schluter (1984) for details.
discrepancy	Discrepancy as proposed by Brualdi & Sanderson (1999); see [ <a href="#">discrepancy</a> for details.
nestedness	Nestedness temperature of the matrix (0 means cold, i.e.~high nestedness, 100 means hot, i.e.~chaos). For details see <a href="#">nestedness</a> and Rodríguez-Gironés & Santamaria (2002). Notice that the function <a href="#">nestedness</a> does not calculate any null model, simply because it is too computer-intensive. <code>networklevel</code> calls <code>nestedtemp</code> ! If you are interested in the different null models, please use the function <a href="#">nestedness</a> or <code>nestedtemp</code> in <b>vegan</b> directly.
weighted nestedness	A nestedness version that considers interaction frequencies (and is hence weighted), proposed by Galeano et al. (2007) and implemented in <a href="#">wine</a> . It ranges between 1 (perfect nestedness) and 0 (perfect chaos). Note that this is the OPPOSITE interpretation of nestedness temperature!
dependence asymmetry/interaction strength asymmetry	Explaining dependence asymmetry will require more space than just a few lines. In essence, it is also a measure of specialisation, across both trophic levels. Proposed by Bascompte et al. (2006) and critiqued and alterations proposed by Blüthgen et al. (2007). The latter also show that dependence asymmetry can be almost entirely explained by web asymmetry (see above). Positive values (only possible of ‘ <code>DAmethod="Bluethgen"</code> ’) indicate higher dependence in the higher trophic level.
specialisation asymmetry	Asymmetry (higher vs. lower trophic level) of specialisation now based on $d'$ (see <a href="#">dfun</a> ), which is insensitive to the dimensions of the web. Again, two options of calculation are available: the one proposed by Blüthgen et al. (2007), where they weight the specialisation value for each species by its abundance (‘ <code>SAmethod="Bluethgen"</code> ’) or where $d'$ -values are log-transformed (arguing that $d'$ -values are indeed log-normally distributed: ‘ <code>SAmethod="log"</code> ’). Since the mean $d'$ -value for the lower trophic level is subtracted from that of the higher, positive values indicate a higher specialisation of the higher trophic level.
extinction slope higher trophic level	Slope of the secondary extinction sequence in the higher trophic level, following extermination of species in the lower trophic level; see details.
extinction slope lower trophic level	Slope of the secondary extinction sequence in the lower trophic level, following extermination of species in the higher trophic level; see details.
robustness	Calculates the area below the extinction curve; see <a href="#">robustness</a> for details.
niche overlap	Mean similarity in interaction pattern between species of the same trophic level, calculated by default as Horn’s index. Values near 0 indicate no common use of niches, 1 indicates perfect niche overlap. (In how far it makes sense for hosts of predators to call their commonness in enemies “niche overlap” is a different

issue. There are people calling predators “negative resources” (couldn’t be asked to look up the reference). I would probably rather call it similarity in predators.)

- generality      Effective mean number of prey per predator; see Tylianakis et al. (2007).
- vulnerability      Effective mean number of predator per prey; see Tylianakis et al. (2007).
- linkage density      Mean number of interactions per species (quantitative); see Tylianakis et al. (2007).
- Fisher’s alpha      An alternative measure of interaction diversity (using `fisherfit` from **vegan**).
- HTL mean interaction diversity      Mean diversity of prey per predator. HTL refers to Higher Trophic Level. HTLmid is analog to Shannon’s diversity index for interactions (see below), but calculated for each species in the HTL, then averaged.
- LTL mean interaction diversity      Mean diversity of predators per prey. LTL refers to Lower Trophic Level. LTLmid is analog to Shannon’s diversity index for interactions (see below), but calculated for each species in the LTL, then averaged.
- interaction evenness      Shannon’s evenness for the web entries, treating zeros as no data. Note that the two options are rather different. Here a brief attempt to justify our default “`prod`”: By definition,  $IE = H/H_{max}$ ;  $H = -\sum(p.i.mat \cdot \log(p.i.mat))$ , where  $p.i.mat = matrix / \sum(\text{entries in matrix})$ . This means, when calculating  $H$ , we treat the matrix cells (=links) as species, and the interactions (cell values) as measure of their abundance. By definition,  $H_{max} = \ln(N)$ . The key question is: What is the right value for  $N$ ? Since we treat the matrix cells as species, it is (clearly?) the number of matrix cells, i.e. number of higher trophic level species  $\times$  number of lower trophic level species. What else? Were we to use the interpretation of Tylianakis et al. (2007), then  $H_{max} = \ln(\text{sum of entries in matrix})$ . This means, we equate “number of interactions” (another phrase for sum of matrix entries) = “number of species”. That means, each interaction is a species. What should that mean? Why should that measure “interactions evenness”? Why should we move from a view of “cells are species” when calculating  $H$  to a view of “interactions are species” when calculating  $N$ ? To say the least, it doesn’t seem consistent.
- Alatalo interaction evenness      A different measure for web entry evenness, as proposed by Müller et al. (1999).
- Shannon diversity      Shannon’s diversity of interactions (i.e. network entries).
- H2      H2’ is a network-level measure of specialisation. It ranges between 0 (no specialisation) and 1 (complete specialisation). H2’ is a measure of discrimination, i.e. calculated in comparison of no specialisation (see [H2fun](#) for details. To avoid confusion of keys (apostrophe vs. accent), we call the H2’ only H2 here.

**Note**

Some indices require rather long computation times on large webs. Particularly slow (for large webs) are: “extinction slope” and “nestedness”. For ordinary-sized webs (tens by tens), `networklevel` is reasonably fast. An example of a large web is `kato1990`: try it, possibly using `system.time` as a wrapper, and prepare to do some reading on the side ...

If you want to increase the speed by omitting some indices, here a rough guide: Ask only for the indices you are interested in! Otherwise, here is the sequence of most time-consuming indices: 1. The slowest function is related to extinction slopes and robustness. Excluding both makes the function faster. 2. Degree distributions are somewhat time consuming. 3. Fisher’s alpha is computed iteratively and hence time consuming. 3. Nestedness and weighted nestedness are not the fastest of routines. 4. Number (and diversity) of compartments calls a recursive and hence relatively slow algorithm. 5. H2 and specialisation asymmetry require an iterative, heuristic search algorithm. Finally, excluding discrepancy can also moderately decrease computation time.

**Author(s)**

Carsten F. Dormann <carsten.dormann@ufz.de>

**References**

- Bascompte, J., Jordano, P. and Olesen, J. M. 2006 Asymmetric coevolutionary networks facilitate biodiversity maintenance. *Science* **312**, 431–433
- Bersier, L. F., Banasek-Richter, C. and Cattin, M. F. (2002) Quantitative descriptors of food-web matrices. *Ecology* **83**, 2394–2407
- Blüthgen, N., Menzel, F., Hovestadt, T., Fiala, B. and Blüthgen N. 2007 Specialization, constraints and conflicting interests in mutualistic networks. *Current Biology* **17**, 1–6
- Burgos, E., H. Ceva, R.P.J. Perazzo, M. Devoto, D. Medan, M. Zimmermann, and A. Maria Delbue (2007) Why nestedness in mutualistic networks? *Journal of Theoretical Biology* **249**, 307–313
- Corso G, de Araújo AIL, de Almeida AM (2008) A new nestedness estimator in community networks. *arXiv* 0803.0007v1 [physics.bio-ph]
- Dormann, C.F., Fründ, J., Blüthgen, N., and Gruber, B. (2009) Indices, graphs and null models: analysing bipartite ecological networks. *The Open Ecology Journal* **2**, 7–24.
- Dunne, J. A., R. J. Williams, and N. D. Martinez. 2002 Food-web structure and network theory: the role of connectance and size. *Proceedings of the National Academy of Science USA* **99**, 12917–12922
- Galeano J, Pastor JM, Iriando JM (2008) Weighted-Interaction Nestedness Estimator (WINE): A new estimator to calculate over frequency matrices. *arXiv* 0808.3397v1 [physics.bio-ph]
- Gotelli, N. J., and G. R. Graves. 1996 Null Models in Ecology. Smithsonian Institution Press, Washington D.C.
- Memmott, J., Waser, N. M. and Price M. V. 2004 Tolerance of pollination networks to species extinctions. *Proceedings of the Royal Society B* **271**, 2605–2611
- Müller, C. B., Adriaanse, I. C. T., Belshaw, R. and Godfray, H. C. J. 1999 The structure of an aphid-parasitoid community. *Journal of Animal Ecology* **68**, 346–370
- Roberts, A. and Stone, L. 1990 Island-sharing by archipelago species. *Oecologia* **83**, 560–567

- Rodríguez-Gironés M.A., and Santamaría L. 2006. A new algorithm to calculate the nestedness temperature of presence-absence matrices. *Journal of Biogeography* **33**, 924–935
- Schluter, D. 1984 A variance test for detecting species associations, with some example applications. *Ecology* **65**, 998-1005.
- Stone, L. and Roberts, A. 1990 The checkerboard score and species distributions. *Oecologia* **85**, 74–79.
- Stone, L. and Roberts, A. 1992 Competitive exclusion, or species aggregation? An aid in deciding. *Oecologia* **91**, 419–424
- Tylianakis, J. M., Tschardt, T. and Lewis, O. T. 2007 Habitat modification alters the structure of tropical host-parasitoid food webs. *Nature* **445**, 202–205
- Watts, D. J. and Strogatz, S. 1998 Collective dynamics of ‘small-world’ networks. *Nature* **393**, 440–442

### See Also

Some functions are implemented separately: [H2fun](#), [second.extinct](#), [degreedistr](#), [C.score](#) and [V.ratio](#)

### Examples

```
data(Safariland)
networklevel(Safariland)
networklevel(Safariland, index="ALLBUTDD") #excludes degree distribution fits
```

---

nodespec

*Calculates the node-based specialisation index*

---

### Description

Calculates a specialisation index based on the node positions for all species in a bipartite network, separately for the higher and lower trophic level.

### Usage

```
nodespec(web, inf.replace = NA)
```

### Arguments

- |                          |  |
|--------------------------|--|
| <code>web</code>         | A matrix with lower trophic level species as rows, higher trophic level species as columns and number of interactions as entries.  |
| <code>inf.replace</code> | What should infinite geodesic distances (e.g. between compartments) be represented as? Defaults to ‘NA’; only currently implemented alternative is ‘Inf’, which replaces infinite distances by the maximum path length plus 1. |

## Details

This index aims to describe the functional specialisation of pollinators and was proposed by Dalgaard et al. (2008). It is a purely qualitative measure.

After calculating the geodesic distances between species, i.e. the minimum number of steps from one species to another, these values are averaged for each species. This mean geodesic distance is interpreted as functional specialisation (Dalgaard et al. 2008).

## Value

A list with two components, names “higher” and “lower”, both containing the node specialisation index for each species.

