

Package ‘metacom’

July 2, 2014

Type Package

Title Analysis of the 'elements of metacommunity structure'

Version 1.3

Date 2012-12-28

Author Tad Dallas

Depends vegan, lattice

Maintainer Tad Dallas <tad.dallas@drakeresearchlab.com>

Description Functions to analyze coherence, boundary clumping, and turnover following the pattern-based metacommunity analysis of Leibold and Mikkelsen 2002

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2013-11-13 16:49:10

R topics documented:

metacom-package	2
BoundaryClump	3
Coherence	5
Imagine	6
Metacommunity	7
NullMaker	9
OrderMatrix	11
TestMatrices	12
Turnover	13

Index	15
--------------	-----------

metacom-package

Pattern-based analysis of metacommunity structure

Description

'metacom' provides functions for the analysis of the elements of metacommunity structure (coherence, boundary clumping, & turnover), following the pattern-based metacommunity framework of Leibold & Mikkelsen 2002 and Presley et al. 2010. This package is designed to allow the user to distinguish between several idealized patterns of metacommunity structure (Presley et al. 2010) utilizing any number of null model algorithms for the randomization procedure. However, these metrics can also be used in isolation, and without ordination via reciprocal averaging, and instead, ordering along some biological gradient.

Details

Package: metacom
Type: Package
Version: 1.0
Date: 2012-12-28
License: GPL-2

A metacommunity is a set of sites (e.g. plants in plant-pollinator networks) associated through interactions (e.g. insect species (columns) interact with plant species (rows) in plant-pollinator networks). The pattern-based metacommunity concept, proposed by Leibold & Mikkelsen 2002 and expounded on by Presley et al. 2010, allows for the evaluation of metacommunity structure by using randomization techniques to discern between 10 patterns of metacommunity structure. This is performed by ordinating site-by-species interaction matrices and calculating three metrics; coherence, boundary clumping & turnover.

The metacom package calculates these three metrics; coherence is calculated using the function `Coherence()`, boundary clumping with `BoundaryClump()`, and turnover (from either species or range perspective) using the `Turnover()` function. These functions are consolidated in the `Metacommunity()` function, which can be used to calculate all three metrics. In order to interpret the output of these functions, it will be helpful to read Leibold & Mikkelsen 2002 and Presley et al. 2010, but to also read Ulrich and Gotelli 2013, as this paper outlines the difficulty seemingly inherent with investigating community structure. Also, these functions do not have to be used strictly in the Leibold and Mikkelsen 2002 framework.

I caution the user to be aware that the creation of null matrices can be performed to allow (or not allow) sites to be empty, or species to not exist at any site (i.e. column sums and/or row sums are allowed to be zero). This is controlled by the logical argument 'allow.empty' in the `metacommunity()`, `nullmaker()`, `Coherence()`, and `Turnover()` functions. Restricting nulls to not allow empty rows or columns may be biologically realistic, but it also reduces the number of unique null matrices that can be built, which will impact computation time, making it infeasible or impossible in some situations. These situations occur when you have a very sparse interaction matrix, and is also influenced by null model algorithm ('method') that you choose.

The 'metacom' package is partially adapted from previous Matlab code written by Christopher Higgins (available at <http://www.tarleton.edu/Faculty/higgins/EMS.htm>) and relies on many functions in the 'vegan' package (Oksanen et al. 2012)

Author(s)

