# Package ‘fuzzySim’

September 19, 2014

**Type** Package

**Title** Fuzzy similarity in species distributions

**Version** 0.1

**Date** 2014-03-19

**Author** Barbosa A.M.

**Maintainer** A. Marcia Barbosa <barbosa@uevora.pt>

**Description** Functions to calculate fuzzy versions of species' distributions based on presence-absence data, and pair-wise fuzzy similarity among those distributions. Includes also functions for data preparation, such as obtaining unique abbreviations of species names, converting species lists (long format) to presence-absence tables (wide format), or transposing part of a data frame. Complete with a couple of sample data sets for providing practical examples.

**License** GPL-3

## R topics documented:

fuzzySim-package *Fuzzy similarity in species distributions*

### Description

Functions to calculate fuzzy versions of species' occurrence patterns based on presence-absence data, and pair-wise fuzzy similarity (based on fuzzy versions of commonly used similarity indices) among those occurrence patterns. Includes also functions for data preparation, such as obtaining unique abbreviations of species names, converting species lists (long format) to presence-absence tables (wide format), or transposing part of a data frame. Complete with a couple of sample data sets for providing practical examples.

### Details

| | |
|---|---|
| Package: | fuzzySim |
| Type: | Package |
| Version: | 0.1 |
| Date: | 2014-03-19 |
| License: | GPL-3 |

### Author(s)

A. Marcia Barbosa

Maintainer: A. Marcia Barbosa <barbosa@uevora.pt>

### References

Barbosa A.M. (submitted) A fuzzy implementation of binary similarity indices for assessing biogeographic associations, biotic regions, and beta diversity.

### See Also

**RMACOQUI** (Olivero, Real & Marquez 2011, Systematic Biology 60:645-60, doi 10.1093/sysbio/syr026)

### Examples

```
data(rotifers)

head(rotifers)


# add column with species name abbreviations:
```

```
rotifers$spcode <- spCodes(rotifers$species, sep.species = "_", nchar.gen = 1,
nchar.sp = 5, nchar.ssp = 0)

head(rotifers)


# convert species list (long format) to presence-absence table (wide format):

rotifers.presabs <- splist2presabs(rotifers, sites.col = "LEVEL3_COD",
sp.col = "spcode", keep.n = FALSE)

head(rotifers.presabs)


# get 3rd-degree spatial trend surface for each species' distribution:

data(rotifers01)

names(rotifers01)

rotifers.tsa <- multTSA(rotifers01, sp.cols = 6:40,
coord.cols = c("Long", "Lati"), id.col = 1)

head(rotifers.tsa)


# get fuzzy version of presence-absence using inverse squared
# distance to presence:

rotifers.dsq <- distPres(rotifers01, sp.cols = 6:ncol(rotifers01),
coord.cols = c("Long", "Lati"), id.col = 1, p = 2, inv = TRUE)

head(rotifers.dsq)


# get matrix of fuzzy similarity between rotifer species distributions:

rot.fuz.sim.mat <- simMat(rotifers.dsq[ , -1], method = "Baroni")

head(rot.fuz.sim.mat)


# transpose fuzzy rotifer distribution data to compare
# regional species composition rather than species' distributions:

names(rotifers.dsq)

rot.fuz.reg <- transpose(rotifers.dsq, sp.cols = 2:ncol(rotifers.dsq),
reg.names = 1)

head(rot.fuz.reg)
```

```
# get matrix of fuzzy similarity between (some) regions' species compositions:

reg.fuz.sim.mat <- simMat(rot.fuz.reg[ , 1:100], method = "Baroni")

head(reg.fuz.sim.mat)
```

## distPres *(Inverse) distance to the nearest presence*

### Description

This function takes a matrix or data frame containing species presence (1) and absence (0) data and their spatial coordinates (optionally also a pre-calculated distance matrix between all localities), and calculates the (inverse) distance from each locality to the nearest presence locality for each species.