## Note

This index is as yet unevaluated. We don’t know how it responds to true specialisation at all. In fact, it is a rather good example of how to get a new thing published without even having demonstrated in which way it differs from existing indices of specialisation (such as standardised `d` included in the function `dfun`), or how it performs on artificial data with known properties.

One major disadvantage of any index based on path lengths is its difficulty with compartments, i.e. species not linked to the rest of the network. There are, generally speaking, three ways to handle this: Firstly, ignore it (that is, set infinite distances to `NA`; our default). Secondly, leave it as it is (that is, leave infinite distances as infinite). This is not really an option, since then ALL species would have infinite specialisation values. Thirdly, replace infinite by the largest distance plus one (see comments in `geodist` in `sna`). That would probably be a plausible thing to do, since we could argue that with a little bit extra observation we might have found a species linking a compartment to the rest of the network. However, this solution is “not canonical”, as put in `geodist` and hence biased to an unknown extent. To use this option, specify `inf.replace=Inf`.

## Author(s)

Carsten F. Dormann <carsten.dormann@ufz.de>

## References

Dalgaard, B., Martín González, A. M., Olesen, J. M., Timmermann, A., Andersen, L. H. and Ollerton, J. (2008) Pollination networks and functional specialization: a test using Lesser Antillean plant-hummingbird assemblages. *Oikos* **117**, 789–793

## See Also

See also as [specieslevel](#), which calls `nodespec`.

## Examples

```
data(Safariland)
nodespec(Safariland, inf.replace=Inf)
```

---

null.t.test	<i>Compares observed pattern to random webs.</i>
-------------	--

---

### Description

A little null-model function to check, if the observed values actually are much different to what one would expect under random numbers given the observed row and column totals (i.e.~information in the structure of the web, not only in its species' abundances). Random matrices are based on the function [r2dtable](#). The test itself is a t-test (with all its assumptions).

### Usage

```
null.t.test(web, N = 30, ...)
```

### Arguments

web	A matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows).
N	Number of null models to be produced; see 'Note' below!
...	Optional parameters to be passed on to the functions <a href="#">networklevel</a> and <a href="#">t.test</a> .

### Details

This is only a very rough null-model test. There are various reasons why one may consider [r2dtable](#) as an incorrect way to construct null models (e.g.~because it yields very different connectance values compared to the original). It is merely used here to indicate into which direction a proper development of null models may start off. Also, if the distribution of null models is very skewed, a t-test is obviously not the test of choice.

Finally, not all indices will be reasonably testable (e.g.~number of species is fixed), or are returned by the function [networklevel](#) in a form that `null.t.test` can make use of (e.g.~degree distribution fits).

### Value

Returns a table with one row per index, and columns giving

obs	observed value
null mean	mean null model value
lower CI	lower 95% confidence interval (or whatever level is specified in the function's call)
upper CI	upper 95% confidence interval (or whatever level is specified in the function's call)
t	t-statistic
P	P-value of t statistic

**Note**

This function is rather slow. Using large replications in combination with iterative indices (degree distribution, compartment diversity, extinction slope, H2) may lead to rather long runtimes!

**Author(s)**

Carsten F. Dormann <carsten.dormann@ufz.de>

**Examples**

```
data(mosquin1967)
null.t.test(mosquin1967, index=c("generality", "vulnerability",
  "cluster coefficient", "H2", "ISA", "SA"), nrep=2, N=10)
```

---

 nullmodel

*Generates null models for network analysis*


---

**Description**

A function for convenient generation of null models for quantitative and binary networks.

**Usage**

```
nullmodel(web, N=1000, method="r2d", ...)
```

**Arguments**

web	Web is a matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
N	number of null models to be generated; defaults to 1000 (more might be better, less probably not).
method	Null model type. Can be given as an integer or name: 1/"r2dtable", 2/"swap.web", 3/"shuffle.web", 4/"mgen"; allows for partial match of names; methods 1 to 3 works for quantitative webs, 3 and 4 for binary.
...	arguments to be passed to the function generating the specific null models

**Details**

This is only a wrapper function to facilitate and standardise the generation of null models.

**Value**

Returns a list of N null model-generated networks. Species names are (obviously) dropped.

**Author(s)**

Carsten F. Dormann <carsten.dormann@ufz.de>

**See Also**

For the functions generating the null model network: [shuffle.web](#), [swap.web](#), [mgen](#), [commsimulator](#) and [r2dtable](#)

**Examples**

```
data(Safariland)
nullmodel(Safariland, N=2, method=1)
nullmodel(Safariland>0, N=2, method=4)
# analysis example:
obs <- unlist(networklevel(Safariland, index="weighted nestedness"))
nulls <- nullmodel(Safariland, N=100, method=1)
null <- unlist(sapply(nulls, networklevel, index="weighted nestedness")) #takes a while ...

plot(density(null), xlim=c(min(obs, min(null)), max(obs, max(null))), main="comparison of ob
abline(v=obs, col="red", lwd=2)

praw <- sum(null>obs) / length(null)
ifelse(praw > 0.5, 1-praw, praw) # P-value

# comparison of null model 3 and 4 for binary:
nulls3 <- nullmodel(Safariland>0, N=100, method=3)
nulls4 <- nullmodel(Safariland>0, N=100, method=4)
null3 <- unlist(sapply(nulls3, networklevel, index="weighted nestedness"))
null4 <- unlist(sapply(nulls4, networklevel, index="weighted nestedness"))

plot(density(null3), xlim=range(c(null3, null4)), lwd=2, main="comparison of null models")
lines(density(null4), col="red", lwd=2)
legend("topright", c("shuffle", "mgen"), col=c("black", "red"), lwd=c(2,2), bty="n", cex=1.5)
abline(v=networklevel(Safariland>0, index="weighted nestedness"))
```

---

olesen2002aigrettes

*A pollination network.*

---

**Description**

The structure of pollination networks were investigated for two oceanic islands, the Azorean Flores and the Mauritian Ile aux Aigrettes.

**Usage**

```
data(olesen2002aigrettes)
```

**Details**

The authors recorded their data by counting the number of visits of each flower visitor species to each plant species. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

**References**

Olesen, J. M., L. I. Eskildsen, and S. Venkatasamy. 2002. Invasion of pollination networks on oceanic islands: importance of invader complexes and endemic super generalists. *Diversity and Distributions* 8:181-192.

**Examples**

```
data(olesen2002aigrettes)
## maybe str(olesen2002aigrettes) ; plot(olesen2002aigrettes) ...
```

---

olesen2002flores *A pollination network.*

---

**Description**

The structure of pollination networks were investigated for two oceanic islands, the Azorean Flores and the Mauritian Ile aux Aigrettes.

**Usage**

```
data(olesen2002flores)
```

**Details**

The authors recorded their data by counting the number of visits of each flower visitor species to each plant species. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

**References**

Olesen, J. M., L. I. Eskildsen, and S. Venkatasamy. 2002. Invasion of pollination networks on oceanic islands: importance of invader complexes and endemic super generalists. *Diversity and Distributions* 8:181-192.

**Examples**

```
data(olesen2002flores)
## maybe str(olesen2002flores) ; plot(olesen2002flores) ...
```

---

`ollerton2003``ollerton2003`

---

**Description**

A pollination network

**Usage**

```
data(ollerton2003)
```

**Format**

A data frame with 9 observations on the following 56 variables.

**Details**

The study was conducted in the KwaZulu-Natal region of South Africa. During 3 months of field-work the flower visitors and pollinators to a assemblage of nine asclepiads at an upland grassland site were studied. Two of the specialized pollination systems that were documented are new to the asclepiads: fruit chafer pollination and pompilid wasp pollination (the latter is almost unique in the angiosperms).

The authors recorded their data by counting the number of individual flower visitors observed and/or caught on each plant species. The total number of individuals observed on each plant species provide a rough estimate of the level of visitation that each species received. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

**Source**

NCEAS data base on interaction webs: <http://www.nceas.ucsb.edu/interactionweb/html/datasets.html>

**References**

Ollerton, J., S.D. Johnson, L. Cranmer and S. Kellie (2003) The pollination ecology of an assemblage of grassland asclepiads in South Africa. *Annals of Botany* **92** 807-834

**Examples**

```
data(ollerton2003)
plotweb(ollerton2003)
## maybe str(ollerton2003) ; plot(ollerton2003) ...
```

**Description**

Quantifies, for each pair of lower trophic level species, the potential for showing apparent competition with another species, mediated through the higher trophic level.

**Usage**

PAC(web)

**Arguments**

web                    A host-parasitoid network (or alike), where the entries represent the **sum of parasitoids emerging from the interactions between parasitoid and host** (i.e. number of interactions \* number of parasitoid individuals emerging from each host). Only if there is only one parasitoid per host this web will be the same as that used in all other calculations in this package!

**Details**

Calculates the potential for apparent competition (Holt 1977), following the formula given in Müller et al. (1999) and Morris et al. (2005). See also Morris et al. (2004) for an experimental test.

**Value**

Returns a  $k \times k$  matrix with entries  $d_{ij}$ , where  $k$  is the number of species in the lower trophic level and  $i$  and  $j$  are lower trophic level species. Entries in the upper triangle represent the effect of the species  $j$  on species  $i$  ( $d_{ij}$ ), i.e. the effect of the species in the column onto that in the row. In analogy, the lower triangle represent the inverse direction ( $d_{ji}$ ), i.e. the effect of the row species onto the column species. Diagonal entries are “apparent intraspecific competition”.

**Note**

The idea is that in host-parasitoid networks one host also affects other hosts by the number of parasitoid that hatch from it and are thus added to the pool of parasitoids. An abundant, large host can (involuntarily) contribute many parasitoids to the pool, thus also increasing the parasitoid burden of other hosts. This looks like competition between the two hosts, while in fact it is mediated through the other trophic level.

Whether this concept can be usefully applied to mutualist networks (such as flower visitation networks, aka pollination webs) is still under debate. The example below has thus to be seen as a technical, not a biological example.

**Author(s)**

Carsten F. Dormann <carsten.dormann@ufz.de>

## References

- Holt, R. D. 1977 Predation, apparent competition and the structure of prey communities. *Theoretical Population Biology* **12**, 197–229.
- Morris, R. J., Lewis, O. T. and Godfray, H. C. J. 2004 Experimental evidence for apparent competition in a tropical forest food web. *Nature* **428**, 310–313.
- Morris, R. J., Lewis, O. T. and Godfray, H. C. J. 2005 Apparent competition and insect community structure: towards a spatial perspective. *Annales Zoologica Fennici* **42**, 449–462.
- Müller, C. B., Adriaanse, I. C. T., Belshaw, R. and Godfray, H. C. J. 1999 The structure of an aphid-parasitoid community. *Journal of Animal Ecology* **68**, 346–370

## Examples

```
data(Safariland)
PAC(Safariland)
```

---

```
plotweb
```

*Visualize a bipartite interaction matrix (e.g. a foodweb)*

---

## Description

A two dimensional matrix is plotted as a bipartite graph.

