Package 'multifunc'

May 9, 2026

Version 0.9.4

Date 2022-05-23

Title Analysis of Ecological Drivers on Ecosystem Multifunctionality

Description Methods for the analysis of how ecological drivers affect the multifunctionality of an ecosystem based on methods of Byrnes et al. 2016 <doi:10.1111/2041-210X.12143> and Byrnes et al. 2022 <doi:10.1101/2022.03.17.484802>. Most standard methods in the literature are implemented (see vignettes) in a tidy format.

URL https://jebyrnes.github.io/multifunc/, https://github.com/jebyrnes/multifunc

BugReports https://github.com/jebyrnes/multifunc/issues

Imports stats, dplyr, purrr, broom, MASS, utils, magrittr

Suggests ggplot2, forcats, tidyr, gridExtra, knitr, patchwork, car

License MIT + file LICENSE

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Jarrett Byrnes [aut, cre, cph]

Maintainer Jarrett Byrnes <jarrett.byrnes@umb.edu>

Repository CRAN

Date/Publication 2022-05-25 07:30:09 UTC

Contents

`all_biodepth` *Biodepth Data*

Description

Data from the pan-European Biodepth grassland diversity manipulation

Author(s)

Jarrett Byrnes

References

Spehn, E. M., A. Hector, J. Joshi, M. Scherer-Lorenzen, B. Schmid, E. Bazeley-White, C. Beierkuhnlein, M. C. Caldeira, M. Diemer, P. G. Dimitrakopoulos, J. A. Finn, H. Freitas, P. S. Giller, J. Good, R. Harris, P. Hogberg, K. Huss-Danell, A. Jumpponen, J. Koricheva, P. W. Leadley, M. Loreau, A. Minns, C. P. H. Mulder, G. O'Donovan, S. J. Otway, C. Palmborg, J. S. Pereira, A. B. Pfisterer, A. Prinz, D. J. Read, E. D. Schulze, A. S. Siamantziouras, A. C. Terry, A. Y. Troumbis, F. I. Woodward, S. Yachi, and J. H. Lawton. 2005. Ecosystem effects of biodiversity manipulations in European grasslands. Ecological Monographs 75:37-63.

cor_dist *cor_dist*

Description

Takes a data frame of functions and calculates the correlation-based distance between functions.

Usage

```
cor_dist(adf)
```

Arguments

adf	A data.frame or matrix of functions

Value

A matrix

divNeeded *divNeeded*

Description

divNeeded Determines, for every combination of functions, how many species influence those functions.

Usage

```
divNeeded(overData, type = "positive")
```

Arguments

overData	Matrix of functions and which species affect them from getRedundancy.
type	Are the kinds of effects we're looking at "positive", "negative" or "all".

Details

Iterates over all possible combinations of functions. Checks the matrix of which species have positive, negative, or both influences on those functions. Tally's total number of species that have an effect on those functions

Value

Returns a data frame of all combinations and how many species are needed to influence all of them.

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)
species <- relevantSp(germany, 26:ncol(germany))

# re-normalize N.Soil so that everything is on the
# same sign-scale (e.g. the maximum level of a
# function is the "best" function)
germany$N.Soil <- -1 * germany$N.Soil + max(germany$N.Soil, na.rm = TRUE)

res.list <- lapply(vars, function(x) sAICfun(x, species, germany))
names(res.list) <- vars

redund <- getRedundancy(vars, species, germany)

posCurve <- divNeeded(redund, type = "positive")
```

---

dmean *dmean*

Description

Calculates the average distance between functions for one or an entire assemblage of replicates

Usage

```
dmean(adf_raw, D)
```

Arguments

adf_raw	A data frame frame with functions in columns and rows as replicates
D	A distance matrix describing dissimilarity between functions.

Value

Single numeric of weighted average of distance matrix

References

Byrnes, J. E. K., Roger, F. and Bagchi, R. 2022. Understandable Multifunctionality Measures Using Hill Numbers. bioRxiv. 2022.03.17.484802. https://doi.org/10.1101/2022.03.17.484802

Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C., Chiang, J.-M. and Sherwin, W. B. 2019. An attribute-diversity approach to functional diversity, functional beta diversity, and related (dis)similarity measures. Ecological Monographs. 89: e01343.

`dmin` *dmin*

Description

Calculates the minimum non-zero value of a distance matrix

Usage