### Usage

```
distPres(data, sp.cols, coord.cols = NULL, id.col = NULL, dist.mat = NULL,
method = "euclidian", suffix = "_D", p = 1, inv = TRUE)
```

### Arguments

data — a matrix or data frame containing, at least, two columns with spatial coordinates, and one column per species containing their presence (1) and absence (0) data, with localities in rows.

sp.cols — names or index numbers of the columns containing the species presences and absences in data. It must contain only zeros (0) for absences and ones (1) for presences.

coord.cols — names or index numbers of the columns containing the spatial coordinates in data (in this order, x and y, or longitude and latitude).

id.col — optionally, the name or index number of a column (to be included in the output) containing locality identifiers in data.

dist.mat — optionally, if you do not want distances calculated with any of the methods available in dist, you may provide a distance matrix calculated elsewhere for the localities in data.

method — the method with which to calculate distances between localities. Available options are those of dist. The default is "euclidian".

suffix — character indicating the suffix to add to the distance columns in the resulting data frame. The default is "_D".

p — the power to which distance should be raised. The default is 1; use 2 or higher if you want more conservative distances.

inv — logical value indicating whether distance should be inverted, so that it varies between 0 and 1 and higher values mean closer to presence. The default is TRUE, which is adequate as a fuzzy version of presence-absence (for using e.g. with fuzSim and simMat).

### Value

distPres returns a matrix or data frame containing the identifier column (if provided in id.col) and one column per species containing the distance (inverse squared by default) from each locality to the nearest presence of that species.

### Author(s)

A. Marcia Barbosa

### See Also

dist

### Examples

```
data(rotifers01)

head(rotifers01)

names(rotifers01)


# calculate plain distance to presence:

rotifers.dist <- distPres(rotifers01, sp.cols = 6:40,
coord.cols = c("Long", "Lati"), id.col = 1, p = 1, inv = FALSE, suffix = "_D")

head(rotifers.dist)


# calculate inverse squared distance to presence:

rotifers.invd2 <- distPres(rotifers01, sp.cols = 6:40,
coord.cols = c("Long", "Lati"), id.col = 1, p = 2, inv = TRUE, suffix = "_iDsq")

head(rotifers.invd2)
```

## fuzSim *Fuzzy similarity*

### Description

This function calculates the fuzzy similarity, based on a fuzzy version of the binary similarity index specified in method, between two binary or fuzzy variables.

### Usage

```
fuzSim(x, y, method)
```

### Arguments

`x` a vector of (optionally fuzzy) presence-absence data, with 1 meaning presence, 0 meaning absence, and values in between meaning fuzzy presence (or the degree to which each locality belongs to the set of species presences, or to which each species belongs to the locality; Zadeh, 1965). Fuzzy presence-absence can be obtained, for example, with functions `distPres` or `multTSA` in this package.

`y` a vector similar to `x`, of the same length and in the same order.

`method` the similarity index to use. Currently available options are `'Baroni'`, `'Jaccard'` and `'Sorensen'`.

### Details

Similarity between ecological communities, beta diversity patterns, biotic regions, and distributional relationships among species are commonly determined based on pair-wise (dis)similarities in species' occurrence patterns. Some of the most commonly employed pair-wise similarity indices are those of Jaccard (1901), Sorensen (1948) and Baroni-Urbani & Buser (1976), which are here implemented in their fuzzy versions (Barbosa, submitted), able to deal with both binary and fuzzy data.

### Value

The function returns a value between 0 and 1 representing the fuzzy similarity between `x` and `y`. Jaccard's similarity can be converted to dissimilarity (or Jaccard distance) if subtracted from 1. In contrast, 1-`Sorensen` is not a proper distance metric, as it lacks the property of triangle inequality.

### Note

The formulas used in this function may look slighty different from some of their published versions, but note that here the A and B are the numbers of attributes present in each element, whether or not they are also present in the other one. Thus, our 'A+B' is equivalent to 'A+B+C' in formulas where A and B are the numbers of attributes present in one but not the other element, and our A+B-C is equivalent to their A+B+C. The formulas used here (adapted from Olivero et al 1998) are faster to calculate, visibly for large datasets.

