

Package ‘indicspecies’

February 4, 2020

Type Package

Title Relationship Between Species and Groups of Sites

Version 1.7.9

Date 2020-02-03

Description

Functions to assess the strength and statistical significance of the relationship between species occurrence/abundance and groups of sites [De Cáceres & Legendre (2009) <doi:10.1890/08-1823.1>]. Also includes functions to measure species niche breadth using resource categories [De Cáceres et al. (2011) <doi:10.1111/J.1600-0706.2011.19679.x>].

Depends permute

Suggests sp, rgeos

Encoding UTF-8

License GPL (>= 2)

URL <https://vegmod.github.io/software/indicspecies>

LazyLoad yes

NeedsCompilation no

Author Miquel De Cáceres [aut, cre],
Florian Jansen [aut],
Noah Dell [aut]

Maintainer Miquel De Cáceres <miquelcaceres@gmail.com>

Repository CRAN

Date/Publication 2020-02-04 11:10:02 UTC

R topics documented:

indicspecies-package	2
birds	3
combinespecies	4
coverage	5
indicators	8

multipatt	11
nicheoverlap	13
nichevar	16
pigeons	19
plotniche	19
predict.indicators	21
pruneindicators	23
signassoc	25
strassoc	27
summary.multipatt	29
wetland	30

Index	32
--------------	-----------

indicspecies-package *Studying the statistical relationship between species and groups of sites*

Description

This package provides a set of functions to assess the strength and statistical significance of the relationship between species occurrence/abundance and groups of sites. It is also possible to check the statistical significance of such associations. New functions have been added (previously distributed in 'resniche') which allow evaluate several niche metrics while accounting for distances between resources on the basis of the similar perception from species (De Cáceres et al. 2011).

Details

Package:	indicspecies
Type:	Package
Version:	1.7.7
Date:	2019-05-20
Depends:	permute
Suggests:	sp, rgeos
License:	GPL (>=2)
LazyLoad:	yes

Use [strassoc](#) for assessing strength of associations and [signassoc](#) to assess the statistical significance of the associations (De Cáceres & Legendre 2009). Much more complete functions are [multipatt](#) (De Cáceres et al. 2010) and [indicators](#) (De Cáceres et al. 2012). Read the package vignette using `vignette("indicspeciesTutorial", package="indicspecies")` to learn how to use this package.

Author(s)

Miquel De Cáceres, Florian Jansen

Maintainer: Miquel De Cáceres <miquelcaceres@gmail.com>

References

De Cáceres, M. and Legendre, P. 2009. Associations between species and groups of sites: indices and statistical inference. *Ecology* 90(12): 3566-3574.

De Cáceres, M., Legendre, P., Moretti, M. 2010. Improving indicator species analysis by combining groups of sites. *Oikos* 119(10): 1674-1684.

De Cáceres, M., Legendre, P., Wisser, S.K. and Brotons, L. 2012. Using species combinations in indicator analyses. *Methods in Ecology and Evolution* 3(6): 973-982.

De Cáceres, M., Sol, D., Lapiedra, O. and P. Legendre. (2011) A framework for estimating niche metrics using the resemblance between qualitative resources. *Oikos* 120: 1341-1350.

birds

Birds resource use data

Description

The resource use data for a set of 10 hypothetical bird species during two different seasons. There are five resource classes, and their relationships are given in matrix resourceD.

Usage

```
data(birds)
```

Format

birdsbreed - a data frame containing the resource use of 10 birds for a set of five resources during the breeding season.

birdswinter - a data frame containing the resource use of 10 birds for a set of five resources during the winter season.

resourceD - a distance matrix (object of class `dist`) containing the relationships between the five resources.

Examples

```
data(birds)
```

combinespecies *Combines species from a community table*

Description

Creates species combinations to be used in indicator value analyses.

Usage

```
combinespecies(X, min.order = 1, max.order = 3, min.occ = 1, FUN = min, verbose = FALSE,
  add.names = TRUE, ...)
```

Arguments

X	A community data table with sites in rows and species in columns. This table can contain either presence-absence or abundance data.
min.order, max.order	Minimum and maximum number of species conforming species combinations.
min.occ	Threshold for minimum occurrence. Combinations with lower values are not kept.
FUN	Function to be calculated for all species forming the species combination. Accepted values are min, max, sum and mean.
verbose	If TRUE, prints the results of each step.
add.names	If TRUE, adds the names of the species combinations to the output matrix. Species combination names are lists of species concatenated using character '+'.
...	Additional arguments for function FUN.

Details

This function allows creating a data table where rows are sites and columns are combinations of species. Values for a given column of this matrix are derived from the abundance values of the species forming the corresponding combination. In particular, the abundance value for a given combination in a given site is equal to the value returned by function 'FUN' (normally the minimum) among the site values of all species forming the combination. The matrix 'XC' returned by this function can be used in functions [strassoc](#) and [signassoc](#). Alternatively, [indicators](#) and related functions provide a more elaborated way to explore the indicator value of the simultaneous occurrence of sets of species (i.e. species combinations).

Value

An list with:

XC	A matrix containing the abundance/occurrence of each species combination.
C	A binary matrix describing the set of species forming each combination ('0' means that the species is not included, and '1' means that the species is included).

Author(s)

Miquel De Cáceres Ainsa, CTFC

References

De Cáceres, M., Legendre, P., Wisser, S.K. & Brotons, L (in preparation). Using species combinations in indicator analyses.

See Also

[indicators](#), [strassoc](#), [signassoc](#)

Examples

```
library(stats)

data(wetland) ## Loads species data

## Create species combinations
Y=combinespecies(X=wetland, max.order=3, min.occ=5, verbose=TRUE)

## Creates three site groups using kmeans
wetkm = kmeans(wetland, centers=3)

## Calculate indicator value of species combinations for each of the three site groups
strassoc(Y$XC, cluster=wetkm$cluster,func="IndVal.g")

## Calculate point biserial correlation value of species combinations
## for each of the three site groups
strassoc(Y$XC, cluster=wetkm$cluster,func="r.g")
```

coverage

Coverage of a set of indicators

Description

Function `coverage` calculates the proportion of sites of the target site group where one or another indicator (a species or a species combination) is found. Parameters are used to select valid indicators before calculating coverage. Function `plotcoverage` plots the coverage against the positive predictive value threshold used to select indicators.

Usage

```
coverage(x, y=NULL, selection=NULL, minstat=NULL, At=NULL, Bt=NULL, type="stat",
alpha=NULL)
plotcoverage(x, y=NULL, by=0.05, type="stat", max.order=NULL, group = NULL,
alpha = NULL, add=FALSE, xlab=expression(A[t]), ...)
```

Arguments

x	An object of class <code>'indicators'</code> or a community data table with sites in rows and species in columns (in this second case, an object of class <code>'multipatt'</code> must be supplied for y).
y	An object of class <code>'multipatt'</code> .
selection	A logical vector restricting the set of indicators used to calculate the coverage.
minstat	Minimum value of the statistic for selecting indicators.
At	Minimum value of positive predictive value (A) for selecting indicators.
Bt	Minimum value for sensitivity (B) for selecting indicators.
alpha	Significance level for selecting indicators.
type	Specifies how to select indicators: (1) using the value of the statistic (type = "stat"); (2) the lower bound of its confidence interval (type = "lowerCI"); or (3) the upper bound of its confidence interval (type = "upperCI"). This parameter makes sense when the function is called using objects of class <code>'indicators'</code> and bootstrap confidence intervals are available for this object. Otherwise type has no effect and the value of the statistic is used for selection. In function <code>coverage</code> , the value of type applies to selection using <code>minstat</code> , <code>At</code> and <code>Bt</code> . In function <code>plotcoverage</code> , the value of type applies to selection using <code>At</code> .
by	Rate of increase in the predictive value threshold (<code>At</code>).
max.order	The maximum number of species conforming species combinations (allows examining the effects of increasing the order of combinations). This parameter is only used when the function is called using objects of class <code>'indicators'</code> .
group	Either an integer or a character string indicating the site group or site group combination for which plot is desired. This parameter is only used when the function is called using objects of class <code>'multipatt'</code> .
add	Flag indicating whether the function should draw on the existing plot.
xlab	Label for the x-axis.
...	Additional plotting parameters that are passed to the <code>plot</code> function.