```
dmin(D)
```

Arguments

`D`	A distance matrix describing dissimilarity between functions.

Value

A numeric

`duffy_2003` *Seagrass Mesocosm Data*

Description

Data from the a seagrass grazer diversity manipulation at the Virginia Institute of Marine Sciences. From Duffy et al. 2003 Ecology Letters.

Author(s)

Jarrett Byrnes

References

Duffy, J. E., J. P. Richardson, and E. A. Canuel. 2003. Grazer diversity effects on ecosystem functioning in seagrass beds. Ecology letters 6:637-645.

eff_num_func *eff_num_func*

Description

Calculate the effective number of functions for rows in a dataset

Usage

```
eff_num_func(dat, vars, q = 1, standardized = FALSE, D = NULL, tau = NULL)
```

Arguments

dat	A data frame with functions in columns and rows as replicates as well as other information.
vars	Column names of function variables
q	Order of the diversity measure. Defaults to the Shannon case where q = 1. For Simpson, q=2.
standardized	Use standardized number of functions (scaled by total number of functions, so between 0-1), or just raw effective number of functions for calculation. Defaults to FALSE.
D	A distance matrix describing dissimilarity between functions. Defaults to NULL, and the index is calculated assuming all functions are different. If it is not null, it must be a symmetric matrix with dimensions matching the number of functions listed in vars.
tau	A cutoff for degree of dissimilarity under which functions are considered to be different. If tau is the minimum non-zero value of D, all functions are different. if tau is the maximum value of D are greater, all functions are considered the same.

Details

Takes a data frame, variable names, whether we want an index standardized by number of functions or not, an order of Hill number for our effective number of functions as well as a dissimilarity matrix (if desired) and value for a dissimilarity cutoff (defaults to the average dissimilarity). It then calculates and returns the effective number of functions using the appropriate method. See Chao et al. 2019 for more.

Value

Returns a vector of effective or standardized effective number of functions

References

Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C., Chiang, J.-M. and Sherwin, W. B. 2019. An attribute-diversity approach to functional diversity, functional beta diversity, and related (dis)similarity measures. Ecological Monographs. 89: e01343.

Jost, L. 2006. Entropy and diversity. Oikos 113(2): 363-375.

Hill, M. 1973. Diversity and evenness: A unifying notation and its consequences. Ecology 54: 427-432.

`eff_num_func_d` *eff_num_func_d*

Description

eff_num_func_d

Usage

```
eff_num_func_d(adf_freq, q = 1, D, tau = NULL)
```

Arguments

`adf_freq`	A data frame of functional "frequencies" - i.e. f_i/sum(f_i)
`q`	Order of hill number used for index. Defaults to q=1, as in Shannon Diversity
`D`	A distance matrix describing dissimilarity between functions.
`tau`	A cutoff for degree of dissimilarity under which functions are considered to be different. If tau is the minimum non-zero value of D, all functions are different. if tau is the maximum value of D are greater, all functions are considered the same.

Value

A vector of effective number of functions

References

Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C., Chiang, J.-M. and Sherwin, W. B. 2019. An attribute-diversity approach to functional diversity, functional beta diversity, and related (dis)similarity measures. Ecological Monographs. 89: e01343.

eff_num_func_d_onerow *eff_num_func_d_onerow*

Description

eff_num_func_d_onerow

Usage

```
eff_num_func_d_onerow(arow_freq, D, tau, q)
```

Arguments

arow_freq	One replicate sample of different functions (a single numeric vector)
D	A distance matrix describing dissimilarity between functions.
tau	A cutoff for degree of dissimilarity under which functions are considered to be different. If tau is the minimum non-zero value of D, all functions are different. if tau is the maximum value of D are greater, all functions are considered the same.
q	Order of hill number used for index. Defaults to q=1, as in Shannon Diversity

Value

A single value of effective number of functions

References

Byrnes, J. E. K., Roger, F. and Bagchi, R. 2022. Understandable Multifunctionality Measures Using Hill Numbers. bioRxiv. 2022.03.17.484802. https://doi.org/10.1101/2022.03.17.484802

Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C., Chiang, J.-M. and Sherwin, W. B. 2019. An attribute-diversity approach to functional diversity, functional beta diversity, and related (dis)similarity measures. Ecological Monographs. 89: e01343.

eff_num_func_no_d *eff_num_func_no_d*

Description

eff_num_func_no_d

Usage