### Author(s)

A. Marcia Barbosa

### References

Baroni-Urbani C. & Buser M.W. (1976) Similarity of Binary Data. Systematic Zoology, 25: 251-259

Jaccard P. (1901) Etude comparative de la distribution florale dans une portion des Alpes et des Jura. Memoires de la Societe Vaudoise des Sciences Naturelles, 37: 547-579

Olivero J., Real R. & Vargas J.M. (1998) Distribution of breeding, wintering and resident waterbirds in Europe: biotic regions and the macroclimate. Ornis Fennica, 75: 153-175

Sorensen T. (1948) A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. Kongelige Danske Videnskabernes Selskab, 5(4): 1-34

Zadeh L.A. (1965) Fuzzy sets. Information and Control, 8: 338-353

### See Also

simMat

### Examples

```
data(rotifers01)

head(rotifers01)

names(rotifers01)

fuzSim(rotifers01[,6], rotifers01[,7], method = "Baroni")
fuzSim(rotifers01[,6], rotifers01[,7], method = "Jaccard")
fuzSim(rotifers01[,6], rotifers01[,7], method = "Sorensen")
```

## multTSA *Trend Surface Analysis for multiple species*

### Description

This function performs trend surface analysis for multiple species at a time. It converts categorical presence-absence (1-0) data into continuous surfaces denoting the spatial trend in species' occurrence patterns.

### Usage

```
multTSA(data, sp.cols, coord.cols, id.col = NULL, degree = 3, step = TRUE,
Favourability = FALSE, suffix = "_TS", save.models = FALSE)
```

### Arguments

| | |
|---|---|
| data | a matrix or data frame containing, at least, two columns with spatial coordinates, and one column per species containing their presence (1) and absence (0) data, with localities in rows. |
| sp.cols | names or index numbers of the columns containing the species presences and absences in data. Must contain only zeros (0) for absences and ones (1) for presences. |
| coord.cols | names or index numbers of the columns containing the spatial coordinates in data (in this order, x and y, or longitude and latitude). |
| id.col | optionally, the name or index number of a column (to be included in the output) containing locality identifiers in data. |

degree — the degree of the spatial polynomial to use (see Details). The default is 3.

step — logical value indicating whether the regression of presence-absence on the spatial polynomial should do a stepwise inclusion of the polynomial terms (using the step function with default settings, namely backward AIC selection), rather than forcing all terms into the equation. The default is TRUE.

Favourability — logical value indicating whether the probability values obtained from the regression should be converted to favourability, so that they are more directly comparable among species with different prevalence (see Real et al., 2006). The default is FALSE.

suffix — character indicating the suffix to add to the trend surface columns in the resulting data frame. The default is "_TS".

save.models — logical value indicating whether the models obtained from the regressions should be saved in the results. The default is FALSE.

## Details

Trend Surface Analysis is a way to model the spatial structure in species' distributions by regressing occurrence data on the spatial coordinates x and y, for a linear trend, or on polynomial terms of these coordinates (x^2, y^2, x*y, etc.), for curvilinear trends (Legendre & Legendre, 1998; Borcard et al., 2011). Second- and third-degree polynomials are often used. multTSA allows specifying the degree of the spatial polynomial to use. By default, it uses a 3rd-degree polynomial and performs stepwise AIC selection of the polynomial terms to include.

## Value

A matrix or data frame containing the identifier column (if provided in id.col) and one column per species containing the value predicted by the trend surface analysis.

## Author(s)

A. Marcia Barbosa

## References

Borcard D., Gillet F. & Legendre P. (2011) Numerical Ecology with R. Springer, New York.

Legendre P. & Legendre L. (1998) Numerical Ecology. Elsevier, Amsterdam.

Real R., Barbosa A.M. & Vargas J.M. (2006) Obtaining environmental favourability functions from logistic regression. Environmental and Ecological Statistics, 13: 237-245

## See Also

distPres, poly

### Examples

```
data(rotifers01)

head(rotifers01)

names(rotifers01)

tsa <- multTSA(rotifers01, sp.cols = 6:40, coord.cols = c("Long", "Lati"),
id.col = 1)

head(tsa)
```

## rotifers *Rotifer species on TDWG level 3 regions of the world*

### Description

These data were extracted from a database of monogonont rotifer species records on the geographical units used by the Biodiversity Information Standards (formerly Taxonomic Database Working Group, TDWG; base maps available at http://www.kew.org/science-research-data/kew-in-depth/gis/resources-and- publications/data/tdwg/index.htm). The original data were compiled and published by Fontaneto et al. (2012) for all TDWG levels. Here they are reduced to the TDWG - level 3 units and to the species recorded in at least 80 of these units. Mind that this is not a complete picture of these species' distributions, due to insufficient sampling in many regions.