Details

The coverage of a set of indicators was defined in De Cáceres et al. (2012) as the proportion of sites in a given site group where one or several indicators are found. This value allows assessing how often the site group will be able to be determined. If all indicators of a site group are rare, then the indication system will not be useful, regardless of how much restricted to the site group the indicators are. The coverage value is a generalization of quantity B of `IndVal`, that applies to a group of indicators instead of a single one. Function `plotcoverage` plots the coverage against the positive predictive value threshold (`At`) used to select indicators, as in De Cáceres et al. (2012). Functions `coverage` and `plotcoverage` can be executed using either an object of class `'indicators'`, or an object of class `'multipatt'`. However, the parameters that apply to each case are slightly different. When using `coverage` and `plotcoverage` on objects of class `'multipatt'` one is expected to calculate the coverage for those indicators that are significant (see `alpha` parameter), although other constraints to select valid indicators can be used. When using `coverage` and `plotcoverage` on objects of class `'indicators'` one is expected to calculate the coverage for indicators that have values

of A larger than a specified threshold (see *At* parameter). In this latter case, it may be advisable to use *stat="lowerCI"*, so that indicators with broad confidence intervals are not included in the selection.

Value

When used with an object of class 'indicators', function *coverage* returns the proportion of sites of the target site group where one or another indicator (species combination) is found. When used with an object of class 'indicators', function *coverage* returns a vector containing the coverage value for each site group or site group combination.

Author(s)

Miquel De Cáceres Ainsa, CTFC

References

De Cáceres, M., Legendre, P., Wisser, S.K. and Brotons, L. 2012. Using species combinations in indicator analyses. *Methods in Ecology and Evolution* 3(6): 973-982.

See Also

[indicators](#), [multipatt](#), [pruneindicators](#)

Examples

```
library(stats)

data(wetland) ## Loads species data

## Creates three clusters using kmeans
wetkm = kmeans(wetland, centers=3)

## Run indicator analysis with species combinations for the first group
sc= indicators(X=wetland, cluster=wetkm$cluster, group=1, verbose=TRUE, At=0.5, Bt=0.2)

## Determine the coverage of the selected set of indicators
coverage(sc)

## Plot the coverage against the threshold At
plotcoverage(sc)
plotcoverage(sc, max.order=2, add=TRUE, lty=2)

## Runs the combination analysis using IndVal.g as statistic
wetpt = multipatt(wetland, wetkm$cluster, control = how(nperm=999))
## Determines the coverage for each site group combination
coverage(wetland, wetpt, alpha = 0.05)
```

indicators

*Indicator analysis for species combinations***Description**

Determines the indicator value of species combinations.

Usage

```
indicators(X, cluster, group, func="IndVal", min.order = 1, max.order=5,
           max.indicators=NULL, At=0, Bt=0, sqrtIVt=0,
           control = how(), permutations = NULL, print.perm = FALSE,
           nboot.ci=NULL, alpha.ci=0.05, XC=TRUE, enableFixed = FALSE,
           verbose = FALSE)
## S3 method for class 'indicators'
print(x, At=0, Bt=0, sqrtIVt=0, alpha = 1.0, selection=NULL, confint=FALSE,...)
## S3 method for class 'indicators'
plot(x, type="sqrtIV", maxline=TRUE,...)
## S3 method for class 'indicators'
summary(object,...)
```

Arguments

X	A community data table with sites in rows and species in columns. This table can contain either presence-absence or abundance data.
x	An object of class 'indicators'.
object	An object of class 'indicators'.
cluster	A vector containing the classification of sites into site groups.
group	The label corresponding to the target site group.
min.order, max.order	Minimum and maximum number of species conforming species combinations.
max.indicators	Maximum number of valid indicators to be kept. If NULL, then all valid indicators are kept.
func	The indicator value variant to be used, either "IndVal" (non-equalized) or "Ind-Val.g" (group-equalized).
At	Threshold for positive predictive value used to select valid indicators. Combinations with lower values are not kept.
Bt	Threshold for sensitivity used to select valid indicators. Combinations with lower values are not kept.
sqrtIVt	Threshold for (square root of) indicator value. Combinations with lower values are not kept.
alpha	Threshold for statistical significance of indicator value. Combinations with higher p-values are not kept.

<code>control</code>	a list of control values describing properties of the permutation test design, as returned by a call to how .
<code>permutations</code>	a custom matrix of permutations, to be used if <code>control = NULL</code> , with permutations in rows and site indices in columns.
<code>print.perm</code>	If TRUE, prints permutation numbers after each set of 100 permutations.
<code>nboot.ci</code>	Number of bootstrap samples for confidence intervals. If <code>nboot.ci = NULL</code> then confidence intervals are not estimated.
<code>alpha.ci</code>	Error in confidence intervals.
<code>XC</code>	If TRUE, outputs the abundance/occurrence matrix of species combinations.
<code>enableFixed</code>	If TRUE, uses species that occur in all sites as fixed elements and creates combinations with the remaining ones.
<code>verbose</code>	If TRUE, prints the results of each step.
<code>selection</code>	A logical vector used to restrict, a priori, the species combinations to be printed.
<code>confint</code>	Flag to indicate that confidence interval bounds are desired when printing.
<code>type</code>	Statistic to plot. Accepted values are "IV" (indicator value), "sqrtIV" (square root of indicator value), "A", "LA", "UA", (positive predictive value and confidence limits), "B", "LB", "UB" (sensitivity and confidence limits).
<code>maxline</code>	Flag to indicate whether a line has to be drawn joining the maximum values for each order of combinations.
<code>...</code>	Additional arguments for functions <code>print</code> , <code>summary</code> or <code>plot</code> .

Details

Function `indicators` creates explores the indicator value of the simultaneous occurrence of sets of species (i.e. species combinations). The method is described in De Cáceres et al. (2012) and is a generalization of the Indicator Value method of Dufrêne & Legendre (1997). The minimum and maximum number of species conforming the species combination can be controlled using `min.order` or `max.order`. For each combination of species it determines its positive predictive value (A), sensitivity (B) and the square root of indicator value (`sqrtIV`). Statistical significance of indicators for the target site group is determined by internal calls to function [signassoc](#). Additionally, if `nboot.ci` is not null then bootstrap confidence intervals are determined with the specified alpha level, as explained in De Cáceres & Legendre (2009). The combinations to be kept can be restricted to those whose positive predictive value, sensitivity and/or indicator value are equal or greater than input thresholds. Function `print` allows printing the results in a nice table, whereas `summary` provides information about candidate species, combinations and coverage of the set of indicators. Function `plot` draws the statistics against the order (i.e. the number of species) of the combination.