```
eff_num_func_no_d(adf_freq, q = 1)
```

Arguments

adf_freq	A data frame of functional "frequencies" - i.e. f_i/sum(f_i)
q	Order of hill number used for index. Defaults to q=1, as in Shannon Diversity

Details

Takes a data frame, with functions standardized against total level of function in their replicate as columns and replicates as rows. Returns the effective number of functions using the appropriate method. See Chao et al. 2019 or Jost 2006 for details. Does not adjust for correlation between functions.

Value

Returns a verctor of effective or standardized effective number of functions

References

Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C., Chiang, J.-M. and Sherwin, W. B. 2019. An attribute-diversity approach to functional diversity, functional beta diversity, and related (dis)similarity measures. Ecological Monographs. 89: e01343.

Jost, L. 2006. Entropy and diversity. Oikos 113(2): 363-375.

Hill, M. 1973. Diversity and evenness: A unifying notation and its consequences. Ecology 54: 427-432.

---

filterCoefData *filterCoefData*

Description

filterCoefData filters contributions of species to function by sign.

Usage

```
filterCoefData(coefData, type = "positive")
```

Arguments

coefData	Matrix of functions and coefficients for which species affect them from getRedundancy.
type	Are the kinds of effects we're looking at "positive", "negative" or "all".

Details

Takes a matrix of functions and coefficients for species and filters out only the sign of contributions desired. Typically used by other functions in the package.

Value

Returns a filtered matrix.

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)
species <- relevantSp(germany, 26:ncol(germany))

# re-normalize N.Soil so that everything is on the same
# sign-scale (e.g. the maximum level of a function is
# the "best" function)
germany$N.Soil <- -1 * germany$N.Soil + max(germany$N.Soil, na.rm = TRUE)

res.list <- lapply(vars, function(x) sAICfun(x, species, germany))
names(res.list) <- vars

coefs <- getRedundancy(vars, species, germany, output = "coef")
stdCoefs <- stdEffects(coefs, germany, vars, species)

filterCoefData(stdCoefs)


#########
# filterCoefData takes a matrix of coefficients
# and filters it so that only the positive, negative, or both contributions
# are present
#########
```

`filterOverData` *filterOverData*

Description

`filterOverData` filters qualitative effects of species to function by sign.

Usage

```
filterOverData(overData, type = "positive")
```

Arguments

overData	Matrix of functions and which species affect them from getRedundancy.
type	Are the kinds of effects we're looking at "positive", "negative" or "all".

Details

Takes a matrix of functions and effects of species - 1's and -1's, s - and filters out only the sign of contributions desired. Typically used by other functions in the package.

Value

Returns a filtered matrix.

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)
species <- relevantSp(germany, 26:ncol(germany))

# re-normalize N.Soil so that everything is on the same
# sign-scale (e.g. the maximum level of a function is the
# "best" function)
germany$N.Soil <- -1 * germany$N.Soil + max(germany$N.Soil, na.rm = TRUE)

res.list <- lapply(vars, function(x) sAICfun(x, species, germany))
names(res.list) <- vars

redund <- getRedundancy(vars, species, germany)

filterOverData(redund, type = "positive")


#########
# filterOverData takes a matrix of 1s, 0s, and -1s
# and filters it so that only the positive, negative, or both contributions
# are 1 for later overlap function usage
#########
```

getCoefTab *getCoefTab*

Description

getCoefTab extract the effect of diversity on number of functions greater than a threshold

Usage