### Usage

```
data(rotifers)
```

### Format

A data frame with 3640 observations on the following 2 variables.

LEVEL3_COD a factor with 201 levels corresponding to the code names of the TDWG level 3 regions on which the records were taken

species a factor with 35 levels corresponding to the names of the (sub)species recorded on at least 80 different TDWG level 3 regions

### Source

Fontaneto D., Barbosa A.M., Segers H. & Pautasso M. (2012) The 'rotiferologist' effect and other global correlates of species richness in monogonont rotifers. Ecography, 35: 174-182.

### Examples

```
data(rotifers)

head(rotifers, 10)
```

rotifers01 *Rotifer presence-absence on TDWG-level3 regions of the world*

## Description

These data were extracted from a database of monogonont rotifer species presence records on the geographical units used by the Biodiversity Information Standards (formerly Taxonomic Database Working Group, TDWG; base maps available at http://www.kew.org/science-research-data/kew-in-depth/gis/resources-and- publications/data/tdwg/index.htm). The original data were compiled and published by Fontaneto et al. (2012) in long (narrow, stacked) format. Here they are presented in wide or unstacked format (presence-absence table, obtained with splist2presabs), reduced to the species recorded in at least 80 different TDWG - level 3 units, and with abbreviations of the species' names (obtained with spCodes). Mind that this is not a complete picture of these species' distributions, due to insufficient sampling in many regions.

## Usage

```
data(rotifers01)
```

## Format

Data frame with 369 rows (TDWG - level 3 regions of the world) and 40 columns:LEVEL3_COD, ID and LEVEL_NAME (from the TDWG 3 map attribute table); Lati and Long (mean latitude and mean longitude of each region); and 35 rotifer species (from Abrigh on), with names abbreviated to first letter of genus + first five letters of specific name.

## Source

Fontaneto D., Barbosa A.M., Segers H. & Pautasso M. (2012) The 'rotiferologist' effect and other global correlates of species richness in monogonont rotifers. Ecography, 35: 174-182.

## Examples

```
data(rotifers01)

head(rotifers01)
```

## simMat *Pair-wise (fuzzy) similarity matrix*

### Description

simMat takes a matrix or data frame containing species occurrence data, either categorical (0 or 1) or fuzzy (between 0 and 1), and uses the fuzSim function to calculate a square matrix of pairwise similarities between them, using a fuzzy logic version (after Zadeh, 1965) of either Jaccard's (1901), Baroni-Urbani & Buser's (1976) or Sorensen's (1948) binary similarity index.

### Usage

```
simMat(data, method)
```

### Arguments

| | |
|---|---|
| data | a matrix or data frame containing (optionally fuzzy) species presence-absence data (one column per species), with 1 meaning presence, 0 meaning absence, and values in between for fuzzy presence (or the degree to which each locality belongs to the set of species presences; see Zadeh, 1965). Fuzzy presence - absence can be obtained, for example, with distPres or with multTSA. |
| method | the similarity index whose fuzzy version to use. Available options are 'Baroni', 'Jaccard' and 'Sorensen'. |

### Details

Spatial associations between species' distributions can provide deep insights into the processes that drive biodiversity patterns. Chorological clustering provides a systematic framework for analysing such associations, by detecting areas with similar species composition (biotic regions) or clusters of species' distribution types (chorotypes; Olivero et al. 2011, 2013). The fuzzy versions of species occurrence data and of the similarity indices introduce tolerance for small spatial differences in species' occurrence localities and may also compensate geo-referencing errors. The results of simMat can be used for determining chorotypes or biotic regions using the RMACOQUI package (Olivero et al. 2011).