## Usage

```
plotweb(web,
  method = "cca", empty = TRUE, labsize = 1, ybig = 1,
  y.width.low = 0.1,
  y.width.high = 0.1,
  low.spacing = NULL,
  high.spacing = NULL,
  arrow="no", col.interaction="grey80",
  col.high = "grey10", col.low="grey10",
  bor.col.interaction = "black", bor.col.high="black", bor.col.low="black",
  high.lablength = NULL, low.lablength = NULL,
  sequence=NULL, low.abun=NULL, low.abun.col="green",
  bor.low.abun.col = "black",
  high.abun=NULL, high.abun.col="red", bor.high.abun.col="black",
  text.rot=0, text.high.col="black", text.low.col="black",
  ad.high=NULL,
  ad.low=NULL,
  plot.axes = FALSE,
  low.y=0.5, high.y=1.5,
  add=FALSE,
  y.lim=NULL,
  x.lim=NULL,
  low.plot=TRUE,
```

```

high.plot=TRUE,
high.xoff = 0,
low.xoff = 0,
high.lab.dis = NULL,
low.lab.dis = NULL)

```

## Arguments

web	Web is a matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
method	Default method is 'cca', which leads to as few crossings of interactions as possible. The other option is 'normal', which leaves order as given by the matrix.
empty	logical; should empty columns or empty rows be omitted from plotting; defaults to true
labsize	factor for size of labels, default is 1
ybig	vertical distance between upper and lower boxes, default is 1
y.width.low	width of lower boxes, default is 0.1
y.width.high	width of upper boxes, default is 0.1
low.spacing	distance between lower boxes, default is NULL, so automatically spaced that length of upper and lower boxes is the same. Be aware if set to any value that x.lim may has to be adjusted to ensure that the network is not plotted outside the plotting region
high.spacing	distance between upper and lower boxes, default is is NULL, so automatically spaced that length of upper and lower boxes is the same. Be aware if set to any value that x.lim may has to be adjusted to ensure that the network is not plotted outside the plotting region
arrow	display type of connection between upper and lower boxes, options are 'up', 'down', 'both', 'up.center', 'down.center', 'both.center' and 'no', default is 'no', which is a polygonal connection between boxes.
col.interaction	color of interaction, default is grey80.
col.high	color of upper boxes, default is grey10.
col.low	color of lower boxes, default is grey10.
bor.col.interaction	border color of interaction, default is black
bor.col.high	border color of upper boxes, default is black
bor.col.low	border color of lower boxes, default is black
high.lablenth	number of characters of upper labels that should be plotted. If zero no labels are shown, default is NULL which plots the complete labels.

<code>low.lablenth</code>	number of characters of lower labels that should be plotted. If zero no labels are shown, default is NULL which plots the complete labels.
<code>sequence</code>	list of two with two names vectors: <code>seq.pred</code> and <code>seq.prey</code> , which specify the order in which species are plotted. Cannot be set for <code>'method="cca"</code> '. Defaults to NULL, where the sequence remains as given or is determined by the CCA internally.
<code>low.abun</code>	Vector with independent abundance estimates for the lower trophic level, NULL if none exists.
<code>low.abun.col</code>	Colour for depicting the abundance estimates for the lower trophic level; defaults to green.
<code>bor.low.abun.col</code>	border color for depicting the abundance estimates for the lower trophic level, default is black
<code>high.abun</code>	Vector with independent abundance estimates for the higher trophic level, NULL if none exists.
<code>high.abun.col</code>	Colour for depicting the abundance estimates for the lower trophic level; defaults to red.
<code>bor.high.abun.col</code>	border color for depicting the abundance estimates for the higher trophic level, default is black
<code>text.rot</code>	orientation of labels in the plot (to avoid overlapping of horizontal labels if dimension of the webs are high), default is 0 for horizontal labels, use <code>text.rot=90</code> for vertical labels.
<code>text.high.col</code>	Colour for text labels of higher trophic level, a vector of colours can be given
<code>text.low.col</code>	Colour for text labels of lower trophic level. A vector of colours can be given
<code>ad.high</code>	Adjust upper labels. See <code>adj</code> in <code>?text</code> how to adjust labels
<code>ad.low</code>	Adjust upper labels. See <code>adj</code> in <code>?text</code> how to adjust labels
<code>plot.axes</code>	axis are plotted. Sometimes useful to place boxes in higher order plots. Defaults to FALSE
<code>low.y</code>	Position of the higher boxes on the y-axis. Defaults to 1.5
<code>high.y</code>	Position of the higher boxes on the y-axis. Defaults to 1.5
<code>add</code>	if set to TRUE a new bipartite network can be added to the previous plot. So multitrophic webs can be plotted, see examples below. Defaults to FALSE
<code>y.lim</code>	To set limits for y-axis. Useful if labels are plotted outside the plotting region and for multitrophic plots, see examples below
<code>x.lim</code>	To set limits for x-axis. Useful if labels are plotted outside the plotting region and for multitrophic plots, see examples below
<code>low.plot</code>	Defines if lower boxes should be drawn. Use in multitrophic plots to avoid plotting boxes of some trophis - see examples below. Defaults to TRUE
<code>high.plot</code>	Defines if higher boxes should be drawn. Use in multitrophic plots to avoid plotting boxes of some trophis - see examples below. Defaults to TRUE

<code>high.xoff</code>	allows to set an offset to upper boxes. Useful if <code>high.spacing</code> is used to center boxes manually. Use <code>plot.axes=TRUE</code> for easy centering
<code>low.xoff</code>	allows to set an offset to lower boxes. Useful if <code>low.spacing</code> is used to center boxes manually. Use <code>plot.axes=TRUE</code> for easy centering
<code>high.lab.dis</code>	normally labels are staggered to avoid plotting over themselves. if set to 0, higher labels are all on one horizontal line. By using <code>ad.low</code> the position of the labels can be adjusted. If set to any other value labels are staggerd with this distance. Defaults to NULL
<code>low.lab.dis</code>	normally labels are staggered to avoid plotting over themselves. if set to 0, lower labels are all on one horizontal line. By using <code>ad.low</code> the position of the labels can be adjusted. If set to any other value labels are staggerd with this distance. Defaults to NULL

**Value**

Returns a window with a bipartite graph of a foodweb. For all colours vectors can be used (which are recycled if length differs. Now more trophic webs can be plotted by using `plotweb` and the `add` switch, which allows to add more webs and staggering them on top of each other. Preferred option is here to order webs by yourself and use `method="normal"` to keep your preferred order. See examples on three and four trophic networks.

**Author(s)**

Bernd Gruber <bernd.gruber@ufz.de>

**References**

Tylianakis, J. M., Tscharnkte, T. and Lewis, O. T. (2007) Habitat modification alters the structure of tropical host-parasitoid food webs. *Nature* **445**, 202–205

**See Also**

For a different plot of foodwebs see [visweb](#)

**Examples**

```
data(Safariland)
plotweb(Safariland)

# shorter labels
plotweb(Safariland, high.lablenth=3, low.lablenth=0, arrow="down")

# centered triangles for displaying interaciones
plotweb(Safariland, text.rot=90, arrow="down.center", col.interaction="wheat2", y.lim=c(-1,2.

#original sequence, up arrows and different box width
plotweb(Safariland, method="normal", arrow="up", y.width.low=0.3, low.lablenth=4)
# interactions as lines
plotweb(Safariland, arrow="both", y.width.low=0.05, text.rot=90, col.high="blue", col.low="g
```

```

# add an abundance vector for lower trophic species
low.abun = round(runif(dim(Safariland)[1],1,40)) #create
names(low.abun) <- rownames(Safariland)
plotweb(Safariland, text.rot=90, low.abun=low.abun, col.interaction="purple", y.width.low=0.1,
        plotweb(Safariland, text.rot=90, low.abun=low.abun, col.interaction = "red", bor.col.interact

# now vectors for all colours can be given, to mark certain species or interactions. Colour
plotweb(Safariland,col.high=c("orange","green"))
plotweb(Safariland,col.low=c("orange","green"),col.high=c("white","grey","purple"),text.high

#example one (tritrophic)
plotweb(Safariland,y.width.low=0.1, y.width.high=0.05,method="normal", y.lim=c(0,3), arrow="no",
        plotweb(t(Safariland), y.width.low=0.05, y.width.high=0.1, method="normal", add=TRUE,low.y=1

#example two (4 trophic with abundance)
low.abun = round(runif(dim(Safariland)[1],1,40)) #create
names(low.abun) <- rownames(Safariland)
plotweb(Safariland, text.rot=90, high.abun=low.abun, col.interaction="purple", y.lim=c(0,4.5),
        plotweb(t(Safariland), y.width.low=0.05, y.width.high=0.1, method="normal", add=TRUE, low.y=1
        plotweb(Safariland,y.width.low=0.05, y.width.high=0.1, method="normal", add=TRUE, low.y=2.95

```

---

plotweb2

*Visualize a tripartite interaction matrix (e.g. a tritrophic foodweb)*

---

## Description

Two two dimensional matrix are plotted as a tripartite graph.

## Usage

```

plotweb2(web, web2, method = "cca", empty = FALSE, labsize = 1, ybig = 1,
         y_width = 0.1, spacing = 0.05, arrow="no", col.interaction="grey80",
         col.pred = "grey10", col.prey="grey10", lab.space=1, lablength = NULL,
         sequence=NULL, low.abun=NULL,low.abun.col="green", high.abun=NULL,
         high.abun.col="red", method2 = "cca", empty2 = TRUE, spacing2 = 0.05,
         arrow2="no", col.interaction2="grey80", col.pred2 = "grey30",
         col.prey2="grey20", lablength2 = NULL, sequence.pred2=NULL,low.abun2=NULL,
         low.abun.col2="green", high.abun2=NULL, high.abun.col2="red")

```

**Arguments**

<code>web</code>	Web is a matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
<code>web2</code>	The other web to be included.
<code>method</code>	Default method is 'cca', which leads to as few crossings of interactions as possible. The other option is 'normal', which leaves order as given by the matrix.
<code>empty</code>	logical; should empty columns or empty rows be omitted from plotting; defaults to true
<code>labsize</code>	factor for size of labels, default is 1
<code>ybig</code>	vertical distance between upper and lower boxes, default is 1
<code>y_width</code>	width of upper and lower boxes, default is 0.1
<code>spacing</code>	horizontal distance between upper and lower boxes, default is 0.05
<code>arrow</code>	display type of connection between upper and lower boxes, options are 'up', 'down', 'both' and 'no', default is 'no', which is a polygonal connection between boxes.
<code>col.interaction</code>	color of interaction, default is grey80.
<code>col.pred</code>	color of upper boxes, default is grey10.
<code>col.prey</code>	color of lower boxes, default is grey10.
<code>lab.space</code>	sometimes it is necessary to add additional space for labels below and above of the boxes, so all labels are shown, default is 1.
<code>lablength</code>	number of characters of labels that should be plotted. If zero no labels are shown, default is NULL which plots the complete labels.
<code>sequence</code>	list of two with two names vectors: <code>seq.pred</code> and <code>seq.prey</code> , which specify the order in which species are plotted. Cannot be set for 'method="cca"'. Defaults to NULL, where the sequence remains as given or is determined by the CCA internally.
<code>low.abun</code>	Vector with independent abundance estimates for the lower trophic level, NULL if none exists.
<code>low.abun.col</code>	Colour for depicting the abundance estimates for the lower trophic level; defaults to green.
<code>high.abun</code>	Vector with independent abundance estimates for the higher trophic level, NULL if none exists.
<code>high.abun.col</code>	Colour for depicting the abundance estimates for the higher trophic level; defaults to red.
<code>method2</code>	Default method is 'cca', which leads to as few crossings of interactions as possible. The other option is 'normal', which leaves order as given by the matrix.