```
getCoefTab(
  eqn,
  fun = stats::glm,
  data,
  groupVar = "thresholds",
  coefVar = NULL,
  ...
)
```

Arguments

eqn	The model to be fit at each threshold.
fun	The fitting function. Defaults to glm.
data	A data frame containing the variables in the model to be fit.
groupVar	Grouping variable. Defaults to "thresholds" to fit the model at different thresholds, but, other types of grouping are possible.
coefVar	The name of the variable from the model whose coefficient we'll be extracting.
...	Other arguments to be supplied to the fitting function

Details

getCoefTab Takes a statistical model and plot level data with the number of functions greater than a threshold at multiple different thresholds and returns the coefficient for the effect of diversity at each threshold

Value

Returns a data frame of thresholds, coefficients, and their statistical properties.

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)

# re-normalize N.Soil so that everything is on the same
# sign-scale (e.g. the maximum level of a function is
# the "best" function)
germany$N.Soil <- -1 * germany$N.Soil + max(germany$N.Soil, na.rm = TRUE)

germanyThresh <- getFuncsMaxed(germany, vars,
  threshmin = 0.05,
  threshmax = 0.99, prepend = c("plot", "Diversity"), maxN = 7
)

germanyLinearSlopes <- getCoefTab(funcMaxed ~ Diversity,
  data = germanyThresh, coefVar = "Diversity", family = quasipoisson(link = "identity")
)
```

---

getFuncMaxed *getFuncMaxed*

Description

getFuncMaxed the number of functions greater than or equal to a single threshold in one experimental unit

Usage

```
getFuncMaxed(
  adf,
  vars = NA,
  thresh = 0.7,
  proportion = FALSE,
  prepend = "Diversity",
  maxN = 1
)
```

Arguments

adf	A data frame with functions.
vars	The column names of the functions to be assessed.
thresh	The threshold value to assess.

proportion	Whether the output will be returned as a proportion of all functions. Defaults to FALSE.
prepend	Additional columns that will be imported from the data for the returned data frame.
maxN	As a 'maximum' value can be subject to outliers, etc., what number of the highest data points for a function will be used to calculate the value against which thresholds will be judged. E.g., if maxN=1 then all thresholds are proportions of the largest value measured for a function. If maxN=8, then it's the proportion of the mean of the highest 8 measurements.

Details

Create a data frame that has the value of number or proportion of functions greater than a single threshold.

Value

Returns a data frame of number or fraction of functions greater than or equal to the selected thresholds in each plot.

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)

# re-normalize N.Soil so that everything is on the same
# sign-scale (e.g. the maximum level of a function is
# the "best" function)
germany$N.Soil <- -1 * germany$N.Soil + max(germany$N.Soil, na.rm = TRUE)

germanyThresh <- getFuncMaxed(germany, vars,
                              thresh = 0.5,
                              prepend = c("plot", "Diversity"),
                              maxN = 7)


# A function that will return a data frame with the first several columns
# being information the user wants for identification purposes (prepend)
# which defaults to Diversity and the final column the number of columns
# which pass a predefined threshold, defined as some proportion of the maximim
# observed for each column.  vars=the names of the vars being specified
# thresh is the threshold, between 0 and 1, of proportion of the max that needs
# to be passed to be counted.
```

```
# changelog
# 2014-03-24 Fixed -1 error in getMaxValue
# 2015-06-24 Fixed column name from prepend error https://github.com/jebyrnes/multifunc/issues/1
# 2022-04-14 Updated to use dplyr
```

---

getFuncsMaxed *getFuncsMaxed*

Description

getFuncsMaxed the number of functions greater than or equal to a wide variety of thresholds in each experimental unit

Usage