Value

An object of class `indicators` with:

<code>candidates</code>	The vector of initial candidate species.
<code>finalsplist</code>	The vector of species finally selected for combinations.
<code>C</code>	A matrix describing all the combinations studied.

XC	A matrix containing the abundance/occurrence of each species combination.
A	Positive predictive power of species combinations. If nboot is not missing then this includes the lower and upper bounds of the confidence interval.
B	Sensitivity of species combinations. If nboot is not missing then this includes the lower and upper bounds of the confidence interval.
sqrtIV	Square root of indicator value of species combinations. If nboot is not missing then this includes the lower and upper bounds of the confidence interval.
sign	P-value of the permutation test of statistical significance.
group.vec	A logical vector indicating the membership to the target group.

Author(s)

Miquel De Cáceres Ainsa, CTFC

References

- De Cáceres, M., Legendre, P., Wisser, S.K. and Brotons, L. 2012. Using species combinations in indicator analyses. *Methods in Ecology and Evolution* 3(6): 973-982.
- De Cáceres, M. and Legendre, P. 2009. Associations between species and groups of sites: indices and statistical inference. *Ecology* 90(12): 3566-3574.
- Dufrêne, M. and P. Legendre. 1997. Species assemblages and indicator species: The need for a flexible asymmetrical approach. *Ecological Monographs* 67:345-366.

See Also

[predict.indicators](#), [pruneindicators](#), [coverage](#), [multipatt](#), [strassoc](#), [signassoc](#)

Examples

```
library(stats)

data(wetland) ## Loads species data

## Creates three clusters using kmeans
wetkm = kmeans(wetland, centers=3)

## Number of sites in each group
table(wetkm$cluster)

## Run indicator analysis with species combinations for the first group
sc= indicators(X=wetland, cluster=wetkm$cluster, group=1, verbose=TRUE,
              At=0.5, Bt=0.2)

#Prints the results
print(sc)

## Plots positive predictive power and sensitivity against the order of
## combinations
plot(sc, type="A")
```

```

plot(sc, type="B")

## Run indicator analysis with species combinations for the first group,
## but forcing 'Orysp' to be in all combinations
sc2= indicators(X=wetland, cluster=wetkm$cluster, group=1, verbose=TRUE,
               At=0.5, Bt=0.2, enableFixed=TRUE)

```

multipatt

*Multi-level pattern analysis***Description**

This function studies the association between species patterns and combinations of groups of sites.

Usage

```

multipatt(x, cluster, func = "IndVal.g", duleg=FALSE, restcomb=NULL,
          min.order = 1, max.order = NULL, control=how(), permutations = NULL,
          print.perm=FALSE)

```

Arguments

<code>x</code>	Community data table
<code>cluster</code>	A vector representing a partition of sites
<code>func</code>	Species-site group association function. Four values are accepted "IndVal", "IndVal.g", "r" and "r.g" (lowercase values are also accepted).
<code>duleg</code>	If TRUE, site group combinations are not considered, only the original site groups, like in Dufrêne & Legendre (1997). Internally, <code>duleg = TRUE</code> equals <code>max.order = 1</code> .
<code>restcomb</code>	A vector of integer values used to restrict the combinations of site groups to those with ecological sense according to the analyst. The default NULL indicates that all combinations are used. If <code>duleg=TRUE</code> this argument is ignored.
<code>min.order</code>	An integer indicating the minimum order of site group combinations (by default <code>max.order=1</code> for singletons). Cannot be larger than <code>max.order</code> .
<code>max.order</code>	An integer indicating the maximum order of site group combinations to be considered: <code>max.order=1</code> for singletons, <code>max.order=2</code> for pairs, <code>max.order=3</code> for triplets... As <code>restcomb</code> , this parameter provide a way to restrict the site group combinations that make ecological sense. By default all possible site group combinations are considered. If <code>max.order=1</code> then the function will behave as if <code>duleg=TRUE</code> .
<code>control</code>	a list of control values describing properties of the permutation design, as returned by a call to how .
<code>permutations</code>	a custom matrix of permutations, to be used if <code>control = NULL</code> , with permutations in rows and site indices in columns.
<code>print.perm</code>	If TRUE, prints permutation numbers after each set of 100 permutations.

Details

This function creates combinations of the input clusters and compares each combination with the species in the input matrix *x*. For each species it chooses the combination with a highest association value. Best matching patterns are tested for statistical significance of the associations. Four association indices are possible (some less than for `strassoc`): "IndVal", "IndVal.g", "r" and "r.g". Indicator value indices will return the pattern that better matches the species observed pattern, whereas correlation indices will return the pattern that creates a highest inside/outside difference. Details are given in De Cáceres et al. (2010). The user can restrict the combinations in three ways: (1) by using `duleg=TRUE`, which leads to consider single site-groups only; (2) by setting the minimum and maximum order of combinations using `min.order` and `max.order`; or (3) by using `restcomb` to restrict combinations at will. In order to carry out the third way, values in `restcomb` must be the indices of combinations that appear in the column index of object `sign` (see below).

Complex permutation designs are allowed through the function `how` from package "permute". If those are not enough, the user can set `control = NULL` and specify a custom matrix of permutations to test with parameter permutations.

Value

An object of class `multipatt` with:

<code>func</code>	The name of the function used.
<code>comb</code>	A matrix describing all the combinations studied.
<code>str</code>	A matrix the association strength for all combinations studied.
<code>A</code>	If <code>func = "IndVal"</code> (or <code>func = "IndVal.g"</code>) a matrix whose values are the "A" (or "A.g") component of indicator values. Otherwise this element is left as <code>NULL</code> .
<code>B</code>	If <code>func = "IndVal"</code> (or <code>func = "IndVal.g"</code>) a matrix whose values are the "B" component of indicator values. Otherwise this element is left as <code>NULL</code> .
<code>sign</code>	Data table with results of the best matching pattern, the association value and the degree of statistical significance of the association (i.e. p-values from permutation test). Note that p-values are not corrected for multiple testing.

Note

This function gives the same results as function `indval` in package "labdsv" when used setting `func="IndVal.g"` and `duleg=TRUE`, excepting the fact that the square root IndVal values is returned instead of the original IndVal.

Author(s)

Miquel De Cáceres Ainsa, CTFC

Florian Jansen, Institute of Botany and Landscape Ecology, Ernst-Moritz-Arndt-University

References

De Cáceres, M. and Legendre, P. 2009. Associations between species and groups of sites: indices and statistical inference. *Ecology* 90(12): 3566-3574.

De Cáceres, M., Legendre, P., Moretti, M. 2010. Improving indicator species analysis by combining groups of sites. *Oikos* 119(10): 1674-1684.

Dufrêne, M. and P. Legendre. 1997. Species assemblages and indicator species: The need for a flexible asymmetrical approach. *Ecological Monographs* 67:345-366.