<code>empty2</code>	logical; should empty columns or empty rows be omitted from plotting; defaults to true
<code>spacing2</code>	horizontal distance between upper and lower boxes, default is 0.05
<code>arrow2</code>	display type of connection between upper and lower boxes, options are 'up', 'down', 'both' and 'no', default is 'no', which is a polygonal connection between boxes.
<code>col.interaction2</code>	color of interaction, default is grey80.
<code>col.pred2</code>	color of upper boxes, default is grey10.
<code>col.prey2</code>	color of lower boxes, default is grey10.
<code>lablength2</code>	number of characters of labels that should be plotted. If zero no labels are shown, default is NULL which plots the complete labels.
<code>sequence.pred2</code>	list of two with two names vectors: <code>seq.pred</code> and <code>seq.prey</code> , which specify the order in which species are plotted. Cannot be set for 'method="cca"'. Defaults to NULL, where the sequence remains as given or is determined by the CCA internally.
<code>low.abun2</code>	Vector with independent abundance estimates for the lower trophic level, NULL if none exists.
<code>low.abun.col2</code>	Colour for depicting the abundance estimates for the lower trophic level; defaults to green.
<code>high.abun2</code>	Vector with independent abundance estimates for the higher trophic level, NULL if none exists.
<code>high.abun.col2</code>	Colour for depicting the abundance estimates for the lower trophic level; defaults to red.

**Value**

Returns a window with a tripartite graph of a foodweb.

**Author(s)**

Bernd Gruber <bernd.gruber@ufz.de>

**See Also**

For a different plot of foodwebs see [visweb](#) and [plotweb](#)

---

`robustness`*Robustness to species extinctions*

---

**Description**

Calculates the area below the extinction curve generated by `second.extinct`.

**Usage**

```
robustness(object)
```

**Arguments**

`object` An object of type class `bipartite`, usually generated by `second.extinct`.

**Details**

This function calculates the area below the extinction curve generated by `second.extinct` as a measure of the robustness of the system to the loss of species.

The curve, first proposed by Memmott et al. (2004), is based on the fact that if a given fraction of species of one guild (for instance, the pollinators) are eliminated, a number of species of the other guild (e.g. plants) which depend on their interactions become extinct. The slope and general shape of the curve provided a straightforward graphic description of the tolerance of a system to the extinction of its component species.

An improvement of Memmott et al.'s curve was developed by Burgos et al. (2007) by introducing a quantitative measure of robustness with a single parameter  $R$ , defined as the area under the extinction curve. It is intuitive that  $R = 1$  corresponds to a curve that decreases very mildly until the point at which almost all animal species are eliminated. This is consistent with a very robust system in which, for instance, most of the plant species survive even if a large fraction of the animal species is eliminated. Conversely  $R = 0$  corresponds to an ATC that decreases abruptly as soon as any species is lost. This is consistent with a fragile system in which, for instance, even if a very small fraction of the animal species is eliminated, most of the plants lose all their interactions and go extinct.

**Value**

Returns the robustness of the web to the removal of species.

**Note**

This index complements the information given by `slope.bipartite`, although it has the advantage of not being constrained by the shape of the particular curve (concave or convex).

**Author(s)**

Mariano Devoto <mdevoto@agro.uba.ar>

## References

- Burgos, E., H. Ceva, R.P.J. Perazzo, M. Devoto, D. Medan, M. Zimmermann, and A. Maria Delbue (2007) Why nestedness in mutualistic networks? *Journal of Theoretical Biology* **249**, 307–313
- Memmott, J., Waser, N. M. and Price, M. V. 2004 Tolerance of pollination networks to species extinctions. *Proceedings of the Royal Society B* **271**, 2605–2611

## See Also

[second.extinct](#) for generating the required input object and [slope.bipartite](#) for an alternative, but inferior measure

## Examples

```
data(Safariland)
ex <- second.extinct(Safariland, participant="low", method="ran", nrep=100, details=FALSE)
robustness(ex)
```

---

Safariland

*Pollination web in Argentina.*

---

## Description

This pollination web was published by Vazquez and Simberloff (2003). See there for details on how it was measured.

## Usage

```
data(Safariland)
```

## Details

The dataset consists of a matrix with 9 rows, representing plant species and 27 columns, representing different pollinators. Values in the matrix are observed flower visitations.

This dataset is fairly representative of a pollination web: more pollinators than plants, and an awful lot of zeros in the matrix.

The study was conducted in four grazed and four ungrazed sites in and around Nahuel Huapi National Park and surrounding areas in Rio Negro, Argentina from September 1999 to February 2000. For each site, the plant-pollinator interaction network was described.

The authors recorded their data by counting the number of visits of each flower visitor species to each plant species. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

### Source

These data can be downloaded, together with the other datasets, on the NCEAS interactionweb website <http://www.nceas.ucsb.edu/interactionweb/html/datasets.html>. See also there for further details, both on the data and their usage.

### References

Vázquez, D. P. 2002 Interactions among Introduced Ungulates, Plants, and Pollinators: A Field Study in the Temperate Forest of the Southern Andes. Doctoral Dissertation Thesis, University of Tennessee, Knoxville, Tennessee, USA.

Vázquez, D. P., and D. Simberloff. 2002 Ecological specialization and susceptibility to disturbance: conjectures and refutations. *American Naturalist* **159**, 606–623.

Vázquez, D. P., and D. Simberloff. 2003 Changes in interaction biodiversity induced by an introduced ungulate. *Ecology Letters*, **6**, 1077–1083.

### Examples

```
data(Safariland)
plotweb(Safariland)
```

---

schemske1978

*A pollination network.*

---

### Description

Populations of seven early flowering, low-growing, perennial woodland herbs were studied in a 24 hectare sit in Brownfield Woods, close to Urbana, Illinois, U.S.A. The sequence and variety of flowers visited by individual insects were recorded for the first foraging insect observed in a randomly selected, 1 m<sup>2</sup> quadrat located within a 10x10 m grid that included representatives of several plant species.

### Usage

```
data(schemske1978)
```

### Details

The authors recorded their data by counting the number of visits of each flower visitor species to each plant species. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

### References

Schemske, D. W., M. F. Willson, M. N. Melampy, L. J. Miller, L. Verner, K. M. Schemske, and L. B. Best. 1978. Flowering Ecology of Some Spring Woodland Herbs. *Ecology* 59:351-366.

**Examples**

```
data(schemske1978)
## maybe str(schemske1978) ; plot(schemske1978) ...
```

---

```
second.extinct      Secondary extinctions in bipartite networks
```

---

**Description**

Calculates the consequences of removing a species from a bipartite network. With plotting and slope-estimation functionality.

**Usage**

```
second.extinct(web, participant = "higher", method = "abun", nrep = 10,
  details = FALSE)
```

**Arguments**

web	Web is a matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
participant	“high” (default) or “low” or “both”, depending if you want to exterminate higher trophic level species, lower trophic level species, or a randomly chosen species of both levels; partial matching.
method	Random deletion of a species (“random”) or according to its abundance, with least abundant going extinct first (“abundance”; default).
nrep	Number of replicates of extermination sequence. Will not be considered for method abundance.
details	Logical; returns details, i.e.~for each replicate the sequence of secondary extinctions. If set to FALSE (default), replicated runs will be averaged.

**Details**

The procedure of this function is simple. For example imagine the web to represent a pollination web, in which pollinators die one by one. Set all entries of a column to zero, see how many rows are now also all-zero (i.e. species that are now not pollinated any more), and count these as secondary extinctions of the primary exterminated pollinator.

Internally, each extermination is achieved by a call to [extinction](#), followed by a call to [empty](#), which counts the number of all-zero columns and rows.

Although written for pollination webs (hence the nomenclature of pollinator and plant), it can be similarly applied to other types of bipartite networks. It is called by [networklevel](#).

‘details=FALSE’ will not work with ‘participant=“both”’, since sequences cannot be averaged, due to different lengths.

**Value**

Function returns an object of class “bipartite” to ensure proper working of function `plot.bipartite`.

If ‘`details=FALSE`’, the returned object contains a matrix with columns representing the number of species going extinct from one step to the next, averaged across all runs.

If ‘`details=FALSE`’, the returned object will be a list of matrices containing the number of species going extinct at each step.

The objects attribute “`exterminated`” gives the name of the trophic level (pollinator or plant).

**Note**

Note: The length of an extinction sequence is obviously given by the number of species in the selected trophic level. When setting ‘`participant="both"`’, lengths will differ for each replicate run, since it is unpredictable in which sequence species go extinct, and hence how many secondary extinctions will pre-empt further primary extinctions.

**Author(s)**

Carsten F. Dormann

**References**

Memmott, J., Waser, N. M. and Price, M. V. (2004) Tolerance of pollination networks to species extinctions. *Proceedings of the Royal Society b* **271**, 2605–2611.

**See Also**

`networklevel` calls `second.extinct`; `extinction` and `empty` are internal helper functions, and `slope.bipartite` calculates extinction slopes from the output of `second.extinct`.

**Examples**

```
data(Safariland)
(ex <- second.extinct(Safariland, participant="low", method="r", nrep=50, details=TRUE))
(ex <- second.extinct(Safariland, participant="low", method="r", nrep=50, details=FALSE))
```

---

shuffle.web

*Shuffle web entries*

---

**Description**

Shuffles (= relocates) entries in a web matrix whilst maintaining the dimensionality.

**Usage**

```
shuffle.web(web, N)
```

## Arguments

web	An interaction matrix.
N	Number of desired shuffled matrices.

## Details

This function is designed to behave similar to `r2dtable`, i.e. it returns a list of randomised matrices. In contrast to `r2dtable` it does not keep marginal sums constant!

This function is thought of as a nullmodel for the analysis of bipartite webs. It keeps two web properties constant: The number of interactions and the number of links (and hence connectance). A comparison of `shuffle.web`- and `r2dtable`-based webs allows to elucidate the effect of marginal sums.