Tad Dallas

Tad Dallas <tad.dallas@drakeresearchlab.com>

References

- Leibold, M. & Mikkelsen, G. (2002) Coherence, species turnover, and boundary clumping: elements of metacommunity structure. *Oikos*, 97, 237-250.
- Leibold, M., Holyoak, M., Mouquet, N., Amarasekare, P., Chase, J., Hoopes, M., Holt, R., Shurin, J., Law, R., Tilman, D. et al. (2004) The metacommunity concept: a framework for multi-scale community ecology. *Ecology letters*, 7, 601-613.
- Oksanen, J., F.G. Blanchet, R. Kindt, P. Legendre, P.R. Minchin, R.B. O'Hara, G.L. Simpson, P. Solymos, M.H.H. Stevens and H. Wagner (2012). *vegan: Community Ecology Package*. R package version 2.0-4. <http://CRAN.R-project.org/package=vegan>
- Presley, S., Higgins, C. & Willig, M. (2010) A comprehensive framework for the evaluation of metacommunity structure. *Oikos*, 119, 908-917.
- Ulrich, W. and Gotelli, N. J. (2013) Pattern detection in null model analysis. *Oikos*, 122: 2-18. doi: 10.1111/j.1600-0706.2012.20325.x
- Willig, M., Presley, S., Bloch, C., Castro-Arellano, I., Cisneros, L., Higgins, C. & Klingbeil, B. (2011) Tropical metacommunities along elevational gradients: effects of forest type and other environmental factors. *Oikos*, 120, 1497-1508.

BoundaryClump

Determines boundary clumping

Description

'BoundaryClump' calculates the Morisita's Index (Morisita 1962) for presence-absence interaction matrices, using a chi-squared test to assess significance.

This statistic is not based on randomization methods, so the function only requires a presence-absence interaction matrix and two arguments regarding the ordination of the empirical matrix.

The default is the range perspective, meaning that the analyses of boundary clumping and species turnover compare the distribution of species among sites. If the 'community' perspective is desired, transpose the matrix before analysis using the transpose function ('t()'). However, the author cautions against misinterpretation of the community perspective, as the biological meaning of turnover and boundary clumping differ between perspectives.

Usage

```
BoundaryClump(comm, order=TRUE, scores=1)
```

Arguments

comm	community data in the form of a presence absence matrix
order	logical argument indicating whether to ordinate the interaction matrix or not. See details.
scores	axis scores to ordinate matrix. 1: primary axis scores (default) 2: secondary axis scores

Details

Boundary clumping, quantified by the Morisita's index, is a measure of the degree to which species range boundaries overlap. This measure, and species turnover, cannot be interpreted unless the network is significantly coherent (see 'Coherence').

If 'order' is FALSE, the interaction matrix is not ordinated, allowing the user to order the matrix based on site characteristics or other biologically relevant characteristics.

Value

'BoundaryClump' returns a data frame containing the calculated Morisita's index ('index'), the corresponding p-value ('P'), and the degrees of freedom ('df').

The p-value is based on a chi-squared test comparing the Morisita's index to a value of 1. If the Morisita's index is less than 1, a left-tailed test is performed (less clumping than expected by chance).

If the Morisita's index is greater than 1, a right-tailed test is performed (more clumping than expected by chance)

Author(s)

Tad Dallas

References

Morisita, M. 1962. Id-index, a measure of dispersion of individuals. Res. Popul. Ecol. 4, 1-7.

Examples

```
## define an interaction matrix
data(TestMatrices)
intmat=TestMatrices[[1]]

## analysis of boundary clumping
bound.test=BoundaryClump(intmat)

## prints a single row data.frame containing the Morisita's index, p-value, and degrees of freedom
bound.test
```

Coherence	<i>Determines coherence</i>
-----------	-----------------------------

Description

This function determines the number of embedded absences in an interaction matrix, and compares this value against null simulated matrices. Species ranges should be coherent along the ordination axis, as this axis represents a latent environmental gradient. A negative value of coherence (empirical matrix has more embedded absences than null matrices) indicates a 'checkerboard' pattern (Leibold & Mikkelsen 2002). Nonsignificance has been historically interpreted as being indicative of a 'random' pattern, though this may be seen as accepting the null hypothesis, as nonsignificance cannot be used to infer a process.