```
getFuncsMaxed(
  adf,
  vars = NA,
  threshmin = 0.05,
  threshmax = 0.99,
  threshstep = 0.01,
  proportion = FALSE,
  prepend = "Diversity",
  maxN = 1
)
```

Arguments

adf	A data frame with functions.
vars	The column names of the functions to be assessed.
threshmin	The lowest threshold value to assess.
threshmax	The highest threshold value to assess
threshstep	The incremental steps between lowest and highest thresholds to be assessed. See seq.
proportion	Whether the output will be returned as a porportion of all functions. Defaults to FALSE.
prepend	Additional columns that will be imported from the data for the returned data frame.
maxN	As a 'maximum' value can be subject to outliers, etc., what number of the highest data points for a function will be used to calculate the value against which thresholds will be judged. E.g., if maxN=1 then all thresholds are porportions of the largest value measured for a function. If maxN=8, then it's the porportion of the mean of the highest 8 measurements.

Details

Create a data frame that has the value of number or proportion of functions greater than a threshold for several different thresholds at the plot.

Value

Returns a data frame of number or fraction of functions greater than or equal to the selected thresholds in each plot over all thresholds within the relevant range.

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)

# re-normalize N.Soil so that everything is on the same
# sign-scale (e.g. the maximum level of a function is
# the "best" function)
germany$N.Soil <- -1 * germany$N.Soil +
                  max(germany$N.Soil, na.rm = TRUE)

germanyThresh <- getFuncsMaxed(germany, vars,
  threshmin = 0.50,
  threshmax = 0.60, prepend = c("plot", "Diversity"), maxN = 7
)
```

`getIndices` *getIndices*

Description

`getIndices` Generates a variety of indices describing multifunctionality based on the number of functions greater than a threshold for many different threshold and coefficients describing the relationship between diversity and number of functoins greater than a threshold.

Usage

```
getIndices(
  slopedata,
  threshdata,
  eqn,
  fun = stats::glm,
  divvar = "Diversity",
  groupVar = "thresholds",
  showNfunc = TRUE
)
```

Arguments

slopedata	A data frame with slopes of the relationship between diversity and number of functions greather than or equal to a threshold from getCoefTab.
threshdata	A data frame with the number of functions greater than a threshold for each plot at each threshold from getFuncsMaxed.
eqn	The formula used for fitting the models in slopedata.
fun	The function used to refit the threshold data at key points to get intercepts, etc., that are needed for the table.
divvar	The name of the variable that has the measure of diversity or other driver in the threshdata data frame.
groupVar	The name of a variable by which data is grouped in the threshdata data frame. Typically "thresholds" from getFuncsMaxed.
showNfunc	Show the functions at Tmin, Tmax, and Tmde. Defaults to TRUE.

Details

See Byrnes et al. In Review.

Value

A data frame of indices

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)

germanyThresh <- getFuncsMaxed(germany, vars,
```

```
  threshmin = 0.05,
  threshmax = 0.99, prepend = c("plot", "Diversity"), maxN = 7
)

germanyLinearSlopes <- getCoefTab(funcMaxed ~ Diversity,
  data = germanyThresh,
  coefVar = "Diversity", family = quasipoisson(link = "identity")
)

getIndices(germanyLinearSlopes, germanyThresh, funcMaxed ~ Diversity)
```

---

getMF_eff *getMF_eff*

Description

A multifunctionality index rooted in Hill numbers. `getMF_eff` get multifunctionality index defined by function and effective number of functions

Usage

```
getMF_eff(
  data,
  vars,
  q = 1,
  standardized = FALSE,
  standardize_function = standardizeUnitScale,
  D = NULL,
  tau = NULL
)
```

Arguments

`data`	A data frame with functions in columns and rows as replicates as well as other information.
`vars`	Name of function variables
`q`	Order of the diversity measure. Defaults to the Shannon case where q = 1. For Simpson, q=2.
`standardized`	Use standardized number of functions (scaled by total number of functions, so between 0-1), or just raw effective number of functions for calculation. Defaults to `FALSE`.
`standardize_function`	A function to standardize each individual function to the same scale, such as `standardizeUnitScale` or `standardizeZScore`

D — A distance matrix describing dissimilarity between functions. Defaults to NULL, and the index is calculated assuming all functions are different. If it is not null, it must be a symmetric matrix with dimensions matching the number of functions listed in vars.

tau — A cutoff for degree of dissimilarity under which functions are considered to be different. If tau is the minimum non-zero value of D, all functions are different. if tau is the maximum value of D are greater, all functions are considered the same.

Details

Takes a data frame, variable names, a standardizing function, whether we want an index standardized by number of functions or not, an order of Hill number for our effective number of functions as well as a dissimilarity matrix (if desired) and value for a dissimilarity cutoff (defaults to the average dissimilarity). It then calculates both the average standardized function in each plot and the effective number of functions and returns their product as a measure of effective multifunctionality.

Value

Returns a vector of effective or standardized effective multifunctionality.

Author(s)

Jarrett Byrnes.

References

Chao, A., Chiu, C.-H., Villéger, S., Sun, I.-F., Thorn, S., Lin, Y.-C., Chiang, J.-M. and Sherwin, W. B. 2019. An attribute-diversity approach to functional diversity, functional beta diversity, and related (dis)similarity measures. Ecological Monographs. 89: e01343.

Jost, L. 2006. Entropy and diversity. Oikos 113(2): 363-375.

Hill, M. 1973. Diversity and evenness: A unifying notation and its consequences. Ecology 54: 427-432.

---

getOverlap *getOverlap*

Description

getOverlap goes through all m-wise combinations of species and returns the amount of overlap between species in functions they perform for each combination

Usage