See Also

[summary.multipatt](#), [strassoc](#), [signassoc](#), [how](#)

Examples

```
library(stats)

data(wetland) ## Loads species data

wetkm = kmeans(wetland, centers=3) ## Creates three clusters using kmeans

## Runs the combination analysis using IndVal.g as statistic
wetpt = multipatt(wetland, wetkm$cluster, control = how(nperm=999))

## Lists those species with significant association to one combination
summary(wetpt)

## Lists those species with significant association to one combination,
## including indval components.
summary(wetpt, indvalcomp=TRUE)
```

nicheoverlap

Metrics to compare pairs of resource niches

Description

Functions `nicheoverlap` and `nichedisp1` compute the overlap and centroid distance between pairs of resource distributions. In both cases resource relationships are given in the distance matrix `D` and the resource use data are given in data frame `P1` (and in some modes also `P2`).

Usage

```
nicheoverlap(P1, P2 = NULL, D = NULL, q1 = NULL, q2 = NULL, mode = "multiple",
  Np1 = NULL, Np2 = NULL, Nq1 = NULL, Nq2 = NULL, nboot = 1000, alpha=0.05)
nichedisp1(P1, P2 = NULL, D = NULL, q1 = NULL, q2 = NULL, mode = "multiple",
  Np1 = NULL, Np2 = NULL, Nq1 = NULL, Nq2 = NULL, nboot = 1000, alpha=0.05)
```

Arguments

`P1` Data frame containing the amount of usage that a set of species (in rows) make of a first set of resources (in columns).

`P2` Data frame containing the amount of usage that a set of species (in rows) make of a second set of resources (in columns). Not used if `mode = "pairwise"`.

D	Object of type <code>dist</code> containing distance values between resources. If no distance matrix is provided (i.e. if <code>D=NULL</code>), the distances between resources is assumed to be maximum.
q1	Vector with the availability of each resource corresponding to P1.
q2	Vector with the availability of each resource corresponding to P2.
mode	Either <code>mode = "single"</code> (rows of matrices P1 and P2 are individual observations to be pooled, for example to compare the niche of two species each with its individual observations), <code>mode = "multiple"</code> (each row in P1 is compared to the corresponding row of P2, for example, to compare seasonal niche shifts in each species) or <code>mode = "pairwise"</code> (all rows in P1 are compared pairwise).
Np1	Vector with the number of observations per species from which the values in P1 come (in <code>mode = "multiple"</code> or <code>mode = "pairwise"</code>).
Np2	Vector with the number of observations per species from which the values in P2 come (only in <code>mode = "multiple"</code>).
Nq1	The number of observations from which the values in q1 come.
Nq2	The number of observations from which the values in q2 come.
nboot	Number of bootstrap samples used to compute bias-corrected percentile confidence intervals.
alpha	Used to set the confidence level (i.e. <code>alpha = 0.05</code> means 95 percent confidence interval).

Details

The method is described in De Caceres et al. (2011). If the distance matrix is not specified (i.e. if `D=NULL`) the function assumes that all resources are at a maximum distance ($d=1$). If the resource availability vector `q1` (and `q2` if supplied) is specified, then the values in P1 (and P2 if supplied) are taken as assessments of resource use and the species preference is calculated taking into account resource availability. Otherwise, resource use is equated to resource preference (i.e. all resources are considered equally available). The functions can compute bootstrap confidence intervals following the bias-corrected percentile method (Manly 2007). If `mode = "multiple"` and `Np1` and `Np2` are not null, bootstrap samples for a given niche are generated assuming a multinomial distribution with the proportions calculated from the corresponding row values in P1 (resp. P2), and the number of observations comes from the corresponding element in `Np1` (resp. `Np2`). Similarly, if `mode = "pairwise"` and `Np1` is not null, bootstrap samples for each niche are generated assuming a multinomial distribution with the proportions calculated from the corresponding row values in P1, and the number of observations comes from the corresponding element in `Np1`. Finally, if `mode = "single"` then the bootstrapped units are the rows of matrices P1 and P2. In both cases, if `Nq1` (and `Nq2`) is indicated, the availability of resources is also bootstrapped. The bias-corrected percentile method is described for overlap niche measures in Mueller and Altenberg (1985).

Value

Function `nicheoverlap` (resp. `nichedisp1`) returns the overlap (resp. the distance between centroids) between the each pair of rows in P1 and P2. If `mode = "multiple"` or `mode = "single"` the values are returned as a data frame. If `mode = "pairwise"` a matrix of values is returned instead. If bootstrap confidence intervals are asked then the functions also compute the lower and upper

bounds of a confidence interval obtained following the bias-corrected percentile method. Upper and lower bounds are returned as additional columns of the data frame in mode = "multiple" or mode = "single" or as additional matrices of a list in mode = "pairwise".

Author(s)

Miquel De Caceres Ainsa, CTFC

References

Mueller, L.D. and L. Altenberg. 1985. Statistical Inference on Measures of Niche Overlap. *Ecology* 66:1204-1210.

Manly, B.F.J. 2007. Randomization, bootstrap and Monte Carlo methods in biology. Chapman and Hall texts in statistical science series. 2nd edition.

De Caceres, M., Sol, D., Lapiedra, O. and P. Legendre. (2011) A framework for estimating niche metrics using the resemblance between qualitative resources. *Oikos* 120: 1341-1350.

See Also

See [nichevar](#) for descriptors of single niches.

Examples

```
# Loads example data
data(birds)

# The overlap and displacement metrics using distances among
# resources and assuming equal availability of resources
nicheoverlap(birdsbreed, birdswinter, D = resourceD, mode="multiple")
nichedispl(birdsbreed, birdswinter, D = resourceD, mode="multiple")

# The overlap and displacement metrics using distances among resources
# and computes 95 percent confidence intervals
nicheoverlap(birdsbreed, birdswinter, D = resourceD, mode="multiple",
Np1 = rowSums(birdsbreed), Np2 = rowSums(birdswinter), Nq1 = 100, Nq2 = 100)
nichedispl(birdsbreed, birdswinter, D = resourceD, mode="multiple",
Np1 = rowSums(birdsbreed), Np2 = rowSums(birdswinter), Nq1 = 100, Nq2 = 100)

# Same computations with different resource availability
q = c(0.18, 0.24, 0.22, 0.21, 0.15)
nicheoverlap(birdsbreed, birdswinter, D = resourceD,
q1 = q, q2 = q, mode="multiple")
nichedispl(birdsbreed, birdswinter, D = resourceD,
q1 = q, q2 = q, mode="multiple")
nicheoverlap(birdsbreed, birdswinter, D = resourceD,
q1 = q, q2 = q, mode="multiple",
Np1 = rowSums(birdsbreed), Np2 = rowSums(birdswinter),
Nq1 = 100, Nq2 = 100)
nichedispl(birdsbreed, birdswinter, D = resourceD,
q1 = q, q2 = q, mode="multiple",
Np1 = rowSums(birdsbreed), Np2 = rowSums(birdswinter),
```

```

Nq1 = 100, Nq2 = 100)

# The overlap metrics using distances among rows of 'birdsbreed'
nicheoverlap(birdsbreed, D = resourceD, mode="pairwise")

```

nichevar

Resource niche metrics for a single niche

Description

Function `nichepref` computes the species resource preference from a the species resource use (and resource availability when given). Function `nichecentroid` computes the centroid on the resource space for a set of species. Function `nichevar` computes the multivariate resource variance for a set of species (i.e. niche breadth). In all functions resources are given in distance matrix `D`, the species resource use is given in `P` and the availability of resources, if present, are given by vector `q`.