## Value

A list of N randomised matrices with the same dimensions as the initial web.

## Note

`shuffle.web` is not an extremely intelligent nullmodel. You may want to think of a better one for your specific application!

## Author(s)

Carsten F. Dormann <carsten.dormann@ufz.de>

## References

This nullmodel can be thought of as a quantitative version of Fortuna & Bascompte (2006) “null model 1”:

Fortuna, M. A., and J. Bascompte. 2006. Habitat loss and the structure of plant-animal mutualistic networks. *Ecology Letters* 9: 281-286.

For a very nice and thorough overview of nullmodels in general see:

Gotelli, N. J., and G. R. Graves. 1996. *Null Models in Ecology*. Smithsonian Institution Press, Washington D.C.

For nullmodels and their application to webs/networks see, e.g.:

Vázquez, D. P., and M. A. Aizen. 2003. Null model analyses of specialization in plant-pollinator interactions. *Ecology* 84: 2493-2501.

Vázquez, D. P., C. J. Melián, N. M. Williams, N. Blüthgen, B. R. Krasnov, and R. Poulin. 2007. Species abundance and asymmetric interaction strength in ecological networks. *Oikos* 116: 1120-1127.

## See Also

[r2dtable](#)

**Examples**

```
data(Safariland)

shuffle.web(Safariland, N=2)
```

---

slope.bipartite      *Slope of extinction simulation*

---

**Description**

Fits a hyperbolic function to the extinction simulation of `second.extinct`.

**Usage**

```
slope.bipartite(object, plot.it = TRUE, ...)
```

**Arguments**

<code>object</code>	An object of class “bipartite”, usually generated by <code>second.extinct</code> .
<code>plot.it</code>	Logical; want to see the graph?
<code>...</code>	Graphical parameters passed on to the <code>plot</code> command used for plotting, NOT to the <code>curve</code> command used for overlaying the curve.

**Details**

Function scales extinction sequences to values between 0 and 1 for each participant. The x-axis of the graph features the proportion of exterminated participants, while the y-axis depicts the proportion of secondary extinctions. Since these curves usually follow a hyperbolic function (see examples in Memmott et al. 2004), this is fitted to the data.

At present, only a function of type  $y \sim 1 - x^a$  is fitted (using `nls`), i.e. without offset. While usually this function provides very good fits, do check the graph and judge for yourself. Fitting this simple function makes its parameter ‘a’ a measure of extinction vulnerability. The more gradual the secondary extinctions, the lower the absolute value of ‘a’. Or, phrased differently, large absolute values of ‘a’ indicate a very abrupt die-off, indicative of high initial redundancy in the network.

**Value**

Returns one number, the exponent of the fitted hyperbolic model.

**Note**

This function is not as vigorously tested as it should probably be. It worked fine for large networks, but small ones may behave strangely, I fathom.

Note also that it CANNOT fit concave curves! The most extreme is a linear decline of secondary extinctions with primary extinctions. The function can be adapted to handle this, but only at the expense of fitting another parameter.

**Author(s)**

Carsten F. Dormann

**References**

Memmott, J., Waser, N. M. and Price, M. V. (2004) Tolerance of pollination networks to species extinctions. *Proceedings of the Royal Society B* **271**, 2605–2611

**See Also**

[second.extinct](#) for generating the required input object.

**Examples**

```
data(Safariland)
ex <- second.extinct(Safariland, participant="low", method="ran", nrep=100,
  details=FALSE)
slope.bipartite(ex)
```

---

small1976

*A pollination network.*

---

**Description**

The study took place in the Mer Bleue peat bog of Ottawa, Canada in 1973. The paper is a preliminary evaluation of the pollination relationships of the major entomophilous plant species of the Mer Bleue.

**Usage**

```
data(small1976)
```

**Details**

The authors recorded their data by counting the number of individual flower visitors caught on each plant species. The total number of individuals collected on each plant species provide a rough estimate of the level of visitation that each species received. Data are presented as an interaction frequency matrix, in which cells with positive integers indicate the frequency of interaction between a pair of species, and cells with zeros indicate no interaction.

**Source**

[http://www.nceas.ucsb.edu/interactionweb/html/small\\_1976.html](http://www.nceas.ucsb.edu/interactionweb/html/small_1976.html)

**References**

Small, E. 1976. Insect pollinators of the Mer Bleue peat bog of Ottawa. *Canadian Field Naturalist* 90:22-28.

**Examples**

```
data(small1976)
## maybe str(small1976) ; plot(small1976) ...
```

---

sortweb	<i>Function to sort bipartite webs</i>
---------	--

---

**Description**

This function sorts bipartite webs by either increasing/decreasing row and column totals or by a given sequence

**Usage**

```
sortweb(web, sort.order="inc", sequence=NULL)
```

**Arguments**

web	A matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
sort.order	sort.order can be either <b>inc</b> : sorted by increasing row/column totals <b>dec</b> : sorted by decreasing row/column totals <b>seq</b> : sorted by a given order, see sequence for how to specify an order
sequence	list of two with two named vectors: <code>seq.lower</code> and <code>seq.higher</code> , which specify the order in which species are sorted. To be able to sort by names obviously the given web must be a NAMED matrix, i.e. has column and row names. If you want to order an unnamed web, you can either order it by row/column totals are you have to simply specify the sequence as follows (example puts last row/column to the front): <code>web[c(9, 1:8), c(27, 1:26)]</code>

**Value**

Returns an ordered bipartite web.

**Author(s)**

Bernd Gruber

**References**

Vazquez, P.D., Chacoff, N.,P. and Cagnolo, L. (2009) Evaluating multiple determinants of the structure of plant-animal mutualistic networks. *Ecology* in press.

**See Also**

For plotting and ordered web see `plotweb`, `'method="normal"'` or `visweb`, `'type="none"'`.

**Examples**

```
data(Safariland)
web <- Safariland

sortweb(Safariland, sort.order="dec")
plotweb(sortweb(Safariland, sort.order="inc"), method="normal") #rares species first
visweb(sortweb(Safariland, sort.order="inc"), type="diagonal", square="compartment", text="no")

# sorted by a given (here random) sequence
sequence <- list(seq.higher=sample(colnames(Safariland)), seq.lower=sample(rownames(Safariland)))
web.ordered <- sortweb(web, sort.order="seq", sequence=sequence)
```

---

specieslevel

*Calculate various indices for network properties at the species level*

---

**Description**

Apart from the properties of the entire web, also its participants can be described specifically. Various simple numbers and indices are calculated and returned.

**Usage**

```
specieslevel(web, index="ALL", logbase="e", low.abun=NULL, high.abun=NULL)
```

**Arguments**

web	Web is a matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
index	Vector of indices to be calculated for each trophic level of the web; options are: “species number” for number of species, “degree”, “dependence”, “interaction” for interaction push/pull (our version of dependence asymmetry: see details), “PSI” for pollination service index (or pollinator support index, depending on the trophic level), “NS” for node specialisation index, “Fisher” for Fisher’s alpha index, “diversity” for Shannon diversity of interactions of that species, “effective partners” for the effective number of interacting partners (e.g. predators/pollinators or hosts/plants), “d” for Blüthgen’s d, “strength” as sum of dependencies or “ALL” for all the aforementioned (default). “ALLBUTD” excludes only the dependence matrix and leads to the output being simplified to a matrix per trophic level.

logbase	Shall the various diversity indices be calculated to the base of e (default) or 2? Log2 is the proposal for generality and vulnerability by Bersier et al. (2002), while Shannon uses ln. The choice of the base will not affect the results qualitatively, only by a scaling factor.
low.abun	Optional vector of independent abundances of the <i>lower</i> trophic level to be passed on to <code>dfun</code> .
high.abun	Optional vector of independent abundances of the <i>higher</i> trophic level to be passed on to <code>dfun</code> .

## Details

This function implements a variety of the many (and still procreating) indices describing species properties. Some are embaracingly simple (such as number of interacting species for each species). Others are variations on Shannon's diversity index applied to within species.

Most indices are straightforward, one-line formulae; some, such as  $d'$ , also require a re-arranging of the matrix. We (Dormann, Blüthgen, Gruber) came up with a new one, called "Pollination Service Index" or  $\psi$ , for which a few more details seem appropriate.

PSI is comprised of three calculation steps: firstly, we calculate, for each pollinator species, the proportion to which it visits each plant species (or, phrased anthropomorphically, the number to the question: To which proportion do I visit dandelion?). Secondly, we calculate the proportion to which a plant is visited by each bee species (Which proportion of my pollinators are red mason bees?). Multiplying, these two proportions gives the portion of own pollen for each plant species (because this depends both on a pollinators specialisation (step 1) and the plant's specific receptiveness (step 2). Finally, we sum the proportions own pollen delivered across all plant species. This value is the PSI-value. At its maximum, 1, it shows that all pollen delivered to exactly one plant species is derived from one monolectic pollinator. At its minimum, 0, it indicates that a pollinator is irrelevant to all plant species. Note that PSI is independent of the abundance of plants or pollinators: a bee been found only once on a plant species visited by no-one else receives a PSI of 1, even if in total 14 million visits were recorded.

We envisage a penalty for the fact that a pollinator has to make two (more or less successive) visits to the same plant species: the first to take the pollen up, the second to pollinate the next. Thus, using 2 as an exponent in step 1 would simulate that a pollinator deposits all pollen at every second visit (or flies to their nest). That is certainly a too strong penalisation. At present we set the exponent to 1, because the step of controlling for "pollen purity" is already a major improvement. Also, we have no idea to which extent pollen gets mixed and/or lost during foraging flights, and the true exponent remains elusive.

## Value

For both the "higher trophic level" and the "higher trophic level" a list with the following components:

```
number of species
      Sic.
species degree
      Sum of interactions per species.
```

dependence	Web entry divided by column totals (for higher trophic level) or row totals (for lower trophic level). Expresses the proportion of interactions with each species, and sums to 1 for each.
strength	Sum of dependencies of each species (proposed in Bascompte et al. 2006).
interaction	push/pull Direction of interaction asymmetry: positive values indicate that a species affects the species of the other level it interacts with stronger than they affect it (“pusher”); negative values indicate that a species is, on average, on the receiving end of the stick (“being pulled”); formula based on Vazquez et al (2007), but push/pull is our own nomenclature. Values are highly correlated with species strengths (see below), but standardised to fall between -1 (being pulled) and 1 (pushing).
Pollination	Service Index PSI for the higher trophic level, and the equivalent Pollinator Support Index for the lower trophic level. Expresses the value of, say, a pollinator for all plant species as the sum (across all plant species) of the proportion of pollen deposited per visit. The more specialised the pollinator and the fewer pollinators the plants it pollinates, the higher is its value for the plant community. It is a really cool measure, but it’s not easy to get one’s head around it. Alternatively, one can see it as the inverse of redundancy of the species for the web: the higher the value, the less redundant.
node specialisation	index Another measure of specialisation, based on the path length between any two higher-trophic level species. Species sharing hosts/prey have an FS-value of 1. See specific function <code>nodespec</code> for details, problems and reference.
Fisher alpha	Fisher’s alpha diversity for each species (see <code>fisher.alpha</code> in <b>vegan</b> for details).
Partner diversity	Shannon diversity (when using ‘ <code>logbase="e"</code> ’) or per-species generality/vulnerability (when using ‘ <code>logbase=2</code> ’) of the interactions of each species. See also <a href="#">networklevel</a> for the aggregated version of this index (i.e. averaged across all species in a trophic level).
effective partners	‘ <code>logbase</code> ’ to the power of “partner.diversity”: Bersier et al. (2002) interpret this as the effective number of partners, if each partner was equally common. Note that “partner” is a bit euphemistic when it comes to predator-prey or host-parasitoid networks.
d	Specialisation of each species based on its discrimination from random selection of partners. More specifically, it returns $d'$ , which is calculated based on the raw $d$ , $d_{min}$ and $d_{max}$ for each species (see <code>dfun</code> . See Blüthgen et al. (2006) for details.