Usage

```
Coherence(comm, method = "r1", sims = 1000, scores = 1,
order=TRUE, allow.empty=FALSE)
```

Arguments

<code>comm</code>	community data in the form of a presence absence matrix
<code>method</code>	null model randomization method used by 'nullmaker'. See details below (and the help file of function 'nullmaker') for more information.
<code>sims</code>	number of simulated null matrices to use in analysis
<code>scores</code>	axis scores to ordinate matrix. 1: primary axis scores (default) 2: secondary axis scores
<code>order</code>	logical argument indicating whether to ordinate the interaction matrix or not. See details.
<code>allow.empty</code>	logical argument indicating whether to allow null matrices to have empty rows or columns

Details

'method' is an argument handed to functions in the 'vegan' package. Leibold & Mikkelsen advocated the use of equiprobable rows and columns (provided that rows and columns had at least one entry). This method is called 'r00'. Methods maintaining row (site) frequencies include 'r0', 'r1' & 'r2'. The default method argument is 'r1', which maintains the species richness of a site (row totals) and fills species ranges (columns) based on their marginal probabilities. Arguably the most conservative null algorithm is the fixed row - fixed column total null, which is implemented as 'fixedfixed'. See the help file for 'commsimulator' or Wright et al. 1998 for more information.

If 'order' is FALSE, the interaction matrix is not ordinated, allowing the user to order the matrix based on site characteristics or other biologically relevant characteristics.

This function can either be used as a standalone, or can be used through the 'metacommunity()' function, which determines all 3 elements of metacommunity structure (coherence, boundary clumping, & turnover) (Leibold & Mikkelsen 2002)

Value

Coherence	the number of embedded absences within the interaction matrix
z	the z statistic comparing observed turnover against species turnover of null matrices
pval	p-value from a z-test to assess significance of species turnover relative to null matrices
simulated mean	mean number of replacements in simulated null matrices
simulated variance	variance in the number of replacements of simulated null matrices
method	randomization method to create null matrices

Author(s)

Tad Dallas

References

Leibold, M.A. and G.M. Mikkelsen. 2002. Coherence, species turnover, and boundary clumping: elements of meta-community structure. *Oikos* 97: 237 - 250.

Wright, D.H., Patterson, B.D., Mikkelsen, G.M., Cutler, A. & Atmar, W. (1998). A comparative analysis of nested subset patterns of species composition. *Oecologia* 113, 1-20.

Examples

```
#define an interaction matrix
data(TestMatrices)
intmat=TestMatrices[[2]]

#determine coherence of interaction matrix
coh.intmat=Coherence(intmat, method='r1', sims=1000, scores=1, order=TRUE)

#return results
coh.intmat
```

Imagine

Plotting of site-by-species interaction matrices

Description

'Imagine' produces an image plot, grid of small rectangles representing species occurrences in sites, of a given interaction matrix.

Usage

```
Imagine(comm, col=c(0,1), ordinate=TRUE, fill=TRUE, xlab='Species',
ylab='Site', sitenames=rownames(comm), speciesnames=colnames(comm))
```

Arguments

<code>comm</code>	community data in the form of a presence absence matrix
<code>col</code>	colors used to plot interactions. First value is the background color (no interaction) and the second color indicates an interaction.
<code>ordinate</code>	logical. Should the interaction matrix be ordinated before plotting?
<code>fill</code>	logical. Should species ranges be made coherent before plotting?
<code>xlab</code>	name of the x axis
<code>ylab</code>	name of the y axis
<code>sitenames</code>	names for each row in the interaction matrix. Default is to not plot names.
<code>speciesnames</code>	names for each site in the interaction matrix. Default is to not plot names.

Value

Produces an image plot of the interaction matrix. The code is very simple, and may need to be modified if you have long site or species names, or wish to make it prettier than I have the ability to.

Author(s)

Tad Dallas

Examples

```
#define an interaction matrix
data(TestMatrices)
pres3c=TestMatrices[[6]]

#plot interaction matrix
Imagine(pres3c, col=c('white','blue'), ordinate=TRUE, fill=FALSE)
```

Description

'Metacommunity' is a wrapper for the analysis of the three elements of metacommunity structure (coherence, boundary clumping, & turnover) following the framework of Leibold & Mikkelsen 2002. It is important to note here that the results of boundary clumping and turnover are only relevant if the matrix is significantly positively coherent (i.e. empirical matrix has fewer embedded absences than null matrices).