### Value

A square matrix of pair-wise similarity among the species distributions (columns) in data. Similarity is calculated with the fuzzy version of the index specified in method, which yields traditional binary similarity if the data are binary (0 or 1), or fuzzy similarity if the data are fuzzy (between 0 and 1).

### Author(s)

A. Marcia Barbosa

## References

Baroni-Urbani, C. & Buser, M.W. (1976) Similarity of Binary Data. Systematic Zoology, 25: 251-259.

Jaccard, P. (1901) Etude comparative de la distribution florale dans une portion des Alpes et des Jura. Memoires de la Societe Vaudoise des Sciences Naturelles, 37: 547-579.

Olivero, J., Real, R. & Marquez, A.L. (2011) Fuzzy chorotypes as a conceptual tool to improve insight into biogeographic patterns. Systematic Biology, 60: 645-60.

Olivero, J., Marquez, A.L., Real, R. (2013) Integrating fuzzy logic and statistics to improve the reliable delimitation of biogeographic regions and transition zones. Systematic Biology, 62: 1-21.

Sorensen T. (1948) A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. Kongelige Danske Videnskabernes Selskab 5(4): 1-34

Zadeh L.A. (1965) Fuzzy sets. Information and Control, 8: 338-353.

## See Also

fuzSim

## Examples

```
# load and look at the rotifers01 presence-absence data:

data(rotifers01)

head(rotifers01)

names(rotifers01)


# build a matrix of similarity among these binary data
# using e.g. Baroni-Urbani & Buser's index:

bin.sim.mat <- simMat(rotifers01[ , 6:40], method = "Baroni")

head(bin.sim.mat)


# calculate a fuzzy version of the presence-absence data
# based on inverse distance to presences:

rotifers.invd <- distPres(rotifers01, sp.cols = 6:40,
coord.cols = c("Long", "Lati"), id.col = 1, suffix = ".d", p = 1, inv = TRUE)

head(rotifers.invd)


# build a matrix of fuzzy similarity among these fuzzy distribution data,
# with the fuzzy version of Baroni-Urbani & Buser's index:
```

```
fuz.sim.mat <- simMat(rotifers.invd[ , -1], method = "Baroni")

head(fuz.sim.mat)


# plot the similarity matrices as colours:

image(x = 1:ncol(bin.sim.mat), y = 1:nrow(bin.sim.mat), z = bin.sim.mat,
col = rev(heat.colors(256)), xlab = "", ylab = "", axes = FALSE,
main = "Binary similarity")
axis(side = 1, at = 1:ncol(bin.sim.mat), tick = FALSE,
labels = colnames(bin.sim.mat), las = 2)
axis(side = 2, at = 1:nrow(bin.sim.mat), tick = FALSE,
labels = rownames(bin.sim.mat), las = 2)

image(x = 1:ncol(fuz.sim.mat), y = 1:nrow(fuz.sim.mat), z = fuz.sim.mat,
col = rev(heat.colors(256)), xlab = "", ylab = "", axes = FALSE,
main = "Fuzzy similarity")
axis(side = 1, at = 1:ncol(fuz.sim.mat), tick = FALSE,
labels = colnames(fuz.sim.mat), las = 2, cex = 0.5)
axis(side = 2, at = 1:nrow(fuz.sim.mat), tick = FALSE,
labels = rownames(fuz.sim.mat), las = 2)


# plot a UPGMA dendrogram from each similarity matrix:

plot(hclust(as.dist(1 - bin.sim.mat), method = "average"),
main = "Binary cluster dendrogram")

plot(hclust(as.dist(1 - fuz.sim.mat), method = "average"),
main = "Fuzzy cluster dendrogram")


# you can get fuzzy chorotypes from these similarity matrices
# (or fuzzy biotic regions if your transpose \code{data} so that localities
# are in columns and species in rows) using the \code{RMACOQUI} package
# (Olivero et al. 2011)
```

## spCodes *Obtain unique abbreviations of species names*

### Description

This function takes a vector of species names and converts them to abbreviated species codes containing the specified numbers of characters from the genus, the specific and optionally also the subspecific name. Separators can be specified by the user. The function checks that the resulting codes are unique.