Usage

```

nichevar(P, D = NULL, q = NULL, mode="multiple", Np = NULL,
Nq = NULL, nboot = 1000, alpha=0.05)
nichecentroid(P, D = NULL, q = NULL, mode="multiple", Np = NULL,
Nq = NULL, nboot = 1000, alpha=0.05)
nichepref(P, D = NULL, q = NULL, mode="multiple", Np = NULL,
Nq = NULL, nboot = 1000, alpha=0.05)

```

Arguments

<code>P</code>	Data frame containing the relative or absolute usage that a set of species (in rows) make of a set of resources (in columns).
<code>D</code>	Object of type <code>dist</code> containing distance values between resources. If no distance matrix is provided (i.e. if <code>D=NULL</code>), the distances between resources is assumed to be maximum.
<code>q</code>	Vector with the availability of each resource.
<code>mode</code>	Either <code>mode = "single"</code> (rows of matrix <code>P</code> are individual observations to be pooled for a single niche) or <code>mode = "multiple"</code> (rows in <code>P</code> represent different niches).
<code>Np</code>	Vector with the number of observations per species from which the values in <code>P</code> come (in <code>mode = "multiple"</code>).
<code>Nq</code>	The number of observations per species from which the values in <code>q</code> come (in <code>mode = "multiple"</code>).
<code>nboot</code>	Number of bootstrap samples used to compute bias-corrected percentile confidence intervals.
<code>alpha</code>	Used to set the confidence level (i.e. <code>alpha = 0.05</code> means 95 percent confidence interval).

Details

The method is described in De Caceres et al. (2010). If the distance matrix is not specified (i.e. if $D=NULL$) the function assumes that all resources are at a maximum distance ($d=1$). If the resource availability vector q is given then the values in P are taken as assessments of resource use and the species preference is calculated taking into account resource availability. Otherwise resource use is equated to resource preference. Moreover, most functions can compute bootstrap confidence intervals following the bias-corrected percentile method (Manly 2007). If $mode = "multiple"$ and $Np \neq NULL$, bootstrap samples for a given species are generated assuming a multinomial distribution with the proportions calculated from the corresponding row values in P , and the number of observations comes from the corresponding element in Np . If $mode = "single"$ then the bootstrapped units are the rows of matrix P . In both cases, if Nq is indicated the availability of resources is also bootstrapped. The bias-corrected percentile method was described for overlap niche measures in Mueller and Altenberg (1985) and is extended here for all niche metrics.

Value

Function `nichepref` returns a matrix of species relative preference. Function `nichevar` returns a vector with the variance of the resources used for each species in P . Function `nichecentroid` returns a matrix niche centroid in the resource space for each species in df . If bootstrap confidence intervals are asked then the three functions also compute two extra data containing respectively the lower and upper bounds of the confidence intervals obtained following the bias-corrected percentile method. Function `nichearea` returns the area of the convex hull occupied by the resources used for each species in P .

Author(s)

Miquel De Caceres Ainsa, CTFC

References

Mueller, L.D. and L. Altenberg. 1985. Statistical Inference on Measures of Niche Overlap. *Ecology* 66:1204-1210.

Manly, B.F.J. 2007. Randomization, bootstrap and Monte Carlo methods in biology. Chapman and Hall texts in statistical science series. 2nd edition.

De Caceres, M., Sol, D., Lapidra, O. and P. Legendre. (2011) A framework for estimating niche metrics using the resemblance between qualitative resources. *Oikos* 120: 1341-1350.

See Also

See [nicheoverlap](#) for descriptors comparing two niches.

Examples

```
# Loads example data
data(birds)

# The niche metrics using distances among resources and assuming equal availability of resources
nichepref(birdsbreed, D = resourceD)
nichevar(birdsbreed, D = resourceD)
```

```

nichecentroid(birdsbreed, D = resourceD)

# The niche metrics using distances among resources and computes
# 95 percent confidence intervals
nichepref(birdsbreed, D = resourceD, mode="multiple",
Np = rowSums(birdsbreed), Nq = 100)
nichevar(birdsbreed, D = resourceD, mode="multiple",
Np = rowSums(birdsbreed), Nq = 100)
nichecentroid(birdsbreed, D = resourceD, mode="multiple",
Np = rowSums(birdsbreed), Nq = 100)

# Same computations with different resource availability
nichepref(birdsbreed, D = resourceD,
q = c(0.18, 0.24, 0.22, 0.21, 0.15), mode="multiple")
nichevar(birdsbreed, D = resourceD,
q = c(0.18, 0.24, 0.22, 0.21, 0.15), mode="multiple")
nichecentroid(birdsbreed, D = resourceD,
q = c(0.18, 0.24, 0.22, 0.21, 0.15), mode="multiple")

# The niche metrics using distances among resources and
# computes 95 percent confidence intervals
nichepref(birdsbreed, D = resourceD,
q = c(0.18, 0.24, 0.22, 0.21, 0.15), mode="multiple", Np = rowSums(birdsbreed), Nq = 100)
nichevar(birdsbreed, D = resourceD,
q = c(0.18, 0.24, 0.22, 0.21, 0.15), mode="multiple", Np = rowSums(birdsbreed), Nq = 100)
nichecentroid(birdsbreed, D = resourceD,
q = c(0.18, 0.24, 0.22, 0.21, 0.15), mode="multiple", Np = rowSums(birdsbreed), Nq = 100)

# The following example defines a function to calculate the area of the niche
# It requires package 'rgeos'
library(rgeos)
nichearea <- function(P, D = NULL, axes=c(1,2)) {
  if (is.null(D))
    D <- as.dist((matrix(1, ncol(P), ncol(P)) - diag(rep(1, ncol(P)))))
  cmd = cmdscale(D,eig=TRUE,k= ncol(P)-1)
  X = cmd$points
  V <- data.frame(Area=rep(0, nrow(P)))
  for (i in 1:nrow(P)) {
    pi = as.numeric(P[i,])
    if (is.na(sum(pi))) V[i, ] <- NA
    else if (sum(pi) < 1e-16) V[i, ] <- 0
    else if (sum(pi>0)==1) V[i,]<-0
    else {
      a =X[pi>0,axes]
      V[i,]=area.poly(as(a[chull(a),],"gpc.poly"))
    }
  }
  return(V)
}
nichearea(birdsbreed, D = resourceD)

```

pigeons

Pigeon seed consumption data

Description

Data set used for the dietary analysis of two feral pigeon (*Columba livia*) populations. There are six seed types, and their distance relationships are given in matrix `dfood`.

Usage

```
data(pigeons)
```

Format

`diet.barcelona` - a data frame containing the seed consumption (in grams) of pigeons taken from the population in Barcelona.

`diet.moia` - a data frame containing the seed consumption (in grams) of pigeons taken from the population in Moia.

`dfood` - a distance matrix (object of class `dist`) containing the relationships between the six seed types. Distances were calculated from data describing the nutritional content and dimensions of each seed type.

References

De Caceres, M., Sol, D., Lapiedra, O. and P. Legendre. (2011) A framework for estimating niche metrics using the resemblance between qualitative resources. *Oikos* 120: 1341-1350.