```
getOverlap(
  overData,
  m = 2,
  type = "positive",
  index = "sorensen",
  denom = "set"
)
```

Arguments

overData	Matrix of functions and which species affect them from getRedundancy.
m	Number of functions. Defaults to 2.
type	Are the kinds of effects we're looking at "positive", "negative" or "all".
index	Type of overlap index to be used. Defaults to "sorenson" but currently incorporates "mountford" and "jaccard" as well.
denom	Should the denominator be "all" species or just the "set" of species with the types of interactions being considered? Defaults to "set".

Details

getOverlap takes a matrix of 1s and -1s, and depending on whether we're interested in positive, negative, or both types of interactions looks for the m-wise overlap between species and returns the overlap index for each combination

Value

Returns a vector of overlap indices.

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)
species <- relevantSp(germany, 26:ncol(germany))

# re-normalize N.Soil so that everything is on the
# same sign-scale (e.g. the maximum level of a function is the "best" function)
germany$N.Soil <- -1 * germany$N.Soil + max(germany$N.Soil, na.rm = TRUE)

res.list <- lapply(vars, function(x) sAICfun(x, species, germany))
names(res.list) <- vars
```

```
redund <- getRedundancy(vars, species, germany)

getOverlap(redund, m = 2)
getOverlap(redund, m = 2, index = "jaccard")
getOverlap(redund, m = 2, index = "mountford")

#########
# getOverlap takes a matrix of 1s and -1s, and depending on whether we're
# interested in positive, negative, or both types of interactions looks for the
# m-wise overlap
#########
```

`getOverlapSummary` *getOverlapSummary*

Description

`getOverlapSummary` summarizes the number of species necessary for each function including means, SDs, and other metrics

Usage

```
getOverlapSummary(
  overData,
  m = 2,
  type = "positive",
  index = "sorensen",
  denom = "set"
)
```

Arguments

`overData`	Matrix of functions and which species affect them from `getRedundancy`.
`m`	Number of functions. Defaults to 2.
`type`	Are the kinds of effects we're looking at "positive", "negative" or "all".
`index`	Type of overlap index to be used by `getOverlap`.
`denom`	Type of denominator to be used by `getOverlap`.

Details

getOverlapSummary takes a matrix of 1s and -1s, and depending on whether we're interested in positive, negative, or both types of interactions looks for the m-wise overlap between species and then reports summary metrics of mean overlap, SD, and number of combinations

Value

Returns a data frame of the mean overlap, SD, and number of possible combinations.

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)
species <- relevantSp(germany, 26:ncol(germany))

# re-normalize N.Soil so that everything is on the same
# sign-scale (e.g. the maximum level of a function is
# the "best" function)
germany$N.Soil <- -1 * germany$N.Soil + max(germany$N.Soil, na.rm = TRUE)

res.list <- lapply(vars, function(x) sAICfun(x, species, germany))
names(res.list) <- vars

redund <- getRedundancy(vars, species, germany)

getOverlapSummary(redund, m = 2)


#########
# getOverlapSummary takes a matrix of 1s and -1s, and depending on whether we're
# interested in positive, negative, or both types of interactions looks for the
# m-wise overlap and then reports summary metrics of mean overlap, SD, and number of combinations
#########
```

---

getRedundancy *getRedundancy*

Description

getRedundancy examines which species have an effect on which function

Usage