### Note

Dependencies are still an open field of debate. Dependencies are calculated as the value in a matrix divided by the `rowSums` (for the lower trophic level) or the `colSums` (for the higher trophic level). As such, any pollinator observed only once will receive a dependency-value of 1, indicating

perfect dependence on this plant species. That may or may not be true. In any case it is based on a sample size of 1, that is why the dependency asymmetry (which is based on the dependencies for both trophic levels) has come under criticism and may be rather sensitive to singletons.

We here provide the code to calculate the strength of a species (i.e. sum of its dependencies), based on the current proposal by Bascompte et al. (2006). It may be a good idea to remove all singletons from the web before calculating this index, to investigate whether it is indeed driven by those scarce observations.

The maximum value of the uncorrected  $d$  is not a trivial issue. We treat it here in the same way as given in the BMC Ecology paper, but please have a look at the (raw) code for further comments.

### Author(s)

Carsten F. Dormann <carsten.dormann@ufz.de>

### References

Bascompte, J., Jordano, P. and Olesen, J. M. (2006) Asymmetric coevolutionary networks facilitate biodiversity maintenance. *Science* **312**, 431–433

Blüthgen, N., Menzel, F. and Blüthgen, N. (2006) Measuring specialization in species interaction networks. *BMC Ecology* **6**, 9

Vázquez, D. P., Melián, C. J., Williams, N. M., Blüthgen N., Krasnov B. R. and Poulin, R. (2007) Species abundance and asymmetric interaction strength in ecological networks. *Oikos* **116**, 1120–1127

### See Also

[networklevel](#) for some further comments; [dfun](#), [nodespec](#), which are called by this function

### Examples

```
data(Safariland)
specieslevel(Safariland)
specieslevel(Safariland, index="ALLBUTD")[[2]]
```

---

swap.web

*Creates null model for bipartite networks*

---

### Description

Function to generate null model webs under the following constraints: 1. marginal totals are identical to those observed (as in `r2dtable`), 2. connectance is as observed (as in `shuffle.web`.)

### Usage

```
swap.web(N, web, verbose=FALSE, c.crit=1e4)
```

## Arguments

N	Number of desired nullmodel matrices.
web	An interaction matrix.
verbose	Should various verbal outputs of this function be shown? Defaults to FALSE, since it was mainly used during the debugging period.
c.crit	Sometimes the algorithm gets stuck in a very sparse matrix. Then c.crit sets the number of swaps it shall attempt before giving up and starting over on a new random matrix. Defaults to 10000.

## Details

This function is designed to behave similar to `r2dtable`, i.e. it returns a list of randomised matrices. In addition to `r2dtable` is also keeps the connectance constant!

This function is thought of as a very constrained nullmodel for the analysis of bipartite webs. It keeps two web properties constant: The marginal totals (as in `r2dtable` and the number of links (and hence connectance). A comparison of `swap.web`- and `r2dtable`-based webs allows to elucidate the effect of evolutionary specialisation, since the non-realised connections may represent “forbidden links”.

This nullmodel is similar to the one employed by Vazquez et al. But while Vazquez starts by assigning 1s to the allowed connections and then fills the web, `swap.web` starts with an `r2dtable`-web and successively “empties” it. The two approaches should result in very similar nullmodels, since both constrain marginal totals and connectance.

A few words on the way `swap.web` works. It starts with a random web created by `r2dtable`. Then, it finds, randomly, 2x2 submatrices with entries all larger than 0. Next, it subtracts the minimum value from the diagonal and adds it to the off-diagonal (minor diagonal). Thereby one cell becomes 0, but the column and row sums do not change. This idea is adapted from the swap-algorithm used in various binary null models by Nick Gotelli. If the random web has too few 0s (which is I have yet to encounter), then the opposite strategy is applied.

The algorithm in our implementation has some variations on finding the submatrix and constraining the number of successless trials before starting on a new random matrix, but they are only for speeding up the process.

## Value

A list of N randomised matrices with the same dimensions as the initial web.

## Note

Long stories can be told about the swap algorithm. I am not the right person to do so, but for a much more detailed coverage of the topic, for many more ways to implement null models for **binary** matrices, with various flavours of the swap and possible alternatives, first brew yourself a cup of tea and then check out the help pages of `commsimulator` in **vegan**. (As usual, Jari Oksanen has spend considerable care to implement even the most bizarr and abstruse way to move 0s and 1s around. His ecological advise between the lines make his package worthwhile already! I, personally, would use `method="quasiswap"`, as is done in the example to [discrepancy](#).)

`swap.web` is a very constraint nullmodel. You need to consider if it is the right one for your application!

**Author(s)**

Carsten F. Dormann <carsten.dormann@ufz.de>

**References**

Vázquez, D. P., and M. A. Aizen (2003) Null model analyses of specialization in plant-pollinator interactions. *Ecology* **84** 2493-2501

Vázquez, D. P., C. J. Melián, N. M. Williams, N. Blüthgen, B. R. Krasnov, and R. Poulin (2007) Species abundance and asymmetric interaction strength in ecological networks. *Oikos* **116**, 1120–1127

For a very nice and thorough overview of nullmodels in general see:

Gotelli, N. J., and G. R. Graves (1996) *Null Models in Ecology*. Smithsonian Institution Press, Washington D.C.

**See Also**

[r2dtable](#) and [shuffle.web](#)

**Examples**

```
data(Safariland)

swap.web(Safariland, N=2)
```

---

togetherness	<i>Calculates the number of identical co-presences and co-absences for species-on-islands</i>
--------------	---

---

**Description**

Togetherness, or T-score, describes the level of similarity in the distributional pattern of two species. Originally proposed by Stone & Roberts (1992) for biogeographical situations can it also be applied e.g. to pollinators on different host plants.

**Usage**

```
togetherness(web, normalise=TRUE, FUN = mean, ...)
```

**Arguments**

web	A matrix with binary or counted interactions/links, where the higher trophic level is represented by columns.
normalise	Logical; shall index be normalised to a range of 0-1?
FUN	The function to summarise species-pair T-scores with; defaults to mean.
...	Arguments passed on to FUN, especially <code>na.rm=T</code> or colours for <code>hist</code> .

**Value**

Returns the average (default) togetherness of all species combinations.

**Author(s)**

Carsten F. Dormann

**References**

Stone, L. and Roberts, A. (1992) Competitive exclusion, or species aggregation? An aid in deciding. *Oecologia* **91**, 419–424

**See Also**

[C.score](#) for another of Stone & Roberts' indices.

**Examples**

```
(m <- matrix(c(0,1,0,0,1,1,0,1,1,0), ncol=2, byrow=TRUE))
togetherness(m)
# or, with two togethernesses:
(n <- matrix(c(0,1,1,0,1,1,0,0,1,1,0,1,0,1), ncol=2, byrow=TRUE))
togetherness(n, normalise=FALSE)

data(Safariland)
togetherness(m)
```

---

V.ratio

*Calculates the variance-ratio as suggested by Schluter (1984)*

---

**Description**

A of species association is provided by the ratio of the variance in total species number (or total density of individuals) in samples to the sum of the variances of the individual species.

**Usage**

```
V.ratio(web)
```

**Arguments**

`web` A matrix with pollinators in columns and plants in rows. For biogeographical applications: rows are islands (or sites).

**Details**

This is a rather straight-forward index, which is described and evaluated extensively in Schluter (1984). He also warns against overinterpretation of the value. In principle, V-ratios larger than 1 indicate positive, smaller than 1 negative associations. Ecologically, competition can lead to small or large values, depending on their exact effects (see discussion in the Schluter paper).

**Value**

Returns the V-ratio, i.e.~a single value representing the ratio of variance in species number and variance in individual numbers within species.

**Note**

Any quantitative matrix is first transformed into a binary (presence-absence) matrix!

Do not interpret without first reading the paper! It's worth it! See also applications in other studies, such as Gotelli and Rohde (2002).

**Author(s)**

Carsten F. Dormann

**References**

Gotelli, N.J. and Rohde, K. (2002) Co-occurrence of ectoparasites of marine fishes: a null model analysis. *Ecology Letters* **5**, 86–94

Schluter, D. (1984) A variance test for detecting species associations, with some example applications. *Ecology* **65**, 998–1005

**See Also**

[C.score](#) for another measure of species associations.

**Examples**

```
data(Safariland)
V.ratio(Safariland)
```

---

vazarr

*A pollination network.*

---

**Description**

[See documentation for Safariland.]

**Usage**

```
data(vazarr)
```

**Format**

The format is: num [1:10, 1:29] 85 0 94 0 0 0 0 0 0 ... - attr(\*, "dimnames")=List of 2 ..\$ : chr [1:10] "X2" "X1" "X13" "X4" ... ..\$ : chr [1:29] "78" "65" "94" "66" ...

**Examples**

```
data(vazarr)
## maybe str(vazarr) ; plot(vazarr) ...
```

---

vazcer	<i>A pollination network.</i>
--------	-------------------------------

---

**Description**

[See documentation for Safariland.]

**Usage**

```
data(vazcer)
```

**Format**

The format is: num [1:9, 1:33] 59 0 110 0 0 0 0 0 0 ... - attr(\*, "dimnames")=List of 2 ..\$ : chr [1:9] "X2" "X1" "X13" "X4" ... ..\$ : chr [1:33] "78" "65" "94" "32" ...

**Examples**

```
data(vazcer)
## maybe str(vazcer) ; plot(vazcer) ...
```

---

vazllao	<i>A pollination network.</i>
---------	-------------------------------

---

**Description**

[See documentation for Safariland.]

**Usage**

```
data(vazllao)
```

**Format**

The format is: num [1:10, 1:29] 64 0 82 0 0 0 0 0 0 ... - attr(\*, "dimnames")=List of 2 ..\$ : chr [1:10] "X2" "X1" "X13" "X4" ... ..\$ : chr [1:29] "78" "65" "94" "32" ...