Usage

```
Metacommunity(comm, scores=1, method = "r1", sims = 1000,
order=TRUE, allow.empty=FALSE)
```

Arguments

comm	community data in the form of a presence absence matrix
scores	Axis scores to ordinate matrix. 1: primary axis scores (default) 2: secondary axis scores. See Details.
method	null model randomization method used by 'nullmaker'. See details.
sims	number of simulated null matrices to use in analysis
order	logical argument indicating whether to ordinate the interaction matrix or not. See details.
allow.empty	logical argument indicating whether to allow null matrices to have empty rows or columns

Details

'method' is an argument handed to functions in the 'vegan' package. Leibold & Mikkelsen advocated the use of equiprobable rows and columns (provided that rows and columns had at least one entry). This method is called 'r00'. Methods maintaining row (site) frequencies include 'r0', 'r1' & 'r2'. The default method argument is 'r1', which maintains the species richness of a site (row totals) and fills species ranges (columns) based on their marginal probabilities. Arguably the most conservative null algorithm is the fixed row - fixed column total null, which is implemented as 'fixedfixed'. See the help file for 'commsimulator' or Wright et al. 1998 for more information.

If 'order' is FALSE, the interaction matrix is not ordinated, allowing the user to order the matrix based on site characteristics or other biologically relevant characteristics.

Value

Comm	ordinated matrix used to calculate coherence, boundary clumping & turnover
Coherence	output of the Coherence() function, giving information on the number of embedded absences and the significance relative to simulated null matrices
Turnover	output of the Turnover() function, testing the number of species replacements relative to simulated null matrices
Boundary	output of the BoundaryClump() function, which calculates the Morisita's index, assessing significance using a chi-squared test

Note

This function may take awhile to finish as a result of the creation of null matrices. If you are running multiple interaction matrices, the code can be parallelized using the 'snow' package.

Author(s)

Tad Dallas

References

- Leibold, M.A. and G.M. Mikkelsen. 2002. Coherence, species turnover, and boundary clumping: elements of meta-community structure. *Oikos* 97: 237 - 250.
- Presley, S. J., C. L. Higgins, and M. R. Willig. 2010. A comprehensive framework for the evaluation of metacommunity structure. *Oikos* 119:908-917
- Wright, D.H., Patterson, B.D., Mikkelsen, G.M., Cutler, A. & Atmar, W. (1998). A comparative analysis of nested subset patterns of species composition. *Oecologia* 113, 1-20.

Examples

```
#define an interaction matrix
data(TestMatrices)
lei2b=TestMatrices[[4]]

#determines the elements of metacommunity structure
ems.intmat=Metacommunity(lei2b, method='r1', sims=1000, scores=1)

#return results in list form
ems.intmat
```

NullMaker

Null matrix creator

Description

Creates null matrices based on the constraints of the null model algorithm ('method'). Also allows for null matrices with a species that occurs at no sites, or a site without any species to be removed from the suite of simulated null matrices. This function borrows heavily from the `commsimulator()` function in the 'vegan' package, but also allows for the fixed-fixed null model.

Usage

```
NullMaker(comm, sims = 1000, method = "r1", allow.empty=FALSE)
```

Arguments

<code>comm</code>	community data in the form of a presence absence matrix
<code>sims</code>	number of simulated null matrices to use in analysis
<code>method</code>	null model randomization method. See details below.
<code>allow.empty</code>	logical argument indicating whether to allow null matrices to have empty rows or columns

Details

'method' is the null model algorithm used to create the null matrices. The choice of a null algorithm is nontrivial. Leibold & Mikkelsen advocated the use of equiprobable rows and columns (provided that rows and columns had at least one entry). This method is called 'r00'. Methods maintaining row (site) frequencies include 'r0', 'r1' & 'r2', whereas species (column) occurrences are preserved with fixed column methods such as 'c0'. The default method argument is 'r1', which maintains the species richness of a site (row totals) and fills species ranges (columns) based on their marginal probabilities. Arguably the most conservative null algorithm is the fixed row - fixed column total null, which is implemented as 'fixedfixed'. There are also sequential swap methods 'tswap', 'swap', and non-sequential 'quasiswap' and 'backtracking' that can be implemented as well. See the help file for 'commsimulator' or Wright et al. 1998 for more information.