Examples

```
data(pigeons)
```

plotniche

Draws a single resource niche

Description

Function `plotniche` draws a resource niche on the resource space computed by means of principal coordinates analysis. Resource relationships are given in distance matrix `D`, the species resource use is given in `P` and the availability of resources is given by vector `q`.

Usage

```
plotniche(P, D = NULL, q = NULL, mode = "multiple", Np = NULL,  
Nq = NULL, nboot = 1000, alpha=0.05, species=NULL, axes=c(1,2),  
chull=TRUE, bubbles=TRUE, writeName=FALSE, add=FALSE, col="black", lty=1,...)
```

Arguments

P	Data frame containing the relative or absolute usage that a set of species (in rows) make of a set of resources (in columns).
D	Object of type <code>dist</code> containing distance values between resources. If no distance matrix is provided (i.e. if <code>D=NULL</code>), the distances between resources is assumed to be maximum.
q	Vector with the availability of each resource.
mode	Either <code>mode = "single"</code> (rows of matrix P are individual observations to be pooled for a single niche) or <code>mode = "multiple"</code> (rows in P represent different niches).
Np	Vector with the number of observations per species from which the values in P come (in <code>mode = "multiple"</code>).
Nq	The number of observations per species from which the values in q come.
nboot	Number of bootstrap samples used to compute bias-corrected percentile confidence intervals.
alpha	Used to set the confidence level (i.e. <code>alpha = 0.05</code> means 95 percent confidence interval).
species	Specifies which species niche is to be plot. This parameter is mandatory and can be either an numeric index or a string for a species name.
axes	PCoA axes used for plotting.
chull	whether or not convex hulls should be drawn (only in <code>type="single"</code>).
bubbles	draws bubbles proportional to resource preference data.
writeName	flag indicating whether or not the name of the species should be drawn beside the centroid.
col	color of the centroid and confidence interval arrows.
lty	line type of the confidence interval arrows.
add	if TRUE, the current plot is used. This is helpful to draw more than one species on the same plot (see examples).
...	additional graphical parameters.

Details

The method is described in De Caceres et al. (in prep). If the distance matrix is not specified (i.e. if `D=NULL`) the function assumes that all resources are at a maximum distance ($d=1$). If the resource availability vector `q` is given then the values in P are taken as assessments of resource use and the species preference is calculated taking into account resource availability. Otherwise resource use is equated to resource preference. The function can also plot bootstrap confidence intervals following the bias-corrected percentile method (Manly 2007). If `mode = "multiple"` and `Np != NULL`, bootstrap samples for a given species are generated assuming a multinomial distribution with the proportions calculated from the corresponding row values in P, and the number of observations comes from the corresponding element in `Np`. If `mode = "single"` then the bootstrapped units are the rows of matrix P. In both cases, if `Nq` is indicated, the availability of resources is also bootstrapped. The bias-corrected percentile method was described for overlap niche measures in Mueller and Altenberg (1985) and is extended here for all niche metrics except `nichearea`.

Author(s)

Miquel De Caceres Ainsa, CTFC

References

Mueller, L.D. and L. Altenberg. 1985. Statistical Inference on Measures of Niche Overlap. Ecology 66:1204-1210.

Manly, B.F.J. 2007. Randomization, bootstrap and Monte Carlo methods in biology. Chapman and Hall texts in statistical science series. 2nd edition.

De Caceres, M., Sol, D., Lapiedra, O. and P. Legendre. (2011) A framework for estimating niche metrics using the resemblance between qualitative resources. Oikos 120: 1341-1350.

See Also

See [nichevar](#) and [nicheoverlap](#) to obtain a resource niche metrics.

Examples

```
# Loads example data
data(birds)

plotniche(birdsbreed, D = resourceD, mode="multiple", species=10)
plotniche(birdsbreed, D = resourceD, mode="multiple",
Np = rowSums(birdsbreed), Nq = 100, species=10)
plotniche(birdsbreed, D = resourceD,
q = c(0.18, 0.24, 0.22, 0.21, 0.15), mode="multiple", species=10)
plotniche(birdsbreed, D = resourceD,
q = c(0.18, 0.24, 0.22, 0.21, 0.15), mode="multiple",
Np = rowSums(birdsbreed), Nq = 100, species=10)

#draw two species
plotniche(birdsbreed, D = resourceD, mode="multiple",
Np = rowSums(birdsbreed), Nq = 100, species=10, writeName=TRUE,
bubbles=FALSE, chull=FALSE)
plotniche(birdsbreed, D = resourceD, mode="multiple",
Np = rowSums(birdsbreed), Nq = 100, species=1, writeName=TRUE,
bubbles=FALSE, chull=FALSE, add=TRUE, col="red", lty=2)
```

predict.indicators *Predicts site group from indicators*

Description

Function `predict.indicators` takes an object of class `indicators` and determines the probability of the indicated site group given a community data set. If no new data set is provided, the function can calculate the probabilities corresponding to the original sites used to build the `indicators` object.

Usage

```
## S3 method for class 'indicators'  
predict(object, newdata = NULL, cv = FALSE,...)
```

Arguments

object	An object of class 'indicators'.
newdata	A community data table (with sites in rows and species in columns) for which predictions are needed. This table can contain either presence-absence or abundance data, but only presence-absence information is used for the prediction. If NULL, then the original data set used to derive the <code>indicators</code> object is used as data.
cv	A boolean flag to indicate that probabilities should be calculated using leave-one-out cross validation (i.e recalculating positive predictive value of indicators after excluding the target site).
...	In function <code>predict</code> , additional arguments not used (included for compatibility with <code>predict</code>).

Details

Function `indicators` explores the indicator value of the simultaneous occurrence of sets of species (i.e. species combinations). The method is described in De Cáceres et al. (2012) and is a generalization of the Indicator Value method of Dufrene & Legendre (1997). The current function `predict.indicators` is used to predict the indicated site group from the composition of a new set of observations. For communities where one or more of the indicator species combinations are found, the function returns the probability associated to the indicator that has the highest positive predictive value (if confidence intervals are available, the maximum value is calculated across the lower bounds of the confidence interval). For communities where none of the indicator species combinations is found, the function returns zeroes. If `newdata = NULL`, the function can be used to evaluate the predictive power of a set of indicators in a cross-validated fashion. For each site in the data set, recalculates the predictive value of indicators after excluding the information of the site, and then evaluates the probability of the site group.

Value

If confidence intervals are available in `x`, function `predict.indicators` returns a matrix where communities are in rows and there are three columns, corresponding to the probability of the indicated site group along with the confidence interval. If confidence intervals are not available in `x`, or if `cv = TRUE`, then `predict.indicators` returns a single vector with the probability of the indicated site group for each community.

Author(s)

Miquel De Cáceres Ainsa, CTFC

References

De Cáceres, M., Legendre, P., Wisser, S.K. and Brotons, L. 2012. Using species combinations in indicator analyses. *Methods in Ecology and Evolution* 3(6): 973-982.

Dufrêne, M. and P. Legendre. 1997. Species assemblages and indicator species: The need for a flexible asymmetrical approach. *Ecological Monographs* 67:345-366.