**Examples**

```
data(vazllao)
## maybe str(vazllao) ; plot(vazllao) ...
```

---

vazmasc

*A pollination network.*


---

**Description**

[See documentation for Safariland.]

**Usage**

```
data(vazmasc)
```

**Format**

The format is: num [1:8, 1:26] 73 0 0 0 0 0 0 0 6 ... - attr(\*, "dimnames")=List of 2 ..\$ : chr [1:8] "X2" "X1" "X13" "X4" ... ..\$ : chr [1:26] "78" "65" "94" "32" ...

**Examples**

```
data(vazmasc)
## maybe str(vazmasc) ; plot(vazmasc) ...
```

---

vazmasnc

*A pollination network.*


---

**Description**

[See documentation for Safariland.]

**Usage**

```
data(vazmasnc)
```

**Format**

The format is: num [1:8, 1:35] 26 0 0 0 0 0 0 0 123 ... - attr(\*, "dimnames")=List of 2 ..\$ : chr [1:8] "X2" "X1" "X13" "X4" ... ..\$ : chr [1:35] "78" "65" "94" "32" ...

**Examples**

```
data(vazmasnc)
## maybe str(vazmasnc) ; plot(vazmasnc) ...
```

---

vazquenc

*A pollination network.*


---

**Description**

[See documentation for Safariland.]

**Usage**

```
data(vazquenc)
```

**Format**

The format is: num [1:8, 1:27] 31 0 34 150 11 66 2 4 0 63 ... - attr(\*, "dimnames")=List of 2 ..\$ : chr [1:8] "X1" "X13" "X4" "X15" ... ..\$ : chr [1:27] "65" "94" "32" "66" ...

**Examples**

```
data(vazquenc)
## maybe str(vazquenc) ; plot(vazquenc) ...
```

---

vazquenc

*A pollination network.*


---

**Description**

[See documentation for Safariland.]

**Usage**

```
data(vazquenc)
```

**Format**

The format is: num [1:7, 1:24] 89 0 404 0 0 0 0 147 0 ... - attr(\*, "dimnames")=List of 2 ..\$ : chr [1:7] "X2" "X1" "X13" "X15" ... ..\$ : chr [1:24] "78" "65" "32" "66" ...

**Examples**

```
data(vazquenc)
## maybe str(vazquenc) ; plot(vazquenc) ...
```

## Description

Describes how to use `bipartite` to calculate the statistics presented in Vazquez et al. (2009). Some of these functions are available in `bipartite` or other packages, and this help page will show how to use them in line with the publication.

## Details

The functions used are:

**confint:** Is the same as `quantile`

**intasymm:** Can be extracted using `specieslevel`

**intereven:** Is similar to interaction evenness in `networklevel`, but only for a specific option

**mgen:** Random web based on the number of links; it usually loses ranks (i.e. not all species will still be represented in this random web), and it is based only on the binary web (i.e. all quantitative information is lost); see `r2dtable`, `swap.web` and `shuffle.web` for quantitative alternatives.

**mlik:** A specific call to `dmultinom` and the calculation of the AIC; the number of parameters entering the AIC-calculation is not obvious; this depends on the constraints used by the null model. In the case of `r2dtable`, column and row totals are constrained, i.e.  $n_{col} + n_{row}$  parameters must be given. In the case of `swap.web`, connectance is also constrained, but how many parameters does that imply? One? In `shuffle.web`, we constrain the dimensionality and connectance, i.e. 3 (?) parameters. Vazquez et al. (2009) argue that they constrain only 2 parameters when producing the probability matrix given as `pweb` in the example below. We tend to disagree: the marginal probabilities of all columns and rows are given, hence  $k = (n_{col}(\text{web}) + n_{row}(\text{web}))$ . To our knowledge, there is no mathematical/statistical treatise of this problem.

**netstats:** A wrapper calling the other functions, in that sense similar to `networklevel`, but also calling some output from `specieslevel`.

**plotmat:** Now part of `visweb`, using the right options.

**quant2bin:** A dedicated function to do a simple thing:  $(\text{web} > 0) * 1$ .

**sortmatr:** newly defined function: `sortweb`

**sortmatrext:** sort matrix by some given sequence; also part of `sortweb`

In the example below, we use the **bipartite**/standard R functions whenever possible.

## Author(s)

Carsten F. Dormann <carsten.dormann@ufz.de> based on code and ideas of Diego Vazquez, Nat-  
acha P. Chacoff and Luciano Cagnolo

## References

Vazquez, P.D., Chacoff, N.,P. and Cagnolo, L. (2009) Evaluating multiple determinants of the structure of plant-animal mutualistic networks. *Ecology* in press.

## See Also

See also [networklevel](#).

## Examples

```
data(Safariland)

# confint:
N100 <- sapply(swap.web(100, Safariland), networklevel, index="nestedness")
quantile(unlist(N100), c(0.025, 0.975))
# intasymm: extract values for the asymmetry of interactions and the
# dependency matrix for pollinators:
specieslevel(Safariland)$"higher trophic level"$"interaction push/pull"
specieslevel(Safariland)$"higher trophic level"$"dependence"
# for plants:
specieslevel(Safariland)$"lower trophic level"$"interaction push/pull"
specieslevel(Safariland)$"lower trophic level"$"dependence"

#intereven
networklevel(Safariland, index="interaction evenness", intereven="sum")[2]
# or, as we recommend (see help on networklevel):
networklevel(Safariland, index="interaction evenness", intereven="prod")[2]

# mgen:
binweb <- Safariland>0 #throw away the information on the number of visits
# make a matrix with probabilities for each link, based on column and row totals:
pweb <- outer(rowSums(binweb)/sum(binweb), colSums(binweb)/sum(binweb), FUN="*")
# make a new, empty matrix:
rbinweb <- matrix(0, nrow=nrow(binweb), ncol=ncol(binweb))
# put the links into random places, with probability as given by the observed data:
rbinweb[sample(1:prod(dim(binweb)), size=sum(binweb), prob=pweb)] <- 1
# this is the new, random realisation given the observed marginal link sums:
rbinweb
# for this null-web any of the networklevel indices can be calculated

# mlik:
# calculates the log-likelihood of observing a network, given a probability
# matrix of the same size (pweb):
dmultinom(Safariland>0, prob=pweb, log=TRUE)
# AIC (the number of parameters is given by how many constraints are put onto the
# null model; here, we constrain 9 rows and 27 columns, i.e. sum(dim(binweb))):
-2*dmultinom(Safariland>0, prob=pweb, log=TRUE) + 2*(sum(dim(binweb)))

# netstats:
networklevel(Safariland,
  index=c("connectance", "interaction evenness", "nestedness", "ISA"))
mean(specieslevel(Safariland)$"higher trophic level"$"interaction push/pull")
```

```

mean(specieslevel(Safariland)$"lower trophic level"$"interaction push/pull")

#plotmat:
visweb(t(unnname(Safariland)), circles=TRUE, boxes=FALSE)

#sortmatr/sortmatrtext:
sortweb(Safariland, sort.order="inc") #rarest species first
plotweb(sortweb(Safariland, sort.order="dec"), method="normal")
plotweb(sortweb(web=Safariland, sort.order="seq",
  sequence=list(seq.higher=sample(colnames(Safariland)),
  seq.lower=sample(rownames(Safariland))),
  method="normal")

```

visweb

*Plotfunction to visualize a bipartite foodweb***Description**

this function draws a foodweb as a grid using a matrix. colnames and rownames are used as labels and entries in the matrix are visualized by text and colours. now it can be used to plot bipartite webs in a Vazquez et al. 2009 style

**Usage**

```

visweb(web, type="nested", prednames=TRUE, preynames=TRUE, labsize=1,
  plotsize=12, square="interaction", text="no", frame=NULL, textsize=1,
  textcol="red", pred.lablenth=NULL, prey.lablenth, clear=TRUE,
  xlabel="", ylabel="", boxes=TRUE, circles=FALSE, circle.col="black", circle.min=0,
  circle.max=2, outerbox.border="white", outerbox.col="white",
  box.border="black", box.col="black")

```

**Arguments**

web	A matrix representing the interactions observed between higher trophic level species (columns) and lower trophic level species (rows). Usually this will be number of pollinators on each species of plants or number of parasitoids on each species of prey.
type	type changes the sorting of rows and columns of the web and can be <b>nested</b> : (sorted by row/colSums) <b>diagonal</b> : (highest number of interactions appear along the diagonal, good for showing compartments) <b>none</b> : (as is)
prednames	labels can be switched of by prednames=F
preynames	labels can be switched of by preynames=F
labsize	factor for size of labels

plotsize	size of plot (length of width or height), depending on the dimension of the web in cm, default is 12 cm.
square	square is used to indicate number of interactions and belonging to compartments by coloured grid cells <b>interaction:</b> (level of grey indicates the number of intercation, white means no interaction) <b>compartment:</b> (level of grey indicates belonging to the same compartment) <b>black :</b> (black grid cells if number of interaction is bigger than one) <b>none :</b> (no coloured squares are drawn)
text	number of interactions or belonging are plotted into each grid cell <b>interaction:</b> (number of interactions are drawn) <b>compartment:</b> (belonging to same compartment indicated by capitel Letters) <b>none :</b> (no text is shown)
frame	a frame is drawn around each compartment (frame=TRUE), best used with type="diagonal"
textsize	factor for size of text in squares, default=1
textcol	color of text in grid cells, default ="red"
pred.lablength	length of predators (upper) labels that should be displayed
prey.lablength	length of prey (lower) labels that should be displayed
clear	delete species with no interactions (compulsory done for "nested" and "diagonal")
xlabel	label on the x-axis, make sure prey.lablength is set accordingly, default is empty
ylabel	label on the y-axis, make sure pred.lablength is set accordingly, default is empty
boxes	logical, if boxes should be drawn. Default is set to TRUE
circles	logical, if circles in a Vazquez et al. style should be drawn. Default is set to FALSE, size and colours of circles and background can be set by the following arguments
circle.col	Colour of circles, works only if circles=TRUE
circle.min	minimal size of circles, use to rescale circles appropriately, default is 0.2
circle.max	maximal size of circle, , use to rescale circles appropriately, default is 2
outerbox.border	Colour of outerbox border if option circles=TRUE
outerbox.col	Colour of background if option circles=TRUE
box.border	Colour of border of boxes, if option square="b" is set
box.col	Colour of boxes, if option square="b" is set

### Value

A plot window with appropriate size according to the dimensions of the web.

**Author(s)**

Bernd Gruber

**References**

Vazquez, P.D., Chacoff, N.,P. and Cagnolo, L. (in press, 2009). Evaluating multiple determinants of the structure of plant-animal mutualistic networks. *Ecology*.