```
getRedundancy(
  vars,
  species,
  data,
  negVars = NA,
  method = "lm",
  combine = "+",
  output = "effect",
```

```
  ...
)
```

Arguments

vars	Vector of column names of functions
species	Vector of column names of species
data	data frame with species presence/absence of values of functions
negVars	Vector of names of species for which a negative coefficient is actually a positive effect.
method	Fitting function for statistical models. Defaults to lm.
combine	How are species combined in the model? Defaults to "+" for additive combinations.
output	Will the output be sign of effect or "coefficient". Defaults to "effect"
...	Other arguments to be supplied to fitting function.

Details

getRedundancy takes a matrix of 1s,0s, and -1s, and depending on whether we're interested in positive, negative, or both types of interactions looks for the m-wise overlap between species and returns the overlap index for each combination. For species whose effect is not different from 0 at the alpha=0.05 level, a 0 is returned.

Value

Returns a matrix of functions and the effect of species on each. 1s, -1s, and 0s for "effect" or coefficients.

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)
species <- relevantSp(germany, 26:ncol(germany))

# re-normalize N.Soil so that everything is on the same
# sign-scale (e.g. the maximum level of a function is
# the "best" function)
germany$N.Soil <- -1 * germany$N.Soil + max(germany$N.Soil, na.rm = TRUE)

res.list <- lapply(vars, function(x) sAICfun(x, species, germany))
names(res.list) <- vars
```

```
getRedundancy(vars, species, germany)
getRedundancy(vars, species, germany, output = "coef")



#########
# takes a vector of responses, the species that may cause them
# and returns a table of 1s, -1s, and 0s with regards to the kind of effect
# or a coefficient table, if asked for.  Arugments can take the form of the fitting function
# how variables are combined, and additional arguments to the fitting function
#########
```

`getStdAndMeanFunctions` *getStdAndMeanFunctions*

Description

`getStdAndMeanFunctions` creates an average function multifunctionality index.

Usage

```
getStdAndMeanFunctions(data, vars, standardizeFunction = standardizeUnitScale)
```

Arguments

`data`	A data frame with functions.
`vars`	The column names of the functions to be assessed.
`standardizeFunction`	A function to standardize each individual function to the same scale, such as `standardizeUnitScale` or `standardizeZScore`

Details

iterates over all functions and standardizes them between 0 and 1. Then it creates an averaged multifunctionality index by averaging over all standardized functions

Value

Returns a data frame with standardized values for each function and an averaged index.

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)

# re-normalize N.Soil so that everything is on the same
# sign-scale (e.g. the maximum level of a function is
# the "best" function)
germany$N.Soil <- -1 * germany$N.Soil + max(germany$N.Soil, na.rm = TRUE)

germany <- cbind(germany, getStdAndMeanFunctions(germany, vars))
```

---

qw *Quote Words*

Description

qw Takes an unquoted vector and adds quotes to it like the qw function in perl.

Usage

```
qw(...)
```

Arguments

...	Any unquoted strings

Details

This is a helper function for data processing. Honestly, I use qw all the time in other languages, and wanted a version for R.

Value

A vector

Author(s)

Jarrett Byrnes.

Examples

```
c("a", "b")

qw(a, b)



# qw - a helper function that we
# will use later to deal with strings
# analagous to qw in PERL
```

`relevantSp` *relevantSp*

Description

`relevantSp` Which species are being used in this analysis.

Usage

```
relevantSp(data, colnums = 26:128)
```

Arguments

`data`	A data frame with presence/abscence of different species.
`colnums`	Column numbers that will be assessed.

Details

Which columns have values that are greater than zero.

Value

A vector of columns names.

Author(s)

Jarrett Byrnes.

sAICfun *sAICfun*

Description

sAICfun examines which species have an effect on which function using a stepwise AIC approach

Usage