See Also

[indicators](#), [pruneindicators coverage](#), [multipatt](#), [strassoc](#), [signassoc](#)

Examples

```
library(stats)

data(wetland) ## Loads species data

## Creates three clusters using kmeans
wetkm = kmeans(wetland, centers=3)

## Run indicator analysis with species combinations for the first group
sc= indicators(X=wetland, cluster=wetkm$cluster, group=1, verbose=TRUE, At=0.5, Bt=0.2)

## Use the indicators to make predictions of the probability of group #1
## Normally an independent data set should be used, because 'wetland' was used to derive
## indicators. The same would be obtained calling 'predict(sc)' without further arguments.
p<-predict(sc, wetland)

## Calculate cross-validated probabilities (recalculates 'A' statistics once for each site
## after excluding it, and then calls predict.indicators for that site)
pcv = predict(sc, cv = TRUE)

## Show original membership to group 1 along with (resubstitution) predicted probabilities
## and cross-validated probabilities. Cross-validated probabilities can be lower for sites
## originally belonging to the target site group and higher for other sites.
data.frame(Group1 = as.numeric(wetkm$cluster==1), Prob = p, Prob_CV = pcv)
```

pruneindicators	<i>Determines the best subset of indicators</i>
-----------------	---

Description

This function allows reducing drastically the number of species combinations to be retained for a given target site group.

Usage

```
pruneindicators(x, At=0, Bt=0, sqrtIVt=0, alpha = 1.0,
                max.indicators=4, verbose=FALSE)
```

Arguments

x	An object of class ' indicators '.
At	Threshold for positive predictive value. Combinations with lower values are not kept.
Bt	Threshold for sensitivity. Combinations with lower values are not kept.
sqrtIVt	Threshold for (square root of) indicator value. Combinations with lower values are not kept.
alpha	Threshold for statistical significance of indicator value. Combinations with higher p-values are not kept.
max.indicators	Maximum number of species combinations to be kept. If NULL, the function returns all the non-nested valid indicators without further selection.
verbose	If TRUE, prints the results of each step.

Details

First, the function selects those indicators (species or species combinations) with valid positive predictive value, sensitivity and indicator value, according to the input thresholds. If the object 'speciescomb' contains confidence intervals, then the lower bounds are used to select the valid indicators. Second, the function discards those valid indicators whose occurrence pattern is nested within other valid indicators. Third, the function evaluates the [coverage](#) of the remaining set of indicators and explores subsets of increasing number of indicators, until the same coverage is attained and the set of indicators is returned. If the maximum allowed members is attained (max.indicators) then the set of indicators with maximum coverage is returned.

Value

An object of class '[indicators](#)' with only the species combinations selected.

Author(s)

Miquel De Cáceres Ainsa, CTFC

References

De Cáceres, M., Legendre, P., Wisser, S.K. and Brotons, L. 2012. Using species combinations in indicator analyses. *Methods in Ecology and Evolution* 3(6): 973-982.

De Cáceres, M. and Legendre, P. 2009. Associations between species and groups of sites: indices and statistical inference. *Ecology* 90(12): 3566-3574.

See Also

[indicators](#), [coverage](#)

Examples

```

library(stats)

data(wetland) ## Loads species data

## Creates three clusters using kmeans
wetkm = kmeans(wetland, centers=3)

## Run indicator analysis with species combinations for the first group
sc= indicators(X=wetland, cluster=wetkm$cluster, group=1, verbose=TRUE, At=0.5, Bt=0.2)

## Finds the 'best' subset of indicators
sc2=pruneindicators(sc, At=0.5, Bt=0.2, verbose=TRUE)
print(sc2)

```

signassoc

Statistical significance of species site-group associations

Description

This function computes the permutation p-value of the association between a species vector and a vector of memberships to a site-group. Four different tests of hypotheses arise from considering either presence/absence or quantitative species data, and from using a non-equalized or group-equalized approach.

Usage

```

signassoc(X, U = NULL, cluster = NULL, mode = 1, alternative = "greater",
          control = how(), permutations = NULL, print.perm = FALSE)

```

Arguments

X	species data (rows are sites, columns are species).
U	A matrix of group memberships. Each column corresponds to one site-group. Membership values can be crisp or fuzzy. If this matrix is not provided, vector <code>gmv</code> will be used.
cluster	a vector of numeric group labels for the sites.
mode	0 (site-based) or 1 (group-based).
alternative	Alternative hypothesis: "greater" (positive association), "less" (negative association), "two.sided" (either positive or negative).
control	a list of control values describing properties of the permutation design, as returned by a call to how .
permutations	a custom matrix of permutations, to be used if <code>control = NULL</code> , with permutations in rows and site indices in columns.
print.perm	If TRUE, prints permutation numbers after each set of 100 permutations.

Details

Input data for this function is the species matrix X and either a matrix of group memberships (U) or a vector of numeric group labels (`cluster`). This R function works for both presence/absence and quantitative species data, depending on the values of the input matrix X . If `mode = 0`, the null ecological hypothesis is that the frequency (or abundance) of the species of interest in sites belonging to the site group is not higher than the frequency (or abundance) in sites not belonging to it. If `mode = 1`, the null ecological hypothesis is that the relative frequency (average abundance) of the species of interest is not higher in the target site group than in other groups. See De Cáceres and Legendre for more details.

Complex permutation designs are allowed through the function `how` from package "permute". If those are not enough, the user can set `control = NULL` and specify a custom matrix of permutations to test with parameter permutations.

Value

Returns a matrix of p-values, where species are in rows and groups are in columns. Two additional columns indicate the group with lowest p-value and the p-value for this group after Sidak's correction for multiple testing.

Note

Users should be aware that the significance test in `signassoc` is not exactly the same as the one in `indval` from `labdsv` package. The `signassoc` function is using the preference for the target group (either non-equalized or group-equalized) as test statistic. After every permutation the preference for the target group is recalculated. The function is therefore testing the null hypothesis stating that the preference of the species for a given site group is due to chance only (as in Bakker 2008). The test is repeated for every group, and this is the reason why there are as many p-values as groups. In contrast, the `indval` function from `labdsv` package uses the maximum preference value as test statistic, and the maximum preference value is recalculated after each permutation. The maximum preference may correspond to other groups than the one chosen for the unpermuted data. `indval` function from `labdsv` package is therefore testing the null hypothesis saying that the group with observed maximum preference is not such, because the maximum preference was in that group due to chance only. In order to get the consistent results compared to the `indval` function, users should use the function `multipatt`, along with the option `duleg=TRUE`.

Author(s)

Miquel De Cáceres Ainsa, CTFC

References

- Bakker, J. 2008. Increasing the utility of Indicator Species Analysis. *Journal of Applied Ecology* 45: 1829-1835.
- De Cáceres, M. and Legendre, P. 2009. Associations between species and groups of sites: indices and statistical inference. *Ecology* 90(12): 3566-3574.

See Also

[strassoc](#), [multipatt](#), [how](#)

Examples

```
library(stats)

data(wetland) ## Load species data

wetkm = kmeans(wetland, centers=3)## Create three clusters using

## Look for species whose abundance is significantly higher in one of the three groups
signassoc(wetland, cluster=wetkm$cluster, mode=1, control = how(nperm=999))

## Look for species whose abundance is significantly higher in sites belonging
## to one group as opposed to sites not belonging to it.
signassoc(wetland, cluster=wetkm$cluster, mode=0, control = how(nperm=999))
```

strassoc

Strength of species site-group associations

Description

This function computes the strength of the association between a species vector and a vector of memberships to the site-group. Several statistics are possible, following De Cáceres and Legendre (2009). The choice of the most appropriate statistic depends on the application. If required, bootstrap confidence interval bounds are also computed.