**See Also**

For a different plot on foodweb see [plotweb](#)

**Examples**

```
data(Safariland)
visweb(Safariland)
visweb(Safariland, type="diagonal", square="compartment", text="none", frame=TRUE)
visweb(Safariland, type="nested", text="compartment")

visweb(Safariland, circles=TRUE, boxes=FALSE, labsize=1, circle.max=3, text="no")
visweb(Safariland, circles=TRUE, boxes=FALSE, labsize=1, text="no", circle.max=1.8, outerb
visweb(Safariland, circles=TRUE, boxes=TRUE, outerbox.col="orange", labsize=1, circle.max=1

visweb(Safariland, square="b", box.col="green", box.border="red")
```

---

wine

*Weighted-Interaction Nestedness Estimator*

---

**Description**

Calculates the nestedness of a network taking into account the weight of the interactions, according to the method proposed by Galeano et al. (2008).

**Usage**

```
wine(web, nreps = 1)
## S3 method for class 'wine':
plot(x, ...)
```

**Arguments**

web	A matrix with elements of a set (e.g., plants) as rows, elements of a second set (e.g., pollinators) as columns and number of interactions as entries.
nreps	Number of replicates for constructing random networks.
x	An object resulting of applying wine function
...	Additional graphical parameters to <a href="#">image.plot</a>

## Details

Nestedness estimators use presence-absence (binary) adjacency matrices as the basis for calculating nestedness, as they provide a simple description and characterization of the topology of the network. However, networks are specified not only by their topology but also by the heterogeneity in the weight (or the intensity) of the connections (Barrat et al., 2004). Characterizing links just with presence-absence data does not take into account the possible differences in intensity among links. WINE (Weighted-Interaction Nestedness Estimator) is a new nestedness estimator that takes into account the weight or intensity of each interaction (e.g., in a plant-pollinator network, the number of registered visits of a particular interaction). Thus, instead of using presence-absence matrices, WINE calculates nestedness from quantitative data matrices that include the number of events of each interaction. This is the first estimator that allows for the characterization of weighted nestedness. WINE calculates a nestedness value that approaches zero when the nestedness pattern of the original data matrix is close that of equivalent random matrices, and it approaches one as it gets closer to the nestedness of the maximal nestedness matrix. Thus, this estimator evaluates the relative position of the data matrix between the corresponding random matrices and the maximal nestedness matrix. Negative values for this estimator can be found in some synthetic matrices that have been described as 'anti-nestedness' matrices (Almeida-Neto et al. 2007).

The calculation of the weighted-interaction nestedness estimator starts with the matrix containing the number of events of each interaction,  $M_{ij}$ . The matrix is packed by arranging rows and columns from top to bottom and from left to right, respectively, in ascending order according to their marginal totals. Nestedness is related to the proximity of existing links to one another in the packed matrix, so that the most nested matrix is the one that after packing shows a minimum mixing of filled cells (links) with empty cells (no links) (Corso et al., 2008, Ulrich et al., 2009). WINE is based on the concept of estimating nestedness through the calculation of a Manhattan distance from each of the matrix cells containing a link to the cell corresponding to the intersection of the row and columns with the lowest marginal totals (number of links). This concept resembles in a way the one used by Corso et al. (2008), although the distances are measured to the opposite corner of the packed matrix. Additionally, in WINE, the Manhattan distance is replaced by a weighted Manhattan distance. The statistical significance of this nestedness index value is tested against a null model that constrains matrix fill to observed values, retains the distribution of number of events in the links but does not constrain marginal totals. Further details can be found in Galeano et al. (2008).

## Value

wine returns an object of class `wine`, basically a list with the following components:

<code>win</code>	Weighted-interaction nestedness of dataset (WIN)
<code>wine</code>	Weighted-interaction nestedness estimator (WINE): The weighted-interaction nestedness estimator value. It will be 0 for random distribution and 1 for maximal nestedness
<code>zscore</code>	z-score of the weighted-interaction nestedness
<code>pvalue</code>	probability of having a z-value equal to or greater than Z (from the tabulated value of the cumulative function). Values of $p < 0.05$ indicate that the dataset is significantly nested.
<code>dmax</code>	Weighted-interaction nestedness of the maximal nestedness matrix.
<code>drnd</code>	Average weighted-interaction nestedness of random replicates

<code>dij.w</code>	Matrix of <code>dijw</code> values. These values provide a measure of the contribution of each interaction (link) to total nestedness
<code>dij.max</code>	Maximal nestedness matrix

The S3 plot method for wine displays `dij.w` in a coloured image plot where red cells have high weights in the network and blue cells have minimum weights.

### Note

This is the first approach to a weighted nestedness and a full ecological interpretation of its meaning is still lacking. It is not possible to perform a systematic comparison between this and other nestedness indices because the latter rely just on presence absence data whereas the former feeds on a quantitative data matrix. For a well-performed comparison of other nestedness indices see Ulrich & Gotelli (2007).

`wine` may return NaN for different parameters essentially for two different reasons: a) if 'nreps' is not specified, `wine` adopts `nreps=1` by default and NaN is returned for z-score and p value. This is due to the fact that with `nreps=1` the variance of `drnd` is zero and z-score becomes infinite. The same outcome may occur in some cases with very low values of `nreps`. To ensure proper values of z-score and p-values `nreps=100` or higher is suggested. b) if `dw = drnd = dmax` `wine` equals 0/0, and if `drnd = dmax` `wine` tends to infinity. In both cases, NaN is returned by `wine`. This is more likely to occur in cases where the dimensions of the matrix are very low (e.g, `(dim < c(4, 4))`) because in those cases the number of possible values of `dw`, `drnd` and `dmax` is also reduced.

This is WINE version 1.1.

### Author(s)

Marcelino de la Cruz <marcelino.delacruz@upm.es>, Juan M. Pastor <juanmanuel.pastor@upm.es>, Javier Galeano <javier.galeano@upm.es> and Jose M. Iriondo <jose.iriondo@urjc.es>

### References

- Barrat, A., Barthélemy, M., Pastor-Satorras, R., and Vespignani, A. (2004) The architecture of complex weighted networks. *PNAS* 101, 3747-3752
- Corso G, de Araújo AIL, de Almeida AM (2008) A new nestedness estimator in community networks. *arXiv* 0803.0007v1 [physics.bio-ph]
- Galeano J, Pastor JM, Iriondo JM (2008) Weighted-Interaction Nestedness Estimator (WINE): A new estimator to calculate over frequency matrices. *arXiv* 0808.3397v2 [physics.bio-ph]
- Ulrich, W., Almeida-Neto, M., and Gotelli, N.J. (2009) A consumer's guide to nestedness analysis. *Oikos* 118, 3-17
- Ulrich, W. and Gotelli, N.J. (2007) Null model analysis of species nestedness patterns. *Ecology* 88, 1824-1831

### See Also

[nestedness](#) and [discrepancy](#).

**Examples**

```
data(Safariland, package="bipartite")
safariland.w <- wine(Safariland, 100)
plot.wine(safariland.w)
```

# Index

## \*Topic **datasets**

- barrett1987, 9
- elberling1999, 18
- inouye1988, 25
- kato1990, 26
- kevan1970, 27
- memmott1999, 28
- mosquin1967, 28
- motten1982, 29
- olesen2002aigrettes, 45
- olesen2002flores, 46
- ollerton2003, 47
- Safariland, 57
- schemske1978, 58
- small1976, 63
- vazarr, 72
- vazcer, 73
- vazllao, 73
- vazmasc, 74
- vazmasnc, 74
- vazquec, 75
- vazquenc, 75

## \*Topic **htest**

- degreedistr, 13
- dfun, 14

## \*Topic **package**

- as.one.mode, 8
- bipartite-package, 2
- C.score, 10
- compart, 11
- discrepancy, 17
- empty, 19
- extinction, 20
- functspec, 21
- genweb, 23
- H2fun, 24
- nested, 30
- nestedness, 32
- networklevel, 34

- nodespec, 41
- null.t.test, 43
- nullmodel, 44
- PAC, 48
- plotweb, 49
- plotweb2, 53
- robustness, 56
- second.extinct, 59
- shuffle.web, 60
- slope.bipartite, 62
- sortweb, 64
- specieslevel, 65
- swap.web, 68
- togetherness, 70
- V.ratio, 71
- vazquez.example, 76
- visweb, 78
- wine, 80

as.one.mode, 7, 8

- barrett1987, 9
- bipartite (*bipartite-package*), 2
- bipartite-package, 2

- C.score, 4, 10, 31, 41, 71, 72
- compart, 5, 11
- confint (*vazquez.example*), 76

- degreedistr, 4, 13, 37, 41
- dfun, 2, 3, 14, 22, 24, 25, 35, 38, 42, 66–68
- discrepancy, 3, 6, 17, 31, 38, 69, 82

- elberling1999, 18
- empty, 4, 19, 59, 60
- extinction, 20, 20, 59, 60

functspec, 6, 7, 9, 21

genweb, 23

- H2fun, 2, 3, 6, 15, 16, 24, 39, 41
- image.plot, 80
- inouye1988, 5, 25
- intasymm (vazquez.example), 76
- intereven (vazquez.example), 76
- kato1990, 26
- kevan1970, 27
- memmott1999, 28
- mgen, 45
- mgen (vazquez.example), 76
- mlik (vazquez.example), 76
- mosquin1967, 28
- motten1982, 29
- nested, 3, 30
- nestedness, 5, 18, 30, 31, 32, 38, 82
- netstats (vazquez.example), 76
- networklevel, 2–7, 12, 14, 34, 43, 59, 60, 67, 68, 76, 77
- nodespec, 5, 41, 67, 68
- null.t.test, 43
- nullmodel, 4, 44
- olesen2002aigrettes, 45
- olesen2002flores, 46
- ollerton2003, 5, 47
- PAC, 3, 48
- plot.wine, 4
- plot.wine (wine), 80
- plotmat (vazquez.example), 76
- plotweb, 2, 3, 5, 7, 49, 55, 65, 80
- plotweb2, 7, 53
- quant2bin (vazquez.example), 76
- r2dtable, 43, 61, 70
- robustness, 4, 5, 38, 56
- Safariland, 57
- schemske1978, 58
- second.extinct, 2, 20, 21, 41, 56, 57, 59, 62, 63
- shuffle.web, 6, 45, 60, 70, 76
- slope.bipartite, 2, 4, 56, 57, 60, 62
- small1976, 63
- sortmatr (vazquez.example), 76
- sortmatrext (vazquez.example), 76
- sortweb, 5, 64, 76
- specieslevel, 2, 3, 16, 22, 42, 65, 76
- swap.web, 45, 68, 76
- t.test, 43
- togetherness, 70
- V.ratio, 2, 11, 41, 71
- vazarr, 72
- vazcer, 73
- vazllao, 73
- vazmasc, 74
- vazmasnc, 74
- vazquec, 75
- vazquenc, 75
- vazquez.example, 76
- visweb, 2, 4, 5, 37, 52, 55, 65, 76, 78
- wine, 4, 18, 31, 38, 80