```
sAICfun(
  response,
  species,
  data,
  positive.desired = TRUE,
  method = "lm",
  combine = "+",
  ...
)
```

Arguments

response	Name of the response column
species	Vector of column names of species
data	data frame with species presence/abscence of values of functions
positive.desired	Is a positive effect the desired sign. Defaults to TRUE
method	Fitting function for statistical models. Defaults to lm.
combine	How are species combined in the model? Defaults to "+" for additive combinations.
...	Other arguments to be supplied to fitting function.

Details

sAICfun takes a dataset, response, and function, and then uses a stepAIC approach to determine the best model. From that it extracts the species with a positive, negative, and neutral effect on that function.

Value

Returns list of species with positive negative or neutral contributions, the relevant coefficient and effect matrices, and response name

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)
species <- relevantSp(germany, 26:ncol(germany))

# re-normalize N.Soil so that everything is on the same
# sign-scale (e.g. the maximum level of a function is
# the "best" function)
germany$N.Soil <- -1 * germany$N.Soil + max(germany$N.Soil, na.rm = TRUE)

spList <- sAICfun("biomassY3", species, germany)
# " spList
res.list <- lapply(vars, function(x) sAICfun(x, species, germany))
names(res.list) <- vars



#########
# sAICfun takes a dataset, response, and function, and then uses a stepAIC approach
# to determine the best model.  From that it extracts the species with a positive,
# negative, and neutral effect on that function
#########
```

`standardizeUnitScale` *standardizeUnitScale*

Description

`standardizeUnitScale` standardized a variable so its maximum is 1

Usage

```
standardizeUnitScale(afun, min0 = TRUE, maxValue = max(afun, na.rm = TRUE))
```

Arguments

`afun`	A vector of measurements of a function.
`min0`	Must a minimum value be greater than or equal to 0? Defaults to TRUE.
`maxValue`	The maximum valye by which the vector will be standardized. Defaults to the vector's maximum.

Details

Takes a vector and then divides it by a maximum value.

Value

Returns a standardized vector.

Author(s)

Jarrett Byrnes.

`standardizeZScore` *standardizeZScore*

Description

`standardizeZScore` Z-standardizes a vector.

Usage

```
standardizeZScore(afun)
```

Arguments

`afun`	A vector of measurements of a function.

Details

Centers a vector and divides it by its standard deviation.

Value

Returns a z-standardized vector.

Author(s)

Jarrett Byrnes.

stdEffects *stdEffects*

Description

stdEffects obtains the standardized effect of each species on each function

Usage

```
stdEffects(cmat, adf, vars, species)
```

Arguments

cmat	Matrix of coefficients of species effects on functions from getRedundancy with output="coef".
adf	Data frame with plot level data for species and functions.
vars	Names of columns with data for functions in adf.
species	Names of columns with data for species in adf.

Details

stdEffects takes a matrix of coefficients for relationships between species and functions, the data frame used to generate those coefficients and the names of species and function, and then it calculates standardized coefficients using std coef = b *sx/sy

Value

Returns a matrix of standardized coefficients.

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)
species <- relevantSp(germany, 26:ncol(germany))

# re-normalize N.Soil so that everything is on the same
# sign-scale (e.g. the maximum level of a function is
# the "best" function)
germany$N.Soil <- -1 * germany$N.Soil + max(germany$N.Soil, na.rm = TRUE)
```

```
res.list <- lapply(vars, function(x) sAICfun(x, species, germany))
names(res.list) <- vars

coefs <- getRedundancy(vars, species, germany, output = "coef")
stdCoefs <- stdEffects(coefs, germany, vars, species)

#########
# A function that uses the coefficient matrix and information from the
# data to calculate standardized effects of species using the method
# std coef = b *sx/sy
#########
```

---

whichVars *whichVars*

Description

whichVars takes a data frame and the names of a set of columns and returns the names of those columns that do not have an excessive fraction of NA values

Usage

```
whichVars(a.df, vars = NA, thresh = 2/3)
```

Arguments

a.df	A data frame
vars	The names of the columns that contain data of interest
thresh	The fraction of NA values in a column that is acceptable

Details

This is a helper function for data processing.

Value

A vector of column names

Author(s)

Jarrett Byrnes.

Examples

```
data(all_biodepth)
allVars <- qw(biomassY3, root3, N.g.m2, light3, N.Soil, wood3, cotton3)

germany <- subset(all_biodepth, all_biodepth$location == "Germany")

vars <- whichVars(germany, allVars)
```

Index