Value

The output is a list of length 'sims' consisting of generated null matrices

Author(s)

Tad Dallas

References

J. Oksanen, F.G. Blanchet, R. Kindt, P. Legendre, P.R. Minchin, R.B. O'Hara, G.L. Simpson, P. Solymos, M.H.H. Stevens and H. Wagner (2012). *vegan: Community Ecology Package*. R package version 2.0-4. <http://CRAN.R-project.org/package=vegan>

See Also

`commsimulator()`

Examples

```
#define an interaction matrix
data(TestMatrices)
intmat=TestMatrices[[4]]

#creation of the null matrices
nulls=NULLMaker(intmat, sims=1000,method='r1')

#output is a list of length sims with each list element corresponding to a null matrix
str(nulls)
```

OrderMatrix	<i>Ordinates interaction matrix</i>
-------------	-------------------------------------

Description

'OrderMatrix' ordines an interaction matrix scores derived from reciprocal averaging (Gauch et al. 1977). These scores represent a latent environmental gradient along which species distributions are structured.

Usage

```
OrderMatrix(comm, scores = 1, output.scores=FALSE)
```

Arguments

comm	community data in the form of a presence absence matrix
scores	axis scores to ordinate matrix. 1: primary axis scores (default) 2: secondary axis scores
output.scores	logical. Default is to return the ordinated matrix. If 'output.scores' is TRUE, the function returns the site and species scores.

Value

'OrderMatrix' returns either an ordinated matrix (output.scores = FALSE) or the site and species scores (output.scores = TRUE) obtained from reciprocal averaging. This function is already contained within functions calculating coherence, species turnover & boundary clumping, but may be useful for visualizations or for hypothesis testing concerning the important variables associated with the site or species scores.

Note

'OrderMatrix', like many of these functions, relies heavily on the 'vegan' package.

Author(s)

Tad Dallas

References

Gauch, H.G., R.H. Whittaker, and T.R. Wentworth. 1977. A comparative study of reciprocal averaging and other ordination techniques. *Journal of Ecology* 65:157-174.

Leibold, M.A. and G.M. Mikkelsen. 2002. Coherence, species turnover, and boundary clumping: elements of meta-community structure. *Oikos* 97: 237 - 250.

Oksanen, J., F.G. Blanchet, R. Kindt, P. Legendre, P.R. Minchin, R.B. O'Hara, G.L. Simpson, P. Solymos, M.H.H. Stevens and H. Wagner (2012). *vegan: Community Ecology Package*. R package version 2.0-4. <http://CRAN.R-project.org/package=vegan>

Examples

```
#define an interaction matrix
data(TestMatrices)
pres3c=TestMatrices[[6]]

#obtain an ordinated interaction matrix
OrderMatrix(pres3c, scores = 1, output.scores = FALSE)

#obtain site and species scores from reciprocal averaging
OrderMatrix(pres3c, scores = 1, output.scores = TRUE)
```

TestMatrices

Test matrices used to evaluate metacommunity functions

Description

A list of 7 test matrices from two of the foundational papers on the elements of metacommunity structure (Leibold & Mikkelsen 2002 and Presley et al. 2010)

Usage

```
data(TestMatrices)
```

Format

A list containing interaction matrices from Leibold & Mikkelsen 2002 and Presley et al. 2010:

```
[[1]] dim=(10 x 10) Leibold & Mikkelsen 2002 Figure 1a
[[2]] dim=(10 x 10) Leibold & Mikkelsen 2002 Figure 1b
[[3]] dim=(10 x 10) Leibold & Mikkelsen 2002 Figure 2a
[[4]] dim=(10 x 10) Leibold & Mikkelsen 2002 Figure 2b
[[5]] dim=(15 x 10) Leibold & Mikkelsen 2002 Figure 3c
[[6]] dim=(20 x 20) Presley et al. Figure 3c
[[7]] dim(20 x 20) Presley et al. Figure 4a
```

Source

Leibold, M. A., & Mikkelsen, G. M. (2002). Coherence, species turnover, and boundary clumping: elements of metacommunity structure. *Oikos*, 97(2), 237-250.