### Usage

```
spCodes(species, nchar.gen = 3, nchar.sp = 3, nchar.ssp = 0, sep.species = " ",
sep.spcode = "")
```

### Arguments

species a character vector containig the species names to be abbreviated.

nchar.gen the number of characters from the genus name to be included in the resulting species code.

nchar.sp the number of characters from the specific name to be included in the resulting species code.

nchar.ssp optionally, the number of characters from the subspecific name to be included in the resulting species code. Set it to 0 if you have subspecific names in species but do not want them included in the resulting species codes.

sep.species the character separating genus, specific and subspecific names in species. The default is a white space.

sep.spcode the character you want separating genus and species abbreviations in the resulting species codes. The default is an empty character (no separator).

### Value

A character vector containing the species codes resulting from the abbreviation. If the numbers of characters specified do not make for unique codes, an error message is displayed showing which species names caused it, so that you can try again with different nchar.gen, nchar.sp and/or nchar.ssp.

### Author(s)

A. Marcia Barbosa

### See Also

substr, strsplit

### Examples

```
data(rotifers)

head(rotifers)

## add a column to 'rotifers' with shorter versions of the species names:

# rotifers$spcode <- spCodes(rotifers$species, sep.species = "_", nchar.gen = 1,
# nchar.sp = 4, nchar.ssp = 0, sep.spcode = ".")
# this would produce an error due to resulting species codes not being unique
# (try it out)

rotifers$spcode <- spCodes(rotifers$species, sep.species = "_", nchar.gen = 1,
nchar.sp = 5, nchar.ssp = 0, sep.spcode = ".")
```

```
# with a larger number of characters from the specific name,
# resulting codes are now unique

## check out the result:
head(rotifers)
```

## splist2presabs *Convert a species list to a presence-absence table*

### Description

This function takes a locality+species dataset in long (stacked) format, i.e., a matrix or data frame containing localities in one column and their recorded species in another column, and converts them to a presence-absence table (wide format) suitable for mapping and for computing distributional similarities (see e.g. simMat). Try out the Examples below for an illustration).

### Usage

```
splist2presabs(data, sites.col, sp.col, keep.n = FALSE)
```

### Arguments

| | |
|---|---|
| data | a matrix or data frame with your localities in one columns and species in another column. See data(rotifers) for an example. |
| sites.col | the name or index number of the column containing the localities in data. |
| sp.col | the name or index number of the column containing the species names or codes in data. |
| keep.n | logical value indicating whether to get in the resulting table the number of times each species appears in each locality; if false (the default), only the presence (1) or absence (0) are recorded. |

### Value

A data frame containing the localities in the first column and then one column per species indicating their presence (or their number of records if keep.n = TRUE) and absence. See data(rotifers01) for an example.

### Author(s)

A. Marcia Barbosa

### See Also

table

### Examples

```
data(rotifers)

head(rotifers)

rotifers.presabs <- splist2presabs(rotifers, sites.col = "LEVEL3_COD",
sp.col = "species", keep.n = FALSE)

head(rotifers.presabs)
```

## transpose *Transpose (part of) a matrix or dataframe*

### Description

This function transposes (a specified part of) a matrix or data frame, optionally using one of its columns as column names for the transposed result. It is useful for turning a species presence-absence table into a regional species composition table.

### Usage

```
transpose(data, sp.cols = 1:ncol(data), reg.names = NULL)
```

### Arguments

data a matrix or data frame containing the species occurrence data to transpose.

sp.cols names or index numbers of the columns containing the species occurrences in data which are meant to be transposed.

reg.names name or index number of the column in data containing the region names, to be used as column names in the transposed result.

### Value

The transposed sp.cols of data, with the column specified in reg.names as column names.

### Author(s)

A. Marcia Barbosa

### See Also

t

### Examples

```
data(rotifers01)

head(rotifers01)

names(rotifers01)

rotif.reg <- transpose(rotifers01, sp.cols = 6:40, reg.names = 1)

head(rotif.reg)
```

# Index