Usage

```
strassoc(X, cluster, func = "r", group = NULL, nboot.ci = NULL, alpha.ci = 0.05, c = 1)
```

Arguments

X	species data (rows are sites, columns are species).
cluster	a vector of numeric group labels for the sites.
func	the association index to be used. Accepted function names: "r", "r.g", "IndVal", "IndVal.g", "A", "A.g", "B", "cos", "cos.g", "r.ind", "r.ind.g", "s.ind", "s.ind.g" (lowercase values are also accepted).
group	Group for which association values are to be computed. If group=NULL, association values will be computed for all groups.
nboot.ci	Number of bootstrap samples. If NULL then confidence intervals are not estimated.
alpha.ci	Error in confidence intervals.
c	Total abundance per site (used in functions "r.ind", "r.ind.g", "s.ind", "s.ind.g" only).

Details

This R function is applicable to both presence-absence and quantitative species data, depending on the values in the input matrix *X*. Indices "r", "r.g", "cos" and "cos.g" correspond to the general correlation generalization ("r" is the point-biserial correlation coefficient), whereas "r.ind", "r.ind.g", "s.ind" and "s.ind.g" correspond to the individual-based generalization. Both approaches give the same results for binary (presence-absence) species data. Indicator value indices "IndVal" and "IndVal.g" are partial generalizations of the indices of the presence-absence indices. The "IndVal" index of Duf rene & Legendre (1997) is called "IndVal.g" here and `strassoc` returns the square root of the original index. Indices "A", "A.g" and "B" are the asymmetric quantities into which indicator values "IndVal" and "IndVal.g" can be decomposed. See De C ceres and Legendre for details on the relationships between these indices and their usage. Bootstrap confidence intervals are computed using the simple percentile method (Manly 1997).

Value

Returns a matrix of association values, where species are in rows and groups are in columns. If `nboot.ci` is not null, then a list is returned, where `'stat'` contains the matrix of association values, and the lower and upper confidence limits are given in supplementary matrices named `'lowerCI'` and `'upperCI'` respectively.

Note

This function gives the same association values as function `indval` in package "labdsv" when used setting `func="IndVal.g"`, excepting the fact that the square root `IndVal` values is returned instead of the original `IndVal`.

Author(s)

Miquel De C ceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnol gic Forestal de Catalunya

References

- De C ceres, M. and Legendre, P. 2009. Associations between species and groups of sites: indices and statistical inference. *Ecology* 90(12): 3566-3574.
- Duf rene, M. and P. Legendre. 1997. Species assemblages and indicator species: The need for a flexible asymmetrical approach. *Ecological Monographs* 67:345-366.
- Manly, B. F. J. 1997. Randomization, bootstrap and Monte Carlo methods in biology. Chapman and Hall Texts in Statistical Science Series.

See Also

[signassoc](#), [multipatt](#)

Examples

```
## Load species data
data(wetland)
```

```
## Create three clusters using
wetkm = kmeans(wetland, centers=3)

## Compute Dufrene and Legendre's IndVal
strassoc(wetland, wetkm$cluster, func="IndVal.g")

## Compute point-biserial correlation, with bootstrap 95 percent confidence intervals
strassoc(wetland, wetkm$cluster, func="r",
         nboot.ci =100)
```

summary.multipatt *Summarizing multi-level pattern analysis*

Description

This is a summary function for multipatt objects.

Usage

```
## S3 method for class 'multipatt'
summary(object, alpha=0.05, minstat=NULL, At = NULL, Bt=NULL, indvalcomp=FALSE, ...)
```

Arguments

object	An object returned by function <code>multipatt</code> .
alpha	Significance level for selecting species in the summary.
minstat	Minimum value of the statistic for selecting species in the summary.
At	Minimum value of positive predictive value (A) for selecting indicators. This argument is effective only if multipatt function was executed using <code>func = "IndVal"</code> or <code>func = "IndVal.g"</code> .
Bt	Minimum value for sensitivity (B) for selecting indicators. This argument is effective only if multipatt function was executed using <code>func = "IndVal"</code> or <code>func = "IndVal.g"</code> .
indvalcomp	Flag used to report components "A" (or "A.g") and "B" of indicator values. This argument is effective only if multipatt function was executed using <code>func = "IndVal"</code> or <code>func = "IndVal.g"</code> .
...	Additional arguments of the summary function.

Details

The summary function for multipatt objects presents the results of the analysis by grouping species following their associated site group combination. Only significantly associated species are listed for each combination. The user can limit the species shown by setting a significance level and the minimum value for the strength of the association.

Author(s)

Miquel De Cáceres Ainsa, CTFC

See Also

[strassoc](#), [signassoc](#), [multipatt](#)

Examples

```
library(stats)

## Loads species data
data(wetland)

## Creates three clusters using kmeans
wetkm = kmeans(wetland, centers=3)

## Runs the combination analysis using IndVal.g as statistic
wetpt = multipatt(wetland, wetkm$cluster)

## Lists those species with significant association to one combination
summary(wetpt)

## Lists those species with significant association to one combination,
## including indval components
summary(wetpt, indvalcomp=TRUE)
```

wetland

Wetland vegetation data set

Description

Vegetation of the Adelaide river alluvial plain (Australia). This data set was published by Bowman & Wilson (1987) and used in Dale (1988) to compare fuzzy classification approaches.

Usage

```
data(wetland)
```

Format

A data frame with 41 sites (rows) and 33 species (columns). Abundance values are represented in abundance classes.

Source

Bowman, D. M. J. S. and B. A. Wilson. 1986. Wetland vegetation pattern on the Adelaide River flood plain, Northern Territory, Australia. *Proceedings of the Royal Society of Queensland* 97:69-77.

References

Dale, M. B. 1988. Some fuzzy approaches to phytosociology. Ideals and instances. *Folia geobotanica et phytotaxonomica* 23:239-274.

Examples

```
data(wetland)
```

Index

*Topic **datasets**

birds, 3
pigeons, 19
wetland, 30

*Topic **package**

indicspecies-package, 2

birds, 3
birdsbreed (birds), 3
birdswinter (birds), 3

combinespecies, 4
coverage, 5, 10, 23, 24

dfood (pigeons), 19
diet.barcelona (pigeons), 19
diet.moia (pigeons), 19
dist, 3, 14, 16, 19, 20

how, 9, 11–13, 25, 26

indicators, 2, 4–7, 8, 21–24
indicspecies (indicspecies-package), 2
indicspecies-package, 2

multipatt, 2, 6, 7, 10, 11, 23, 26, 28–30

nichecentroid (nichevar), 16
nichedispl (nicheoverlap), 13
nicheoverlap, 13, 17, 21
nichepref (nichevar), 16
nichevar, 15, 16, 21

pigeons, 19
plot.indicators (indicators), 8
plotcoverage (coverage), 5
plotniche, 19
predict, 22
predict.indicators, 10, 21
print.indicators (indicators), 8
pruneindicators, 7, 10, 23, 23

resourceD (birds), 3

signassoc, 2, 4, 5, 9, 10, 13, 23, 25, 28, 30
strassoc, 2, 4, 5, 10, 12, 13, 23, 26, 27, 30
summary.indicators (indicators), 8
summary.multipatt, 13, 29

wetland, 30