Presley, S. J., C. L. Higgins, and M. R. Willig. 2010. A comprehensive framework for the evaluation of metacommunity structure. *Oikos* 119:908-917

Examples

```
#load list containing interaction matrices
data(TestMatrices)

length(TestMatrices)
names(TestMatrices)

#image plot of interaction matrix, using the Imagine() function
lei1a=TestMatrices[[1]]
Imagine(lei1a)
```

Turnover

Determines species turnover

Description

'Turnover' is a function that assesses species turnover from the range perspective (traditional method). If the 'community' perspective is desired, simply transpose the matrix before analysis using the transpose function ('t()'), but make sure you understand the implications of this action, as the interpretation of the output changes dramatically.

Usage

```
Turnover(comm, method = "r1", sims = 1000, scores = 1,
order=TRUE, allow.empty=FALSE)
```

Arguments

comm	community data in the form of a presence absence matrix
method	null model randomization method used by 'nullmaker'. See details.
sims	number of simulated null matrices to use in analysis
scores	axis scores to ordinate matrix. 1: primary axis scores (default) 2: secondary axis scores
order	logical argument indicating whether to ordinate the interaction matrix or not. See details.
allow.empty	logical argument indicating whether to allow null matrices to have empty rows or columns

Details

'method' is an argument handed to functions in the 'vegan' package. Leibold & Mikkelsen advocated the use of equiprobable rows and columns (provided that rows and columns had at least one entry). This method is called 'r00'. Methods maintaining row (site) frequencies include 'r0', 'r1' & 'r2'. The default method argument is 'r1', which maintains the species richness of a site (row totals)

and fills species ranges (columns) based on their marginal probabilities. Arguably the most conservative null algorithm is the fixed row - fixed column total null, which is implemented as 'fixedfixed'. See the help file for 'commsimulator' or Wright et al. 1998 for more information.

If 'order' is FALSE, the interaction matrix is not ordinated, allowing the user to order the matrix based on site characteristics or other biologically relevant characteristics.

This function can either be used as a standalone, or can be used through the 'metacommunity()' function, which determines all 3 elements of metacommunity structure (coherence, boundary clumping, & turnover) (Leibold & Mikkelsen 2002)

Value

turnover	species turnover, quantified as the number of replacements across species ranges
z	the z statistic comparing observed turnover against species turnover of null matrices
pval	p-value from a z-test to assess significance of species turnover relative to null matrices
simulated mean	mean number of replacements in simulated null matrices
simulated variance	variance in the number of replacements of simulated null matrices
method	randomization method to create null matrices

Author(s)

Tad Dallas

References

Leibold, M.A. and G.M. Mikkelsen. 2002. Coherence, species turnover, and boundary clumping: elements of meta-community structure. *Oikos* 97: 237 - 250.

Wright, D.H., Patterson, B.D., Mikkelsen, G.M., Cutler, A. & Atmar, W. (1998). A comparative analysis of nested subset patterns of species composition. *Oecologia* 113, 1-20.

Examples

```
#define an interaction matrix
data(TestMatrices)
intmat=TestMatrices[[3]]

#determine species turnover
turnover.intmat=Turnover(intmat, method='r1', sims=1000, scores=1)

#return results in list form
turnover.intmat
```

Index

*Topic **datasets**

TestMatrices, [12](#)

*Topic **ordination**

Imagine, [6](#)

NullMaker, [9](#)

OrderMatrix, [11](#)

BoundaryClump, [3](#)

Coherence, [5](#)

Imagine, [6](#)

metacom (metacom-package), [2](#)

metacom-package, [2](#)

Metacommunity, [7](#)

NullMaker, [9](#)

OrderMatrix, [11](#)

TestMatrices, [12](#)

Turnover, [13](#)